

# RX113 Group

User's Manual: Hardware

RENESAS 32-Bit MCU  
RX Family / RX100 Series

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

## 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.

## NOTES FOR CMOS DEVICES

- (1) **VOLTAGE APPLICATION WAVEFORM AT INPUT PIN:** Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).
- (2) **HANDLING OF UNUSED INPUT PINS:** Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.
- (3) **PRECAUTION AGAINST ESD:** A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.
- (4) **STATUS BEFORE INITIALIZATION:** Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.
- (5) **POWER ON/OFF SEQUENCE:** In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.
- (6) **INPUT OF SIGNAL DURING POWER OFF STATE :** Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

## 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 document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

### 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 a product with a different part number, confirm that the change will not lead to problems.

- The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

## Specification Differences between Products

There are the following specification differences in these MCU products depending on the package.

**Table 1 Specification Differences Depending on Packages**

Chapter		Specification Differences	
		Products with 40 pins or less	Products with 48 pins or more
9. Clock Generation Circuit	9.8.4 Notes on Sub-Clock	Although sub-clock oscillator pins are not available, the sub-clock circuit must be initialized at a cold start.	At a cold start, initialize the sub-clock control circuit regardless of whether the sub-clock is in use or not.

# 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 RX113 Group. Before using any of the documents, please visit our website to verify that you have the most up-to-date available version of the document.

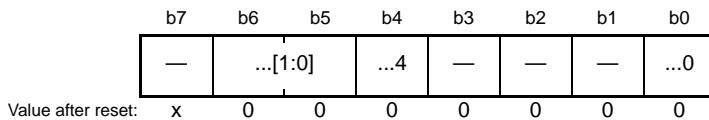
Document Type	Contents	Document Title	Document No.
Datasheet	Overview of hardware and electrical characteristics	RX113 Group Datasheet	R01DS0216EJ
User's Manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	RX113 Group User's Manual: Hardware	This document
User's Manual: Software	Detailed descriptions of the CPU and instruction set	RX Family 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 xxxh



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	...0	.....	0: ..... 1: <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">Setting prohibited</span> (3)	<span style="border: 1px solid black; border-radius: 50%; padding: 2px;">R/W</span> (1)
b3 to b1	—	<span style="border: 1px solid black; border-radius: 50%; padding: 2px;">Reserved</span> (2)	These bits are read as 0. The write value should be 0.	R/W
b4	...4	.....	0: ..... 1: .....	R
b6, b5	...[1:0]	.....	0 0: ..... 0 1: ..... <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">Settings other than above are prohibited.</span> (3)	R/(W)*1
b7	—	Reserved	The read value is undefined. Writing to this bit has no effect.	R

- (1) R/W: The bit or field is readable and writable.  
 R/(W): The bit or field is readable and writable. However, writing to this bit or field has some limitations. For details on the limitations, see the description or notes of respective registers.  
 R: The bit or field is readable. Writing to this bit or field has no effect.
- (2) Reserved.  
 Use the specified value when writing to this bit or field; otherwise, the correct operation is not guaranteed.
- (3) Setting prohibited. The correct operation is not guaranteed if such a setting is performed.

### 3. List of Abbreviations and Acronyms

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

All trademarks and registered trademarks are the property of their respective owners.

SuperFlash® is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

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



# Contents

Features .....	42
1. Overview .....	43
1.1 Outline of Specifications .....	43
1.2 List of Products .....	47
1.3 Block Diagram .....	49
1.4 Pin Functions .....	50
1.5 Pin Assignments .....	54
2. CPU .....	65
2.1 Features .....	65
2.2 Register Set of the CPU .....	66
2.2.1 General-Purpose Registers (R0 to R15) .....	67
2.2.2 Control Registers .....	67
2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP) .....	68
2.2.2.2 Interrupt Table Register (INTB) .....	68
2.2.2.3 Program Counter (PC) .....	68
2.2.2.4 Processor Status Word (PSW) .....	69
2.2.2.5 Backup PC (BPC) .....	70
2.2.2.6 Backup PSW (BPSW) .....	71
2.2.2.7 Fast Interrupt Vector Register (FINTV) .....	71
2.2.3 Register Associated with DSP Instructions .....	71
2.2.3.1 Accumulator (ACC) .....	71
2.3 Processor Mode .....	72
2.3.1 Supervisor Mode .....	72
2.3.2 User Mode .....	72
2.3.3 Privileged Instruction .....	72
2.3.4 Switching between Processor Modes .....	72
2.4 Data Types .....	73
2.5 Endian .....	73
2.5.1 Switching the Endian .....	73
2.5.2 Access to I/O Registers .....	77
2.5.3 Notes on Access to I/O Registers .....	77
2.5.4 Data Arrangement .....	77
2.5.4.1 Data Arrangement in Registers .....	77
2.5.4.2 Data Arrangement in Memory .....	78
2.5.5 Notes on the Allocation of Instruction Codes .....	78
2.6 Vector Table .....	79
2.6.1 Fixed Vector Table .....	79
2.6.2 Relocatable Vector Table .....	80
2.7 Operation of Instructions .....	81
2.7.1 Data Prefetching by the RMPA Instruction and the String-Manipulation Instructions .....	81
2.8 Pipeline .....	81

2.8.1	Overview .....	81
2.8.2	Instructions and Pipeline Processing .....	83
2.8.2.1	Instructions Converted into Single Micro-Operation and Pipeline Processing .....	83
2.8.2.2	Instructions Converted into Multiple Micro-Operations and Pipeline Processing .....	85
2.8.2.3	Pipeline Basic Operation .....	88
2.8.3	Calculation of the Instruction Processing Time .....	90
2.8.4	Numbers of Cycles for Response to Interrupts .....	91
3.	Operating Modes .....	92
3.1	Operating Mode Types and Selection .....	92
3.2	Register Descriptions .....	93
3.2.1	Mode Monitor Register (MDMONR) .....	93
3.2.2	System Control Register 1 (SYSCR1) .....	94
3.3	Details of Operating Modes .....	95
3.3.1	Single-Chip Mode .....	95
3.3.2	Boot Mode .....	95
3.3.2.1	Boot Mode (USB Interface) .....	95
3.3.2.2	Boot Mode (SCI) .....	95
3.4	Transitions of Operating Modes .....	95
3.4.1	Mode Setting Pin Levels and Operating Mode Transitions .....	95
4.	Address Space .....	96
4.1	Address Space .....	96
5.	I/O Registers .....	98
5.1	I/O Register Addresses (Address Order) .....	100
6.	Resets .....	123
6.1	Overview .....	123
6.2	Register Descriptions .....	125
6.2.1	Reset Status Register 0 (RSTSR0) .....	125
6.2.2	Reset Status Register 1 (RSTSR1) .....	126
6.2.3	Reset Status Register 2 (RSTSR2) .....	127
6.2.4	Software Reset Register (SWRR) .....	128
6.3	Operation .....	129
6.3.1	RES# Pin Reset .....	129
6.3.2	Power-On Reset .....	129
6.3.3	Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset .....	131
6.3.4	Independent Watchdog Timer Reset .....	132
6.3.5	Software Reset .....	133
6.3.6	Determination of Cold/Warm Start .....	133
6.3.7	Determination of Reset Generation Source .....	134
7.	Option-Setting Memory .....	135
7.1	Overview .....	135
7.2	Register Descriptions .....	136

7.2.1	Option Function Select Register 0 (OFS0)	136
7.2.2	Option Function Select Register 1 (OFS1)	138
7.2.3	Endian Select Register (MDE)	140
7.3	Usage Note	141
7.3.1	Setting Example of Option-Setting Memory	141
8.	Voltage Detection Circuit (LVDAa)	142
8.1	Overview	142
8.2	Register Descriptions	145
8.2.1	Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1)	145
8.2.2	Voltage Monitoring 1 Circuit Status Register (LVD1SR)	146
8.2.3	Voltage Monitoring 2 Circuit Control Register 1 (LVD2CR1)	147
8.2.4	Voltage Monitoring 2 Circuit Status Register (LVD2SR)	148
8.2.5	Voltage Monitoring Circuit Control Register (LVCMPCR)	149
8.2.6	Voltage Detection Level Select Register (LVDLVLR)	150
8.2.7	Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0)	151
8.2.8	Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0)	152
8.3	VCC Input Voltage Monitor	153
8.3.1	Monitoring Vdet1	153
8.3.2	Monitoring Vdet2	153
8.4	Interrupt and Reset from Voltage Monitoring 1	154
8.5	Interrupt and Reset from Voltage Monitoring 2	156
8.6	Event Link Output	158
8.6.1	Interrupt Handling and Event Linking	158
9.	Clock Generation Circuit	159
9.1	Overview	159
9.2	Register Descriptions	161
9.2.1	System Clock Control Register (SCKCR)	161
9.2.2	System Clock Control Register 3 (SCKCR3)	163
9.2.3	PLL Control Register (PLLCR)	164
9.2.4	PLL Control Register 2 (PLLCR2)	165
9.2.5	USB-dedicated PLL Control Register (UPLLCR)	166
9.2.6	USB-dedicated PLL Control Register 2 (UPLLCR2)	167
9.2.7	Main Clock Oscillator Control Register (MOSCCR)	168
9.2.8	Sub-Clock Oscillator Control Register (SOSCCR)	169
9.2.9	Low-Speed On-Chip Oscillator Control Register (LOCOCR)	170
9.2.10	IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR)	171
9.2.11	High-Speed On-Chip Oscillator Control Register (HOCOCCR)	172
9.2.12	Oscillation Stabilization Flag Register (OSCOVFSR)	173
9.2.13	Oscillation Stop Detection Control Register (OSTDCR)	175
9.2.14	Oscillation Stop Detection Status Register (OSTDSR)	176
9.2.15	LCD Source Clock Control Register (LCDSCLKCR)	177

9.2.16	LCD Source Clock Control Register 2 (LCDSCLKCR2) .....	178
9.2.17	Main Clock Oscillator Wait Control Register (MOSCWTCR) .....	179
9.2.18	High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR) .....	180
9.2.19	CLKOUT Output Control Register (CKOCR) .....	181
9.2.20	Main Clock Oscillator Forced Oscillation Control Register (MOFCR) .....	182
9.3	Main Clock Oscillator .....	183
9.3.1	Connecting a Crystal .....	183
9.3.2	External Clock Input .....	184
9.3.3	Handling of Pins When the Main Clock is Not Used .....	184
9.3.4	Notes on the External Clock Input .....	184
9.4	Sub-Clock Oscillator .....	185
9.4.1	Connecting 32.768-kHz Crystal .....	185
9.4.2	Handling of Pins When Sub-Clock is Not Used .....	185
9.5	Oscillation Stop Detection Function .....	186
9.5.1	Oscillation Stop Detection and Operation after Detection .....	186
9.5.2	Oscillation Stop Detection Interrupts .....	187
9.6	PLL Circuit .....	188
9.7	Internal Clock .....	188
9.7.1	System Clock .....	188
9.7.2	Peripheral Module Clock .....	188
9.7.3	FlashIF Clock .....	188
9.7.4	USB Clock .....	189
9.7.5	LCD Source Clock .....	189
9.7.6	CAC Clock .....	189
9.7.7	RTC-Dedicated Clock .....	189
9.7.8	IWDT-Dedicated Clock .....	189
9.7.9	SSI Clock .....	189
9.8	Usage Notes .....	190
9.8.1	Notes on Clock Generation Circuit .....	190
9.8.2	Notes on Resonator .....	190
9.8.3	Notes on Board Design .....	190
9.8.4	Notes on Sub-Clock .....	191
10.	Clock Frequency Accuracy Measurement Circuit (CAC) .....	195
10.1	Overview .....	195
10.2	Register Descriptions .....	197
10.2.1	CAC Control Register 0 (CACR0) .....	197
10.2.2	CAC Control Register 1 (CACR1) .....	198
10.2.3	CAC Control Register 2 (CACR2) .....	199
10.2.4	CAC Interrupt Request Enable Register (CAICR) .....	200
10.2.5	CAC Status Register (CASTR) .....	201
10.2.6	CAC Upper-Limit Value Setting Register (CAULVR) .....	202

10.2.7	CAC Lower-Limit Value Setting Register (CALLVR) .....	202
10.2.8	CAC Counter Buffer Register (CACNTBR) .....	202
10.3	Operation .....	203
10.3.1	Measuring Clock Frequency .....	203
10.3.2	Digital Filtering of Signals on the CACREF Pin .....	204
10.4	Interrupt Requests .....	204
10.5	Usage Notes .....	205
10.5.1	Module Stop Function Setting .....	205
11.	Low Power Consumption .....	206
11.1	Overview .....	206
11.2	Register Descriptions .....	210
11.2.1	Standby Control Register (SBYCR) .....	210
11.2.2	Module Stop Control Register A (MSTPCRA) .....	211
11.2.3	Module Stop Control Register B (MSTPCRB) .....	212
11.2.4	Module Stop Control Register C (MSTPCRC) .....	214
11.2.5	Module Stop Control Register D (MSTPCRD) .....	215
11.2.6	Operating Power Control Register (OPCCR) .....	216
11.2.7	Sub Operating Power Control Register (SOPCCR) .....	217
11.2.8	Sleep Mode Return Clock Source Switching Register (RSTCKCR) .....	222
11.3	Reducing Power Consumption by Switching Clock Signals .....	224
11.4	Module Stop Function .....	224
11.5	Function for Lower Operating Power Consumption .....	224
11.5.1	Setting Operating Power Control Mode .....	224
11.6	Low Power Consumption Modes .....	226
11.6.1	Sleep Mode .....	226
11.6.1.1	Entry to Sleep Mode .....	226
11.6.1.2	Exit from Sleep Mode .....	227
11.6.1.3	Sleep Mode Return Clock Source Switching Function .....	227
11.6.2	Deep Sleep Mode .....	228
11.6.2.1	Entry to Deep Sleep Mode .....	228
11.6.2.2	Exit from Deep Sleep Mode .....	229
11.6.3	Software Standby Mode .....	230
11.6.3.1	Entry to Software Standby Mode .....	230
11.6.3.2	Exit from Software Standby Mode .....	231
11.6.3.3	Example of Software Standby Mode Application .....	232
11.7	Usage Notes .....	233
11.7.1	I/O Port States .....	233
11.7.2	Module Stop State of DTC .....	233
11.7.3	On-Chip Peripheral Module Interrupts .....	233
11.7.4	Write Access to MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD .....	233
11.7.5	Timing of WAIT Instructions .....	233

11.7.6	Rewrite the Register by DTC in Sleep Mode .....	233
12.	Register Write Protection Function .....	234
12.1	Register Descriptions .....	235
12.1.1	Protect Register (PRCR) .....	235
13.	Exception Handling .....	236
13.1	Exception Events .....	236
13.1.1	Undefined Instruction Exception .....	237
13.1.2	Privileged Instruction Exception .....	237
13.1.3	Reset .....	237
13.1.4	Non-Maskable Interrupt .....	237
13.1.5	Interrupts .....	237
13.1.6	Unconditional Trap .....	237
13.2	Exception Handling Procedure .....	238
13.3	Acceptance of Exception Events .....	240
13.3.1	Acceptance Timing and Saved PC Value .....	240
13.3.2	Vector and Site for Saving the Values in the PC and PSW .....	240
13.4	Hardware Processing for Accepting and Returning from Exceptions .....	241
13.5	Hardware Pre-Processing .....	242
13.5.1	Undefined Instruction Exception .....	242
13.5.2	Privileged Instruction Exception .....	242
13.5.3	Reset .....	242
13.5.4	Non-Maskable Interrupt .....	242
13.5.5	Interrupt .....	243
13.5.6	Unconditional Trap .....	243
13.6	Return from Exception Handling Routine .....	244
13.7	Priority of Exception Events .....	244
14.	Interrupt Controller (ICUb) .....	245
14.1	Overview .....	245
14.2	Register Descriptions .....	247
14.2.1	Interrupt Request Register n (IRn) (n = interrupt vector number) .....	247
14.2.2	Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh) .....	248
14.2.3	Interrupt Source Priority Register n (IPRn) (n = 000 to 249) .....	249
14.2.4	Fast Interrupt Set Register (FIR) .....	250
14.2.5	Software Interrupt Activation Register (SWINTR) .....	251
14.2.6	DTC Activation Enable Register n (DTCERn) (n = interrupt vector number) .....	252
14.2.7	IRQ Control Register i (IRQCRi) (i = 0 to 7) .....	253
14.2.8	IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0) .....	254
14.2.9	IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0) .....	255
14.2.10	Non-Maskable Interrupt Status Register (NMISR) .....	256
14.2.11	Non-Maskable Interrupt Enable Register (NMIER) .....	258
14.2.12	Non-Maskable Interrupt Status Clear Register (NMICLR) .....	259

14.2.13	NMI Pin Interrupt Control Register (NMICR)	260
14.2.14	NMI Pin Digital Filter Enable Register (NMIFLTE)	260
14.2.15	NMI Pin Digital Filter Setting Register (NMIFLTC)	261
14.3	Vector Table	262
14.3.1	Interrupt Vector Table	262
14.3.2	Fast Interrupt Vector Table	270
14.3.3	Non-maskable Interrupt Vector Table	270
14.4	Interrupt Operation	270
14.4.1	Detecting Interrupts	270
14.4.1.1	Operation of Status Flags for Edge-Detected Interrupts	270
14.4.1.2	Operation of Status Flags for Level-Detected Interrupts	272
14.4.2	Enabling and Disabling Interrupt Sources	273
14.4.3	Selecting Interrupt Request Destinations	273
14.4.4	Determining Priority	274
14.4.5	Multiple Interrupts	274
14.4.6	Fast Interrupt	275
14.4.7	Digital Filter	275
14.4.8	External Pin Interrupts	276
14.5	Non-maskable Interrupt Operation	277
14.6	Return from Power-Down States	278
14.6.1	Return from Sleep Mode or Deep Sleep Mode	278
14.6.2	Return from Software Standby Mode	278
14.7	Usage Note	278
14.7.1	Note on WAIT Instruction Used with Non-Maskable Interrupt	278
15.	Buses	279
15.1	Overview	279
15.2	Description of Buses	281
15.2.1	CPU Buses	281
15.2.2	Memory Buses	281
15.2.3	Internal Main Buses	281
15.2.4	Internal Peripheral Buses	282
15.2.5	Write Buffer Function (Internal Peripheral Bus)	283
15.2.6	Parallel Operation	284
15.2.7	Restrictions	284
15.3	Register Descriptions	285
15.3.1	Bus Error Status Clear Register (BERCLR)	285
15.3.2	Bus Error Monitoring Enable Register (BEREN)	285
15.3.3	Bus Error Status Register 1 (BERSR1)	286
15.3.4	Bus Error Status Register 2 (BERSR2)	286
15.3.5	Bus Priority Control Register (BUSPRI)	287
15.4	Bus Error Monitoring Section	289

15.4.1	Type of Bus Error .....	289
15.4.1.1	Illegal Address Access .....	289
15.4.1.2	Timeout .....	289
15.4.2	Operations When a Bus Error Occurs .....	289
15.4.3	Conditions Leading to Bus Errors .....	290
16.	Data Transfer Controller (DTCa) .....	291
16.1	Overview .....	291
16.2	Register Descriptions .....	293
16.2.1	DTC Mode Register A (MRA) .....	293
16.2.2	DTC Mode Register B (MRB) .....	294
16.2.3	DTC Transfer Source Register (SAR) .....	295
16.2.4	DTC Transfer Destination Register (DAR) .....	295
16.2.5	DTC Transfer Count Register A (CRA) .....	296
16.2.6	DTC Transfer Count Register B (CRB) .....	297
16.2.7	DTC Control Register (DTCCR) .....	297
16.2.8	DTC Vector Base Register (DTCVBR) .....	298
16.2.9	DTC Address Mode Register (DTCADM0D) .....	298
16.2.10	DTC Module Start Register (DTCST) .....	299
16.2.11	DTC Status Register (DTCSTS) .....	300
16.3	Activation Sources .....	301
16.3.1	Allocating Transfer Information and DTC Vector Table .....	301
16.4	Operation .....	303
16.4.1	Transfer Information Read Skip Function .....	305
16.4.2	Transfer Information Write-Back Skip Function .....	306
16.4.3	Normal Transfer Mode .....	307
16.4.4	Repeat Transfer Mode .....	308
16.4.5	Block Transfer Mode .....	309
16.4.6	Chain Transfer .....	310
16.4.7	Operation Timing .....	311
16.4.8	Execution Cycles of the DTC .....	314
16.4.9	DTC Bus Mastership Release Timing .....	314
16.5	DTC Setting Procedure .....	315
16.6	Examples of DTC Usage .....	316
16.6.1	Normal Transfer .....	316
16.6.2	Chain Transfer When the Counter = 0 .....	317
16.7	Interrupt Source .....	318
16.8	Event Link .....	318
16.9	Low Power Consumption Function .....	319
16.10	Usage Notes .....	320
16.10.1	Transfer Information Start Address .....	320
16.10.2	Allocating Transfer Information .....	320



17.	Event Link Controller (ELC)	321
17.1	Overview	321
17.2	Register Descriptions	323
17.2.1	Event Link Control Register (ELCR)	323
17.2.2	Event Link Setting Register n (ELSRn) (n = 1 to 4, 7, 10, 12, 14, 15, 17, 18, 20, 22, 24, 25)	324
17.2.3	Event Link Option Setting Register A (ELOPA)	326
17.2.4	Event Link Option Setting Register B (ELOPB)	327
17.2.5	Event Link Option Setting Register C (ELOPC)	327
17.2.6	Event Link Option Setting Register D (ELOPD)	328
17.2.7	Port Group Setting Register 1 (PGR1)	329
17.2.8	Port Group Control Register 1 (PGC1)	330
17.2.9	Port Buffer Register 1 (PDBF1)	331
17.2.10	Event Link Port Setting Register n (PELn) (n = 0, 1)	332
17.2.11	Event Link Software Event Generation Register (ELSEGR)	333
17.3	Operation	334
17.3.1	Relation between Interrupt Handling and Event Linking	334
17.3.2	Event Linkage	335
17.3.3	Operation of Peripheral Timer Modules When Event is Input	336
17.3.4	Operation of A/D and D/A Converters When Event is Input	336
17.3.5	I/O Port Operation upon Event Input and Event Generation	336
17.3.6	Example of Procedure for Linking Events	341
17.4	Usage Notes	342
17.4.1	Setting ELSRn Register	342
17.4.2	Setting Bit-Rotating Operation of Output Port Groups	342
17.4.3	Linking DTC Transfer End Signals as Events	342
17.4.4	Setting Clocks	342
17.4.5	Module Stop Function Setting	342
18.	I/O Ports	343
18.1	Overview	343
18.2	I/O Port Configuration	345
18.3	Register Descriptions	354
18.3.1	Port Direction Register (PDR)	354
18.3.2	Port Output Data Register (PODR)	355
18.3.3	Port Input Data Register (PIDR)	356
18.3.4	Port Mode Register (PMR)	357
18.3.5	Open Drain Control Register 0 (ODR0)	358
18.3.6	Open Drain Control Register 1 (ODR1)	359
18.3.7	Pull-Up Control Register (PCR)	360
18.3.8	Port Switching Register A (PSRA)	361
18.4	Initialization of the Port Direction Register (PDR)	362

18.5	Handling of Unused Pins .....	363
19.	Multi-Function Pin Controller (MPC) .....	364
19.1	Overview .....	364
19.2	Register Descriptions .....	374
19.2.1	Write-Protect Register (PWPR) .....	374
19.2.2	P0n Pin Function Control Register (P0nPFS) (n = 2, 4, 7) .....	375
19.2.3	P1n Pin Function Control Registers (P1nPFS) (n = 0 to 7) .....	376
19.2.4	P2n Pin Function Control Registers (P2nPFS) (n = 0 to 7) .....	378
19.2.5	P3n Pin Function Control Registers (P3nPFS) (n = 0 to 2) .....	380
19.2.6	P4n Pin Function Control Registers (P4nPFS) (n = 0 to 4, 6) .....	382
19.2.7	P5n Pin Function Select Registers (P5nPFS) (n = 0 to 6) .....	383
19.2.8	P9n Pin Function Select Registers (P9nPFS) (n = 0 to 2) .....	384
19.2.9	PAn Pin Function Control Registers (PAnPFS) (n = 0 to 7) .....	385
19.2.10	PBn Pin Function Control Registers (PBnPFS) (n = 0 to 7) .....	387
19.2.11	PCn Pin Function Control Registers (PCnPFS) (n = 0 to 7) .....	389
19.2.12	PDn Pin Function Control Registers (PDnPFS) (n = 0 to 4) .....	391
19.2.13	PEn Pin Function Control Registers (PEnPFS) (n = 0 to 7) .....	392
19.2.14	PFn Pin Function Control Registers (PFnPFS) (n = 6, 7) .....	393
19.2.15	PJn Pin Function Control Registers (PJnPFS) (n = 0, 2, 3, 6, 7) .....	394
19.3	Usage Notes .....	395
19.3.1	Procedure for Specifying I/O Pin Functions .....	395
19.3.2	Notes on MPC Register Setting .....	395
19.3.3	Notes on Using Analog Functions .....	396
19.3.4	Notes on Using LCD Controller/Driver (LCDC) Function .....	396
19.3.5	Notes on Using Capacitive Touch Sensing Unit (CTSU) Function .....	396
20.	Multi-Function Timer Pulse Unit 2 (MTU2a) .....	397
20.1	Overview .....	397
20.2	Register Descriptions .....	402
20.2.1	Timer Control Register (TCR) .....	402
20.2.2	Timer Mode Register (TMDR) .....	405
20.2.3	Timer I/O Control Register (TIOR) .....	407
20.2.4	Timer Compare Match Clear Register (TCNTCMPCLR) .....	418
20.2.5	Timer Interrupt Enable Register (TIER) .....	419
20.2.6	Timer Status Register (TSR) .....	422
20.2.7	Timer Buffer Operation Transfer Mode Register (TBTM) .....	423
20.2.8	Timer Input Capture Control Register (TICCR) .....	424
20.2.9	Timer A/D Converter Start Request Control Register (TADCR) .....	425
20.2.10	Timer A/D Converter Start Request Cycle Set Registers A and B (TADCORA and TADCORB) .....	426
20.2.11	Timer A/D Converter Start Request Cycle Set Buffer Registers A and B (TADCOBRA and TADCOBRB) .....	426

20.2.12	Timer Counter (TCNT) .....	427
20.2.13	Timer General Register (TGR) .....	427
20.2.14	Timer Start Registers (TSTR) .....	428
20.2.15	Timer Synchronous Registers (TSYR) .....	430
20.2.16	Timer Read/Write Enable Registers (TRWER) .....	431
20.2.17	Timer Output Master Enable Registers (TOER) .....	432
20.2.18	Timer Output Control Registers 1 (TOCR1) .....	433
20.2.19	Timer Output Control Registers 2 (TOCR2) .....	435
20.2.20	Timer Output Level Buffer Registers (TOLBR) .....	438
20.2.21	Timer Gate Control Registers (TGCR) .....	439
20.2.22	Timer Subcounters (TCNTS) .....	440
20.2.23	Timer Dead Time Data Registers (TDDR) .....	440
20.2.24	Timer Cycle Data Registers (TCDR) .....	441
20.2.25	Timer Cycle Buffer Registers (TCBR) .....	441
20.2.26	Timer Interrupt Skipping Set Registers (TITCR) .....	442
20.2.27	Timer Interrupt Skipping Counters (TITCNT) .....	444
20.2.28	Timer Buffer Transfer Set Registers (TBTER) .....	445
20.2.29	Timer Dead Time Enable Registers (TDER) .....	446
20.2.30	Timer Waveform Control Registers (TWCR) .....	447
20.2.31	Noise Filter Control Registers (NFCR) .....	448
20.2.32	Bus Master Interface .....	451
20.3	Operation .....	452
20.3.1	Basic Functions .....	452
20.3.2	Synchronous Operation .....	458
20.3.3	Buffer Operation .....	460
20.3.4	Cascaded Operation .....	465
20.3.5	PWM Modes .....	470
20.3.6	Phase Counting Mode .....	474
20.3.7	Reset-Synchronized PWM Mode .....	480
20.3.8	Complementary PWM Mode .....	483
20.3.9	A/D Converter Start Request Delaying Function .....	515
20.3.10	External Pulse Width Measurement .....	518
20.3.11	Dead Time Compensation .....	519
20.3.12	Noise Filter .....	522
20.4	Interrupt Sources .....	523
20.4.1	Interrupt Sources and Priorities .....	523
20.4.2	DTC Activation .....	524
20.4.3	A/D Converter Activation .....	524
20.5	Operation Timing .....	526
20.5.1	Input/Output Timing .....	526
20.5.2	Interrupt Signal Timing .....	532

20.6	Usage Notes .....	535
20.6.1	Module Clock Stop Mode Setting .....	535
20.6.2	Input Clock Restrictions .....	535
20.6.3	Note on Cycle Setting .....	536
20.6.4	Contention between TCNT Write and Clear Operations .....	536
20.6.5	Contention between TCNT Write and Increment Operations .....	537
20.6.6	Contention between TGR Write Operation and Compare Match .....	537
20.6.7	Contention between Buffer Register Write Operation and Compare Match .....	538
20.6.8	Contention between Buffer Register Write and TCNT Clear Operations .....	539
20.6.9	Contention between TGR Read Operation and Input Capture .....	540
20.6.10	Contention between TGR Write Operation and Input Capture .....	541
20.6.11	Contention between Buffer Register Write Operation and Input Capture .....	542
20.6.12	Contention between MTU2.TCNT Write Operation and Overflow/ Underflow in Cascaded Operation .....	543
20.6.13	Counter Value When Count Operation is Stopped in Complementary PWM Mode .....	544
20.6.14	Buffer Operation Setting in Complementary PWM Mode .....	544
20.6.15	Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode .....	545
20.6.16	Overflow Flags in Reset-Synchronized PWM Mode .....	546
20.6.17	Contention between Overflow/Underflow and Counter Clearing .....	547
20.6.18	Contention between TCNT Write Operation and Overflow/Underflow .....	548
20.6.19	Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode .....	548
20.6.20	Output Level in Complementary PWM Mode or Reset-Synchronized PWM Mode .....	548
20.6.21	Interrupts during Periods in the Module Stop State .....	548
20.6.22	Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection .....	549
20.6.23	Notes When Complementary PWM Mode Output Protection Functions are Not Used .....	549
20.6.24	Point for Caution Regarding MTU5.TCNT and MTU5.TGR Registers .....	549
20.6.25	Points for Caution to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode .....	550
20.6.26	Continuous Output of Interrupt Signal in Response to a Compare Match .....	552
20.7	MTU Output Pin Initialization .....	553
20.7.1	Operating Modes .....	553
20.7.2	Operation in Case of Re-Setting Due to Error during Operation .....	553
20.7.3	Overview of Pin Initialization Procedures and Mode Transitions in Case of Error during Operation .....	554
20.8	Operations Linked by the ELC .....	583
20.8.1	Event Signal Output to the ELC .....	583
20.8.2	MTU Operations in Response to Receiving Event Signals from the ELC .....	583
20.8.3	Notes on MTU by Event Signal Reception from the ELC .....	584
21.	Port Output Enable 2 (POE2a) .....	585
21.1	Overview .....	585
21.2	Register Descriptions .....	588

21.2.1	Input Level Control/Status Register 1 (ICSR1) .....	588
21.2.2	Output Level Control/Status Register 1 (OCSR1) .....	590
21.2.3	Input Level Control/Status Register 2 (ICSR2) .....	591
21.2.4	Software Port Output Enable Register (SPOER) .....	592
21.2.5	Port Output Enable Control Register 1 (POECR1) .....	593
21.2.6	Port Output Enable Control Register 2 (POECR2) .....	594
21.2.7	Input Level Control/Status Register 3 (ICSR3) .....	595
21.3	Operation .....	596
21.3.1	Input Level Detection Operation .....	598
21.3.2	Output-Level Compare Operation .....	599
21.3.3	High-Impedance Control Using Registers .....	600
21.3.4	High-Impedance Control on Detection of Stopped Oscillation .....	600
21.3.5	Release from the High-Impedance .....	600
21.4	Interrupts .....	601
21.5	Usage Notes .....	601
21.5.1	Transitions to Software Standby Mode .....	601
21.5.2	When the POE Is Not Used .....	601
21.5.3	Specifying Pins Corresponding to the MTU .....	601
22.	8-Bit Timer (TMR) .....	602
22.1	Overview .....	602
22.2	Register Descriptions .....	607
22.2.1	Timer Counter (TCNT) .....	607
22.2.2	Time Constant Register A (TCORA) .....	608
22.2.3	Time Constant Register B (TCORB) .....	608
22.2.4	Timer Control Register (TCR) .....	609
22.2.5	Timer Counter Control Register (TCCR) .....	610
22.2.6	Timer Control/Status Register (TCSR) .....	612
22.2.7	Time Count Start Register (TCSTR) .....	614
22.3	Operation .....	615
22.3.1	Pulse Output .....	615
22.3.2	Reset Input .....	616
22.4	Operation Timing .....	617
22.4.1	TCNT Count Timing .....	617
22.4.2	Timing of Interrupt Signal Output on a Compare Match .....	618
22.4.3	Timing of Timer Output at Compare Match .....	618
22.4.4	Timing of Counter Clear by Compare Match .....	619
22.4.5	Timing of the External Reset for TCNT .....	619
22.4.6	Timing of Interrupt Signal Output on an Overflow .....	620
22.5	Operation with Cascaded Connection .....	621
22.5.1	16-Bit Count Mode .....	621
22.5.2	Compare Match Count Mode .....	621

22.6	Interrupt Sources .....	622
22.6.1	Interrupt Sources and DTC Activation .....	622
22.7	Link Operation by ELC .....	623
22.7.1	Event Signal Output to ELC .....	623
22.7.2	TMR Operation when Receiving an Event Signal from ELC .....	623
22.7.3	Notes on Operating TMR According to an Event Signal from ELC .....	624
22.8	Usage Notes .....	625
22.8.1	Module Stop State Setting .....	625
22.8.2	Notes on Setting Cycle .....	625
22.8.3	Conflict between TCNT Write and Counter Clear .....	625
22.8.4	Conflict between TCNT Write and Increment .....	626
22.8.5	Conflict between TCORA or TCORB Write and Compare Match .....	626
22.8.6	Conflict between Compare Matches A and B .....	627
22.8.7	Switching of Frequency Dividing Clocks and TCNT Operation .....	627
22.8.8	Clock Source Setting with Cascaded Connection .....	629
22.8.9	Continuous Output of Compare Match Interrupt Signal .....	629
<b>23.</b>	<b>Compare Match Timer (CMT) .....</b>	<b>630</b>
23.1	Overview .....	630
23.2	Register Descriptions .....	631
23.2.1	Compare Match Timer Start Register 0 (CMSTR0) .....	631
23.2.2	Compare Match Timer Start Register 1 (CMSTR1) .....	631
23.2.3	Compare Match Timer Control Register (CMCR) .....	632
23.2.4	Compare Match Counter (CMCNT) .....	633
23.2.5	Compare Match Constant Register (CMCOR) .....	633
23.3	Operation .....	634
23.3.1	Periodic Count Operation .....	634
23.3.2	CMCNT Count Timing .....	634
23.4	Interrupts .....	635
23.4.1	Interrupt Sources .....	635
23.4.2	Timing of Compare Match Interrupt Generation .....	635
23.5	Link Operations by ELC .....	636
23.5.1	Event Signal Output to ELC .....	636
23.5.2	CMT Operation When Receiving an Event Signal from ELC .....	636
23.5.3	Notes on Operating CMT According to an Event Signal from ELC .....	636
23.6	Usage Notes .....	637
23.6.1	Setting the Module Stop Function .....	637
23.6.2	Conflict between CMCNT Counter Writing and Compare Match .....	637
23.6.3	Conflict between CMCNT Counter Writing and Incrementing .....	637
<b>24.</b>	<b>Realtime Clock (RTCA) .....</b>	<b>638</b>
24.1	Overview .....	638
24.2	Register Descriptions .....	640

24.2.1	64-Hz Counter (R64CNT) .....	640
24.2.2	Second Counter (RSECCNT)/Binary Counter 0 (BCNT0) .....	641
24.2.3	Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1) .....	642
24.2.4	Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2) .....	643
24.2.5	Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3) .....	644
24.2.6	Date Counter (RDAYCNT) .....	645
24.2.7	Month Counter (RMONCNT) .....	646
24.2.8	Year Counter (RYRCNT) .....	646
24.2.9	Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR) .....	647
24.2.10	Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR) .....	648
24.2.11	Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR) .....	649
24.2.12	Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR) ...	650
24.2.13	Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER) ..	651
24.2.14	Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER) .....	652
24.2.15	Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER) .....	653
24.2.16	Year Alarm Enable Register (RYRAREN)/Binary Counter 3 Alarm Enable Register (BCNT3AER) .....	654
24.2.17	RTC Control Register 1 (RCR1) .....	655
24.2.18	RTC Control Register 2 (RCR2) .....	656
24.2.19	RTC Control Register 3 (RCR3) .....	660
24.2.20	Time Error Adjustment Register (RADJ) .....	661
24.3	Operation .....	662
24.3.1	Outline of Initial Settings of Registers after Power On .....	662
24.3.2	Clock and Count Mode Setting Procedure .....	663
24.3.3	Setting the Time .....	664
24.3.4	30-Second Adjustment .....	664
24.3.5	Reading 64-Hz Counter and Time .....	665
24.3.6	Alarm Function .....	666
24.3.7	Procedure for Disabling Alarm Interrupt .....	667
24.3.8	Time Error Adjustment Function .....	667
24.3.8.1	Automatic Adjustment .....	667
24.3.8.2	Adjustment by Software .....	668
24.3.8.3	Procedure for Changing the Mode of Adjustment .....	669
24.3.8.4	Procedure for Stopping Adjustment .....	669
24.4	Interrupt Sources .....	670
24.5	Usage Notes .....	672
24.5.1	Register Writing during Counting .....	672
24.5.2	Use of Periodic Interrupts .....	672
24.5.3	RTCOUT (1-Hz/64-Hz) Clock Output .....	672
24.5.4	Transitions to Low Power Consumption Modes after Setting Registers .....	673
24.5.5	Notes When Writing to and Reading from Registers .....	673

24.5.6	Changing the Count Mode .....	673
24.5.7	Initialization Procedure When the Realtime Clock is Not to be Used .....	674
25.	Independent Watchdog Timer (IWDtA) .....	675
25.1	Overview .....	675
25.2	Register Descriptions .....	677
25.2.1	IWDt Refresh Register (IWDTRR) .....	677
25.2.2	IWDt Control Register (IWDTCR) .....	678
25.2.3	IWDt Status Register (IWDTSR) .....	681
25.2.4	IWDt Reset Control Register (IWDTRCR) .....	682
25.2.5	IWDt Count Stop Control Register (IWDTCSTPR) .....	683
25.2.6	Option Function Select Register 0 (OFS0) .....	683
25.3	Operation .....	684
25.3.1	Count Operation in Each Start Mode .....	684
25.3.1.1	Register Start Mode .....	684
25.3.1.2	Auto-Start Mode .....	686
25.3.2	Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers .....	688
25.3.3	Refresh Operation .....	689
25.3.4	Status Flags .....	691
25.3.5	Reset Output .....	691
25.3.6	Interrupt Sources .....	691
25.3.7	Reading the Counter Value .....	692
25.3.8	Correspondence between Option Function Select Register 0 (OFS0) and IWDt Registers ...	693
25.4	Usage Notes .....	693
25.4.1	Refresh Operations .....	693
25.4.2	Clock Division Ratio Setting .....	693
26.	USB 2.0 Host/Function Module (USBC) .....	694
26.1	Overview .....	694
26.2	Register Descriptions .....	696
26.2.1	System Configuration Control Register (SYSCFG) .....	696
26.2.2	System Configuration Status Register 0 (SYSSTS0) .....	698
26.2.3	Device State Control Register 0 (DVSTCTR0) .....	699
26.2.4	CFIFO Port Register (CFIFO) D0FIFO Port Register (D0FIFO) D1FIFO Port Register (D1FIFO) .....	702
26.2.5	CFIFO Port Select Register (CFIFOSEL) D0FIFO Port Select Register (D0FIFOSEL) D1FIFO Port Select Register (D1FIFOSEL) .....	704
26.2.6	CFIFO Port Control Register (CFIFOCTR) D0FIFO Port Control Register (D0FIFOCTR) D1FIFO Port Control Register (D1FIFOCTR) .....	708
26.2.7	Interrupt Enable Register 0 (INTENB0) .....	710
26.2.8	Interrupt Enable Register 1 (INTENB1) .....	711
26.2.9	BRDY Interrupt Enable Register (BRDYENB) .....	712



26.2.10	NRDY Interrupt Enable Register (NRDYENB) .....	713
26.2.11	BEMP Interrupt Enable Register (BEMPENB) .....	714
26.2.12	SOF Output Configuration Register (SOFCFG) .....	715
26.2.13	Interrupt Status Register 0 (INTSTS0) .....	716
26.2.14	Interrupt Status Register 1 (INTSTS1) .....	719
26.2.15	BRDY Interrupt Status Register (BRDYSTS) .....	722
26.2.16	NRDY Interrupt Status Register (NRDYSTS) .....	723
26.2.17	BEMP Interrupt Status Register (BEMPSTS) .....	724
26.2.18	Frame Number Register (FRMNUM) .....	725
26.2.19	USB Request Type Register (USBREQ) .....	726
26.2.20	USB Request Value Register (USBVAL) .....	727
26.2.21	USB Request Index Register (USBINDX) .....	728
26.2.22	USB Request Length Register (USBLENG) .....	729
26.2.23	DCP Configuration Register (DCPCFG) .....	730
26.2.24	DCP Maximum Packet Size Register (DCPMAXP) .....	731
26.2.25	DCP Control Register (DCPCTR) .....	732
26.2.26	Pipe Window Select Register (PIPESEL) .....	735
26.2.27	Pipe Configuration Register (PIPECFG) .....	736
26.2.28	Pipe Maximum Packet Size Register (PIPEMAXP) .....	738
26.2.29	Pipe Cycle Control Register (PIPEPERI) .....	739
26.2.30	PIPE <sub>n</sub> Control Registers (PIPE <sub>n</sub> CTR) (n = 1 to 9) .....	740
26.2.31	PIPE <sub>n</sub> Transaction Counter Enable Register (PIPE <sub>n</sub> TRE) (n = 1 to 5) .....	748
26.2.32	PIPE <sub>n</sub> Transaction Counter Register (PIPE <sub>n</sub> TRN) (n = 1 to 5) .....	749
26.2.33	Device Address n Configuration Register (DEVADD <sub>n</sub> ) (n = 0 to 5) .....	750
26.2.34	USB Module Control Register (USBMC) .....	751
26.2.35	BC Control Register 0 (USBBCCTRL0) .....	752
26.3	Operation .....	754
26.3.1	System Control .....	754
26.3.1.1	Setting Data to the USB Related Register .....	754
26.3.1.2	Controller Function Selection .....	754
26.3.1.3	Controlling USB Data Bus Resistors .....	754
26.3.1.4	Example of USB External Connection Circuit .....	755
26.3.2	Interrupt Sources .....	760
26.3.3	Interrupt Descriptions .....	762
26.3.3.1	BRDY Interrupt .....	762
26.3.3.2	NRDY Interrupt .....	766
26.3.3.3	BEMP Interrupt .....	769
26.3.3.4	Device State Transition Interrupt .....	770
26.3.3.5	Control Transfer Stage Transition Interrupt .....	771
26.3.3.6	Frame Update Interrupt .....	772
26.3.3.7	VBUS Interrupt .....	772

26.3.3.8	Resume Interrupt .....	772
26.3.3.9	OVRCR Interrupt .....	772
26.3.3.10	BCHG Interrupt .....	772
26.3.3.11	DTCH Interrupt .....	772
26.3.3.12	SACK Interrupt .....	773
26.3.3.13	SIGN Interrupt .....	773
26.3.3.14	ATTCH Interrupt .....	773
26.3.3.15	EOFERR Interrupt .....	773
26.3.3.16	Portable Device Detection Interrupt .....	773
26.3.4	Pipe Control .....	774
26.3.4.1	Pipe Control Register Switching Procedures .....	775
26.3.4.2	Transfer Types .....	775
26.3.4.3	Endpoint Number .....	775
26.3.4.4	Maximum Packet Size Setting .....	776
26.3.4.5	Transaction Counter (For PIPE1 to PIPE5 in Reading Direction) .....	776
26.3.4.6	Response PID .....	777
26.3.4.7	Data PID Sequence Bit .....	778
26.3.4.8	Response PID = NAK Function .....	778
26.3.4.9	Auto Response Mode .....	778
26.3.4.10	OUT-NAK Mode .....	778
26.3.4.11	Null Auto Response Mode .....	779
26.3.5	FIFO Buffer Memory .....	779
26.3.5.1	FIFO Buffer Memory .....	779
26.3.5.2	FIFO Buffer Clearing .....	780
26.3.5.3	FIFO Port Functions .....	781
26.3.6	Control Transfers Using DCP .....	782
26.3.6.1	Control Transfers When the Host Controller is Selected .....	782
26.3.6.2	Control Transfers When the Function Controller is Selected .....	783
26.3.7	Bulk Transfers (PIPE1 to PIPE5) .....	784
26.3.8	Interrupt Transfers (PIPE6 to PIPE9) .....	784
26.3.8.1	Interval Counter during Interrupt Transfers When the Host Controller is Selected .....	784
26.3.9	Isochronous Transfers (PIPE1 and PIPE2) .....	785
26.3.9.1	Error Detection in Isochronous Transfers .....	785
26.3.9.2	DATA-PID .....	786
26.3.9.3	Interval Counter .....	786
26.3.10	SOF Interpolation Function .....	793
26.3.11	Pipe Schedule .....	794
26.3.11.1	Conditions for Generating a Transaction .....	794
26.3.11.2	Transfer Schedule .....	794
26.3.11.3	Enabling USB Communication .....	794
26.4	Usage Notes .....	795

26.4.1	Setting the Module Stop Function .....	795
26.5	Battery Charging Detection Processing .....	796
26.5.1	Processing When Function Controller is Selected .....	796
26.5.2	Processing When Host Controller is Selected .....	798
27.	Serial Communications Interface (SCle, SCIf) .....	801
27.1	Overview .....	801
27.2	Register Descriptions .....	811
27.2.1	Receive Shift Register (RSR) .....	811
27.2.2	Receive Data Register (RDR) .....	811
27.2.3	Transmit Data Register (TDR) .....	811
27.2.4	Transmit Shift Register (TSR) .....	811
27.2.5	Serial Mode Register (SMR) .....	812
27.2.6	Serial Control Register (SCR) .....	816
27.2.7	Serial Status Register (SSR) .....	821
27.2.8	Smart Card Mode Register (SCMR) .....	825
27.2.9	Bit Rate Register (BRR) .....	827
27.2.10	Serial Extended Mode Register (SEMR) .....	834
27.2.11	Noise Filter Setting Register (SNFR) .....	836
27.2.12	I <sup>2</sup> C Mode Register 1 (SIMR1) .....	837
27.2.13	I <sup>2</sup> C Mode Register 2 (SIMR2) .....	838
27.2.14	I <sup>2</sup> C Mode Register 3 (SIMR3) .....	839
27.2.15	I <sup>2</sup> C Status Register (SISR) .....	841
27.2.16	SPI Mode Register (SPMR) .....	842
27.2.17	Extended Serial Module Enable Register (ESMER) .....	843
27.2.18	Control Register 0 (CR0) .....	844
27.2.19	Control Register 1 (CR1) .....	844
27.2.20	Control Register 2 (CR2) .....	845
27.2.21	Control Register 3 (CR3) .....	846
27.2.22	Port Control Register (PCR) .....	846
27.2.23	Interrupt Control Register (ICR) .....	847
27.2.24	Status Register (STR) .....	848
27.2.25	Status Clear Register (STCR) .....	849
27.2.26	Control Field 0 Data Register (CF0DR) .....	849
27.2.27	Control Field 0 Compare Enable Register (CF0CR) .....	850
27.2.28	Control Field 0 Receive Data Register (CF0RR) .....	850
27.2.29	Primary Control Field 1 Data Register (PCF1DR) .....	850
27.2.30	Secondary Control Field 1 Data Register (SCF1DR) .....	851
27.2.31	Control Field 1 Compare Enable Register (CF1CR) .....	851
27.2.32	Control Field 1 Receive Data Register (CF1RR) .....	851
27.2.33	Timer Control Register (TCR) .....	852
27.2.34	Timer Mode Register (TMR) .....	852

27.2.35	Timer Prescaler Register (TPRE) .....	853
27.2.36	Timer Count Register (TCNT) .....	853
27.3	Operation in Asynchronous Mode .....	854
27.3.1	Serial Data Transfer Format .....	854
27.3.2	Receive Data Sampling Timing and Reception Margin in Asynchronous Mode .....	856
27.3.3	Clock .....	857
27.3.4	CTS and RTS Functions .....	857
27.3.5	SCI Initialization (Asynchronous Mode) .....	858
27.3.6	Serial Data Transmission (Asynchronous Mode) .....	859
27.3.7	Serial Data Reception (Asynchronous Mode) .....	862
27.4	Multi-Processor Communications Function .....	866
27.4.1	Multi-Processor Serial Data Transmission .....	867
27.4.2	Multi-Processor Serial Data Reception .....	868
27.5	Operation in Clock Synchronous Mode .....	871
27.5.1	Clock .....	871
27.5.2	CTS and RTS Functions .....	872
27.5.3	SCI Initialization (Clock Synchronous Mode) .....	873
27.5.4	Serial Data Transmission (Clock Synchronous Mode) .....	874
27.5.5	Serial Data Reception (Clock Synchronous Mode) .....	878
27.5.6	Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode) .....	881
27.6	Operation in Smart Card Interface Mode .....	882
27.6.1	Sample Connection .....	882
27.6.2	Data Format (Except in Block Transfer Mode) .....	883
27.6.3	Block Transfer Mode .....	884
27.6.4	Receive Data Sampling Timing and Reception Margin .....	885
27.6.5	SCI Initialization (Smart Card Interface Mode) .....	886
27.6.6	Serial Data Transmission (Except in Block Transfer Mode) .....	887
27.6.7	Serial Data Reception (Except in Block Transfer Mode) .....	890
27.6.8	Clock Output Control .....	892
27.7	Operation in Simple I <sup>2</sup> C Mode .....	893
27.7.1	Generation of Start, Restart, and Stop Conditions .....	894
27.7.2	Clock Synchronization .....	896
27.7.3	SSDA Output Delay .....	897
27.7.4	SCI Initialization (Simple I <sup>2</sup> C Mode) .....	898
27.7.5	Operation in Master Transmission (Simple I <sup>2</sup> C Mode) .....	899
27.7.6	Master Reception (Simple I <sup>2</sup> C Mode) .....	901
27.8	Operation in Simple SPI Mode .....	903
27.8.1	States of Pins in Master and Slave Modes .....	904
27.8.2	SS Function in Master Mode .....	904
27.8.3	SS Function in Slave Mode .....	904
27.8.4	Relationship between Clock and Transmit/Receive Data .....	905

27.8.5	SCI Initialization (Simple SPI Mode)	906
27.8.6	Transmission and Reception of Serial Data (Simple SPI Mode)	906
27.9	Extended Serial Mode Control Section: Description of Operation	907
27.9.1	Serial Transfer Protocol	907
27.9.2	Transmitting a Start Frame	908
27.9.3	Receiving a Start Frame	911
27.9.3.1	Priority Interrupt Bit	916
27.9.4	Detection of Bus Collisions	917
27.9.5	Digital Filter for Input on the RXDX12 Pin	918
27.9.6	Bit Rate Measurement	919
27.9.7	Selectable Timing for Sampling Data Received through RXDX12	920
27.9.8	Timer	921
27.10	Noise Cancellation Function	923
27.11	Interrupt Sources	924
27.11.1	Buffer Operations for TXI and RXI Interrupts	924
27.11.2	Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode	924
27.11.3	Interrupts in Smart Card Interface Mode	925
27.11.4	Interrupts in Simple I <sup>2</sup> C Mode	926
27.11.5	Interrupt Requests from the Extended Serial Mode Control Section	927
27.12	Event Linking	928
27.13	Usage Notes	929
27.13.1	Setting the Module Stop Function	929
27.13.2	Break Detection and Processing	929
27.13.3	Mark State and Production of Breaks	929
27.13.4	Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)	929
27.13.5	Writing Data to TDR	929
27.13.6	Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)	930
27.13.7	Restrictions on Using DTC	931
27.13.8	Notes on Starting Transfer	931
27.13.9	SCI Operations during Low Power Consumption State	931
27.13.10	External Clock Input in Clock Synchronous Mode and Simple SPI Mode	934
27.13.11	Limitations on Simple SPI Mode	935
27.13.12	Limitation 1 on Usage of the Extended Serial Mode Control Section	935
27.13.13	Limitation 2 on Usage of the Extended Serial Mode Control Section	936
27.13.14	Note on Transmit Enable Bit (TE bit)	937
28.	IrDA Interface	938
28.1	Overview	938
28.2	Register Descriptions	939
28.2.1	IrDA Control Register (IRCR)	939
28.3	Operation	940

28.3.1	Transmission .....	940
28.3.2	Reception .....	941
28.3.3	Selecting High-Level Pulse Width .....	941
28.4	Usage Notes .....	942
28.4.1	Module Stop Function Setting .....	942
28.4.2	SCI5 Setting .....	942
28.4.3	Minimum Pulse-Width during Reception .....	942
28.4.4	Notes on IrDA Initial Setting/Resetting .....	942
29.	I <sup>2</sup> C Bus Interface (IIC) .....	943
29.1	Overview .....	943
29.2	Register Descriptions .....	946
29.2.1	I <sup>2</sup> C Bus Control Register 1 (ICCR1) .....	946
29.2.2	I <sup>2</sup> C Bus Control Register 2 (ICCR2) .....	948
29.2.3	I <sup>2</sup> C Bus Mode Register 1 (ICMR1) .....	952
29.2.4	I <sup>2</sup> C Bus Mode Register 2 (ICMR2) .....	953
29.2.5	I <sup>2</sup> C Bus Mode Register 3 (ICMR3) .....	955
29.2.6	I <sup>2</sup> C Bus Function Enable Register (ICFER) .....	957
29.2.7	I <sup>2</sup> C Bus Status Enable Register (ICSER) .....	959
29.2.8	I <sup>2</sup> C Bus Interrupt Enable Register (ICIER) .....	961
29.2.9	I <sup>2</sup> C Bus Status Register 1 (ICSR1) .....	963
29.2.10	I <sup>2</sup> C Bus Status Register 2 (ICSR2) .....	966
29.2.11	Slave Address Register Ly (SARLy) (y = 0 to 2) .....	969
29.2.12	Slave Address Register Uy (SARUy) (y = 0 to 2) .....	970
29.2.13	I <sup>2</sup> C Bus Bit Rate Low-Level Register (ICBRL) .....	971
29.2.14	I <sup>2</sup> C Bus Bit Rate High-Level Register (ICBRH) .....	972
29.2.15	I <sup>2</sup> C Bus Transmit Data Register (ICDRT) .....	974
29.2.16	I <sup>2</sup> C Bus Receive Data Register (ICDRR) .....	974
29.2.17	I <sup>2</sup> C Bus Shift Register (ICDRS) .....	974
29.2.18	Timeout Internal Counter (TMOCNT) .....	975
29.3	Operation .....	976
29.3.1	Communication Data Format .....	976
29.3.2	Initial Settings .....	977
29.3.3	Master Transmit Operation .....	978
29.3.4	Master Receive Operation .....	981
29.3.5	Slave Transmit Operation .....	987
29.3.6	Slave Receive Operation .....	990
29.4	SCL Synchronization Circuit .....	992
29.5	SDA Output Delay Function .....	993
29.6	Digital Noise Filter Circuits .....	994
29.7	Address Match Detection .....	995
29.7.1	Slave-Address Match Detection .....	995

29.7.2	Detection of the General Call Address .....	997
29.7.3	Device-ID Address Detection .....	998
29.7.4	Host Address Detection .....	1000
29.8	Automatic Low-Hold Function for SCL .....	1001
29.8.1	Function to Prevent Wrong Transmission of Transmit Data .....	1001
29.8.2	NACK Reception Transfer Suspension Function .....	1002
29.8.3	Function to Prevent Failure to Receive Data .....	1002
29.9	Arbitration-Lost Detection Functions .....	1004
29.9.1	Master Arbitration-Lost Detection (MALE Bit) .....	1004
29.9.2	Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit) .....	1006
29.9.3	Slave Arbitration-Lost Detection (SALE Bit) .....	1007
29.10	Start Condition/Restart Condition/Stop Condition Issuing Function .....	1008
29.10.1	Issuing a Start Condition .....	1008
29.10.2	Issuing a Restart Condition .....	1008
29.10.3	Issuing a Stop Condition .....	1009
29.11	Bus Hanging .....	1010
29.11.1	Timeout Function .....	1010
29.11.2	Extra SCL Clock Cycle Output Function .....	1012
29.11.3	RIIC Reset and Internal Reset .....	1013
29.12	SMBus Operation .....	1014
29.12.1	SMBus Timeout Measurement .....	1014
29.12.2	Packet Error Code (PEC) .....	1015
29.12.3	SMBus Host Notification Protocol (Notify ARP Master Command) .....	1015
29.13	Interrupt Sources .....	1016
29.13.1	Buffer Operation for TXI and RXI Interrupts .....	1016
29.14	Resets and Register and Function States When Issuing Each Condition .....	1017
29.15	Event Link Output .....	1018
29.15.1	Interrupt Handling and Event Linking .....	1018
29.16	Usage Notes .....	1019
29.16.1	Setting Module Stop Function .....	1019
29.16.2	Notes on Starting Transfer .....	1019
30.	Serial Sound Interface (SSI) .....	1020
30.1	Overview .....	1020
30.2	Register Descriptions .....	1022
30.2.1	Control Register (SSICR) .....	1022
30.2.2	Status Register (SSISR) .....	1026
30.2.3	FIFO Control Register (SSIFCR) .....	1028
30.2.4	FIFO Status Register (SSIFSR) .....	1030
30.2.5	Transmit FIFO Data Register (SSIFTDR) .....	1032
30.2.6	Receive FIFO Data Register (SSIFRDR) .....	1032
30.2.7	TDM Mode Register (SSITDMR) .....	1033

30.3	Operation .....	1034
30.3.1	Bus Format .....	1034
30.3.2	Non-Compressed Mode .....	1034
30.3.3	WS Continue Mode .....	1040
30.3.4	Operating States .....	1041
30.3.5	Transmit Operation .....	1042
30.3.6	Receive Operation .....	1045
30.3.7	Serial Bit Clock Control .....	1047
30.4	Interrupt Sources .....	1047
30.5	Usage Notes .....	1048
30.5.1	Setting the Module Stop Function .....	1048
30.5.2	Notes on Changing Transfer Modes .....	1048
30.5.3	Limits on WS Continue Mode .....	1048
31.	Serial Peripheral Interface (RSPI) .....	1049
31.1	Overview .....	1049
31.2	Register Descriptions .....	1053
31.2.1	RSPI Control Register (SPCR) .....	1053
31.2.2	RSPI Slave Select Polarity Register (SSLP) .....	1055
31.2.3	RSPI Pin Control Register (SPPCR) .....	1056
31.2.4	RSPI Status Register (SPSR) .....	1057
31.2.5	RSPI Data Register (SPDR) .....	1059
31.2.6	RSPI Sequence Control Register (SPSCR) .....	1062
31.2.7	RSPI Sequence Status Register (SPSSR) .....	1063
31.2.8	RSPI Bit Rate Register (SPBR) .....	1064
31.2.9	RSPI Data Control Register (SPDCR) .....	1065
31.2.10	RSPI Clock Delay Register (SPCKD) .....	1067
31.2.11	RSPI Slave Select Negation Delay Register (SSLND) .....	1068
31.2.12	RSPI Next-Access Delay Register (SPND) .....	1069
31.2.13	RSPI Control Register 2 (SPCR2) .....	1070
31.2.14	RSPI Command Registers 0 to 7 (SPCMD0 to SPCMD7) .....	1071
31.3	Operation .....	1074
31.3.1	Overview of RSPI Operations .....	1074
31.3.2	Controlling RSPI Pins .....	1075
31.3.3	RSPI System Configuration Examples .....	1076
31.3.3.1	Single Master/Single Slave (with This MCU Acting as Master) .....	1076
31.3.3.2	Single Master/Single Slave (with This MCU Acting as Slave) .....	1077
31.3.3.3	Single Master/Multi-Slave (with This MCU Acting as Master) .....	1078
31.3.3.4	Single Master/Multi-Slave (with This MCU Acting as Slave) .....	1079
31.3.3.5	Multi-Master/Multi-Slave (with This MCU Acting as Master) .....	1080
31.3.3.6	Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master) .....	1081



31.3.3.7	Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave) .....	1081
31.3.4	Data Format .....	1082
31.3.4.1	When Parity is Disabled (SPCR2.SPPE = 0) .....	1083
31.3.4.2	When Parity is Enabled (SPCR2.SPPE = 1) .....	1087
31.3.5	Transfer Format .....	1091
31.3.5.1	CPHA = 0 .....	1091
31.3.5.2	CPHA = 1 .....	1092
31.3.6	Communications Operating Mode .....	1093
31.3.6.1	Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0) .....	1093
31.3.6.2	Transmit Operations Only (SPCR.TXMD = 1) .....	1094
31.3.7	Transmit Buffer Empty/Receive Buffer Full Interrupts .....	1095
31.3.8	Error Detection .....	1097
31.3.8.1	Overrun Error .....	1098
31.3.8.2	Parity Error .....	1099
31.3.8.3	Mode Fault Error .....	1100
31.3.9	Initializing RSPI .....	1101
31.3.9.1	Initialization by Clearing the SPE Bit .....	1101
31.3.9.2	System Reset .....	1101
31.3.10	SPI Operation .....	1102
31.3.10.1	Master Mode Operation .....	1102
31.3.10.2	Slave Mode Operation .....	1112
31.3.11	Clock Synchronous Operation .....	1116
31.3.11.1	Master Mode Operation .....	1116
31.3.11.2	Slave Mode Operation .....	1120
31.3.12	Loopback Mode .....	1122
31.3.13	Self-Diagnosis of Parity Bit Function .....	1123
31.3.14	Interrupt Sources .....	1124
31.4	Usage Notes .....	1125
31.4.1	Setting Module Stop Function .....	1125
31.4.2	Note on Low Power Consumption Functions .....	1125
31.4.3	Notes on Starting Transfer .....	1125
32.	CRC Calculator (CRC) .....	1126
32.1	Overview .....	1126
32.2	Register Descriptions .....	1127
32.2.1	CRC Control Register (CRCCR) .....	1127
32.2.2	CRC Data Input Register (CRCDIR) .....	1127
32.2.3	CRC Data Output Register (CRCDOR) .....	1128
32.3	Operation .....	1129
32.4	Usage Notes .....	1132
32.4.1	Module Stop Function Setting .....	1132

32.4.2	Note on Transmission .....	1132
<b>33.</b>	<b>LCD Controller/Driver (LCDC) .....</b>	<b>1133</b>
33.1	Overview .....	1133
33.2	Register Descriptions .....	1136
33.2.1	LCD Mode Register 0 (LCDM0) .....	1136
33.2.2	LCD Mode Register 1 (LCDM1) .....	1137
33.2.3	LCD Clock Control Register 0 (LCDC0) .....	1138
33.2.4	LCD Boost Level Control Register (VLCD) .....	1139
33.3	LCD Display Data Registers .....	1140
33.4	Selection of LCD Display Data Register .....	1143
33.4.1	A-Pattern Area and B-pattern Area Data Display .....	1144
33.4.2	Blinking Display (Alternately Displaying A-Pattern and B-Pattern Area Data) .....	1144
33.5	Setting the LCD Controller/Driver .....	1145
33.6	Operation Stop Procedure .....	1148
33.7	Supplying LCD Drive Voltages VL1, VL2, VL3, and VL4 .....	1149
33.7.1	External Resistance Division Method .....	1149
33.7.2	Internal Voltage Boosting Method .....	1151
33.7.3	Capacitor Split Method .....	1152
33.8	Common and Segment Signals .....	1153
33.9	Display Modes .....	1161
33.9.1	Static Display Example .....	1161
33.9.2	Two-Time-Slice Display Example .....	1164
33.9.3	Three-Time-Slice Display Example .....	1167
33.9.4	Four-Time-Slice Display Example .....	1171
33.9.5	Eight-Time-Slice Display Example .....	1175
33.10	Usage Notes .....	1179
33.10.1	Module Stop Function Setting .....	1179
<b>34.</b>	<b>Capacitive Touch Sensing Unit (CTSU) .....</b>	<b>1180</b>
34.1	Overview .....	1181
34.2	Register Descriptions .....	1182
34.2.1	CTSU Control Register 0 (CTSUCR0) .....	1182
34.2.2	CTSU Control Register 1 (CTSUCR1) .....	1184
34.2.3	CTSU Synchronous Noise Reduction Setting Register (CTSUSDPRS) .....	1185
34.2.4	CTSU Sensor Stabilization Wait Control Register (CTSUSST) .....	1186
34.2.5	CTSU Measurement Channel Register 0 (CTSUMCH0) .....	1187
34.2.6	CTSU Measurement Channel Register 1 (CTSUMCH1) .....	1188
34.2.7	CTSU Channel Enable Control Register 0 (CTSUCHAC0) .....	1188
34.2.8	CTSU Channel Enable Control Register 1 (CTSUCHAC1) .....	1189
34.2.9	CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0) .....	1189
34.2.10	CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1) .....	1190
34.2.11	CTSU High-Pass Noise Reduction Control Register (CTSUDCLKC) .....	1191

34.2.12	CTSUSUST	CTSUSUST	1192
34.2.13	CTSUSUSSC	CTSUSUSSC	1194
34.2.14	CTSUSO0	CTSUSO0	1195
34.2.15	CTSUSO1	CTSUSO1	1196
34.2.16	CTSUSC	CTSUSC	1197
34.2.17	CTSURC	CTSURC	1198
34.2.18	CTSUERRS	CTSUERRS	1199
34.3	Operation		1200
34.3.1	Principles of Measurement Operation		1200
34.3.2	Measurement Modes		1202
34.3.2.1	Initial Setting Flowchart		1203
34.3.2.2	Status Counter		1204
34.3.2.3	Self-Capacitance Single Scan Mode Operation		1205
34.3.2.4	Self-Capacitance Multi-Scan Mode Operation		1207
34.3.2.5	Mutual Capacitance Simple Scan Mode Operation		1209
34.3.2.6	Mutual Capacitance Full Scan Mode Operation		1212
34.3.3	Items Common to Multiple Modes		1215
34.3.3.1	Sensor Stabilization Wait Time and Measurement Time		1215
34.3.3.2	Interrupts		1216
34.4	Usage Notes		1218
34.4.1	Measurement Result Data (CTSUSC and CTSURC Counters)		1218
34.4.2	Software Trigger		1218
34.4.3	External Trigger		1218
34.4.4	Notes on Forcibly Stopping Operation		1218
34.4.5	TSCAP Pin		1219
34.4.6	Notes during Measurement Operation (CTSUCR0.CTSUSTRT Bit = 1)		1219
35.	12-Bit A/D Converter (S12ADb)		1220
35.1	Overview		1220
35.2	Register Descriptions		1224
35.2.1	A/D Data Registers y (ADDRy) (y = 0 to 15, 21)		1224
35.2.2	A/D Data Duplication Register (ADDBLDR)		1226
35.2.3	A/D Temperature Sensor Data Register (ADTSDR)		1227
35.2.4	A/D Internal Reference Voltage Data Register (ADOCDR)		1228
35.2.5	A/D Control Register (ADCSR)		1229
35.2.6	A/D Channel Select Register A (ADANSA)		1232
35.2.7	A/D Channel Select Register A1 (ADANSA1)		1233
35.2.8	A/D Channel Select Register B (ADANSB)		1234
35.2.9	A/D Channel Select Register B1 (ADANSB1)		1235
35.2.10	A/D-Converted Value Addition Mode Select Register (ADADS)		1236
35.2.11	A/D Converted Value Addition Mode Select Register 1 (ADADS1)		1237
35.2.12	A/D-Converted Value Addition Count Select Register (ADADC)		1238

35.2.13	A/D Control Extended Register (ADCER) .....	1239
35.2.14	A/D Start Trigger Select Register (ADSTRGR) .....	1240
35.2.15	A/D Converted Extended Input Control Register (ADEXICR) .....	1242
35.2.16	A/D Sampling State Register n (ADSSTRn) (n = 0 to 7, 21, L, T, O) .....	1243
35.2.17	A/D High-Side Reference Voltage Control Register (ADHVREFCNT) .....	1244
35.3	Operation .....	1245
35.3.1	Scanning Operation .....	1245
35.3.2	Single Scan Mode .....	1246
35.3.2.1	Basic Operation .....	1246
35.3.2.2	A/D Conversion When Temperature Sensor Output is Selected .....	1247
35.3.2.3	A/D Conversion When Internal Reference Voltage is Selected .....	1248
35.3.2.4	A/D Conversion in Double Trigger Mode .....	1249
35.3.3	Continuous Scan Mode .....	1250
35.3.3.1	Basic Operation .....	1250
35.3.4	Group Scan Mode .....	1251
35.3.4.1	Basic Operation .....	1251
35.3.4.2	A/D Conversion in Double Trigger Mode .....	1252
35.3.4.3	Notes on Using Software Trigger .....	1253
35.3.5	Analog Input Sampling and Scan Conversion Time .....	1254
35.3.6	Usage Example of Automatic Register Clearing Function .....	1255
35.3.7	A/D-Converted Value Addition Function .....	1255
35.3.8	Starting A/D Conversion with an Asynchronous Trigger .....	1256
35.3.9	Starting A/D Conversion with Synchronous Trigger from Peripheral Modules .....	1256
35.4	Interrupt Sources .....	1256
35.4.1	Interrupt Request on Completion of Each Scanning Conversion .....	1256
35.5	Event Linkage .....	1257
35.5.1	Event Output to ELC .....	1257
35.5.2	12-Bit A/D Converter Operation by Event from ELC .....	1257
35.5.3	Notes on Event Reception from ELC during 12-Bit A/D Conversion .....	1257
35.6	Selecting Reference Voltage .....	1257
35.6.1	When Selecting Internal Reference Voltage as High-Side Reference Voltage .....	1257
35.7	A/D Conversion Accuracy Definitions .....	1258
35.8	Usage Notes .....	1259
35.8.1	Notes on Reading Data Registers .....	1259
35.8.2	Notes on Stopping A/D Conversion .....	1259
35.8.3	A/D Conversion Restarting Timing and Termination Timing .....	1259
35.8.4	Notes on Scan End Interrupt Handling .....	1259
35.8.5	Module Stop Function Setting .....	1259
35.8.6	Notes on Entering Low Power Consumption States .....	1259
35.8.7	Voltage Relationship between AVCC0 and VCC .....	1260
35.8.8	Notes on Releasing Software Standby Mode .....	1260

35.8.9	Allowable Impedance of Signal Source .....	1260
35.8.10	Influence on Absolute Accuracy .....	1262
35.8.11	Voltage Range of Analog Power Supply Pins .....	1262
35.8.12	Notes on Board Design .....	1262
35.8.13	Notes on Noise Prevention .....	1263
35.8.14	Port Setting When 12-Bit A/D Converter Inputs are Used .....	1263
35.8.15	Sequence of Powering on AVCC0 and VCC .....	1263
<b>36.</b>	<b>12-Bit D/A Converter (R12DAA) .....</b>	<b>1264</b>
36.1	Overview .....	1264
36.2	Register Descriptions .....	1265
36.2.1	D/A Data Register m (DADRm) (m = 0, 1) .....	1265
36.2.2	D/A Control Register (DACR) .....	1266
36.2.3	DADRm Format Select Register (DADPR) .....	1266
36.2.4	D/A A/D Synchronous Start Control Register (DAADSCR) .....	1267
36.2.5	D/A VREF Control Register (DAVREFCR) .....	1268
36.3	Operation .....	1269
36.3.1	Measure against Interference between D/A and A/D Conversion .....	1270
36.3.2	Notes on Using the Internal Reference Voltage as the Reference Voltage .....	1272
36.4	Event Link Operation Setting Procedure .....	1273
36.5	Usage Notes on Event Link Operation .....	1273
36.6	Usage Notes .....	1274
36.6.1	Module Stop Function Setting .....	1274
36.6.2	Operation of the D/A Converter in Module Stop State .....	1274
36.6.3	Operation of the D/A Converter in Software Standby Mode .....	1274
36.6.4	Note on Usage When Measure against Interference between D/A and A/D Conversion is Enabled .....	1274
36.6.5	Voltage Relationship between AVCC0 and VCC .....	1274
<b>37.</b>	<b>Temperature Sensor (TEMPSA) .....</b>	<b>1275</b>
37.1	Overview .....	1275
37.2	Register Descriptions .....	1276
37.2.1	Temperature Sensor Calibration Data Register (TSCDRH, TSCDRL) .....	1276
37.3	Using the Temperature Sensor .....	1277
37.3.1	Before Using the Temperature Sensor .....	1277
37.3.2	Setting the 12-Bit A/D Converter .....	1279
37.3.3	A/D Conversion Result of Temperature Sensor Output .....	1279
<b>38.</b>	<b>Comparator B (CMPBa) .....</b>	<b>1280</b>
38.1	Overview .....	1280
38.2	Register Descriptions .....	1283
38.2.1	Comparator B Control Register 1 (CPBCNT1) .....	1283
38.2.2	Comparator B Control Register 2 (CPBCNT2) .....	1283
38.2.3	Comparator B Flag Register (CPBFLG) .....	1284

38.2.4	Comparator B Interrupt Control Register (CPBINT) .....	1285
38.2.5	Comparator B Filter Select Register (CPBF) .....	1286
38.2.6	Comparator B Mode Select Register (CPBMD) .....	1286
38.2.7	Comparator B Reference Input Voltage Select Register (CPBREF) .....	1287
38.2.8	Comparator B Output Control Register (CPBOCR) .....	1288
38.3	Operation .....	1289
38.3.1	Comparator Bn Digital Filter (n = 0, 1) .....	1293
38.4	Comparator B0 and Comparator B1 Interrupts .....	1294
38.5	Event Link Output .....	1294
38.5.1	Relationship between Interrupt Handling and Event Linking .....	1294
38.5.2	Comparator Bn Output (n = 0, 1) Function .....	1294
38.5.3	Example of Using Comparator B to Exit Software Standby Mode .....	1295
38.6	Usage Note .....	1295
38.6.1	Module Stop Function Setting .....	1295
39.	Data Operation Circuit (DOC) .....	1296
39.1	Overview .....	1296
39.2	Register Descriptions .....	1297
39.2.1	DOC Control Register (DOCR) .....	1297
39.2.2	DOC Data Input Register (DODIR) .....	1298
39.2.3	DOC Data Setting Register (DODSR) .....	1298
39.3	Operation .....	1299
39.3.1	Data Comparison Mode .....	1299
39.3.2	Data Addition Mode .....	1300
39.3.3	Data Subtraction Mode .....	1301
39.4	Interrupt Requests .....	1301
39.5	Event Link Output .....	1302
39.5.1	Interrupt Handling and Event Linking .....	1302
39.6	Usage Note .....	1302
39.6.1	Module Stop Function Setting .....	1302
40.	RAM .....	1303
40.1	Overview .....	1303
40.2	Operation .....	1303
40.2.1	Low Power Consumption Function .....	1303
41.	Flash Memory .....	1304
41.1	Overview .....	1304
41.2	ROM Area and Block Configuration .....	1305
41.3	E2 DataFlash Area and Block Configuration .....	1306
41.4	Register Descriptions .....	1307
41.4.1	E2 DataFlash Control Register (DFLCTL) .....	1307
41.4.2	Flash P/E Mode Entry Register (FENTRYR) .....	1308
41.4.3	Protection Unlock Register (FPR) .....	1309

41.4.4	Protection Unlock Status Register (FPSR) .....	1309
41.4.5	Flash P/E Mode Control Register (FPMCR) .....	1310
41.4.6	Flash Initial Setting Register (FISR) .....	1311
41.4.7	Flash Reset Register (FRESETR) .....	1313
41.4.8	Flash Area Select Register (FASR) .....	1313
41.4.9	Flash Control Register (FCR) .....	1314
41.4.10	Flash Extra Area Control Register (FEXCR) .....	1315
41.4.11	Flash Processing Start Address Register H (FSARH) .....	1316
41.4.12	Flash Processing Start Address Register L (FSARL) .....	1317
41.4.13	Flash Processing End Address Register H (FEARH) .....	1317
41.4.14	Flash Processing End Address Register L (FEARL) .....	1318
41.4.15	Flash Read Buffer Register H (FRBH) .....	1318
41.4.16	Flash Read Buffer Register L (FRBL) .....	1318
41.4.17	Flash Write Buffer Register H (FWBH) .....	1319
41.4.18	Flash Write Buffer Register L (FWBL) .....	1319
41.4.19	Flash Status Register 0 (FSTATR0) .....	1320
41.4.20	Flash Status Register 1 (FSTATR1) .....	1322
41.4.21	Flash Error Address Monitor Register H (FEAMH) .....	1323
41.4.22	Flash Error Address Monitor Register L (FEAML) .....	1323
41.4.23	Flash Start-Up Setting Monitor Register (FSCMR) .....	1324
41.4.24	Flash Access Window Start Address Monitor Register (FAWSMR) .....	1324
41.4.25	Flash Access Window End Address Monitor Register (FAWEMR) .....	1325
41.4.26	Unique ID Register n (UIDRn) (n = 0 to 31) .....	1325
41.5	Start-Up Program Protection .....	1326
41.6	Area Protection .....	1327
41.7	Programming and Erasure .....	1328
41.7.1	Sequencer Modes .....	1328
41.7.1.1	E2 DataFlash Access Disabled Mode .....	1328
41.7.1.2	Read Mode .....	1328
41.7.1.3	P/E Modes .....	1329
41.7.2	Software Commands .....	1329
41.7.3	Software Command Usage .....	1329
41.8	Boot Mode .....	1343
41.8.1	Boot Mode (USB Interface) .....	1344
41.8.1.1	System Configuration in Boot Mode (USB Interface) .....	1344
41.8.2	Boot Mode (SCI) .....	1348
41.8.2.1	System Configuration in Boot Mode (SCI) .....	1348
41.8.2.2	Starting Up in Boot Mode (SCI) .....	1350
41.9	Flash Memory Access Disable Function .....	1351
41.9.1	ID Code .....	1351
41.9.1.1	Boot Mode ID Code Protection .....	1352

41.9.1.2	On-Chip Debugging Emulator ID Code Protection .....	1354
41.10	Communication Protocol .....	1355
41.10.1	State Transition in Boot Mode (SCI) .....	1355
41.10.2	Command and Response Configuration .....	1356
41.10.3	Boot Mode Status Inquiry .....	1357
41.10.4	Inquiry Commands .....	1358
41.10.4.1	Supported Device Inquiry .....	1358
41.10.4.2	Data Area Availability Inquiry .....	1358
41.10.4.3	User Area Information Inquiry .....	1359
41.10.4.4	Data Area Information Inquiry .....	1359
41.10.4.5	Block Information Inquiry .....	1360
41.10.5	Setting Commands .....	1361
41.10.5.1	Device Select .....	1361
41.10.5.2	Operating Frequency Select .....	1362
41.10.5.3	Program/Erase State Transition .....	1363
41.10.6	ID Code Authentication Command .....	1364
41.10.6.1	ID Code Check .....	1364
41.10.6.2	Erase Ready .....	1365
41.10.7	Program/Erase Commands .....	1365
41.10.7.1	User/Data Area Program Preparation .....	1366
41.10.7.2	Program .....	1366
41.10.7.3	Erase Preparation .....	1367
41.10.7.4	Block Erase .....	1367
41.10.8	Read-Check Commands .....	1368
41.10.8.1	Memory Read .....	1368
41.10.8.2	User Area Checksum .....	1369
41.10.8.3	Data Area Checksum .....	1369
41.10.8.4	User Area Blank Check .....	1370
41.10.8.5	Data Area Blank Check .....	1370
41.10.8.6	Access Window Information Program .....	1371
41.10.8.7	Access Window Read .....	1372
41.10.9	Programmer Operation in Boot Mode (SCI) .....	1373
41.10.9.1	Bit Rate Automatic Adjustment Procedure .....	1374
41.10.9.2	Procedure to Receive the MCU Information .....	1375
41.10.9.3	Procedure to Select the Device and Change the Bit Rate .....	1376
41.10.9.4	Transition to the Program/Erase State .....	1377
41.10.9.5	Disable Boot Mode ID Code Protection .....	1378
41.10.9.6	Erase Ready Operation .....	1379
41.10.9.7	Erase the User Area and Data Area .....	1380
41.10.9.8	Program the User Area and Data Area .....	1381
41.10.9.9	Check Data in the User Area .....	1382



41.10.9.10	Check Data in the Data Area .....	1383
41.10.9.11	Set the Access Window in the User Area .....	1384
41.11	Rewriting by Self-Programming .....	1385
41.11.1	Overview .....	1385
41.12	Usage Notes .....	1386
41.13	Usage Notes in Boot Mode .....	1387
<b>42.</b>	<b>Electrical Characteristics .....</b>	<b>1388</b>
42.1	Absolute Maximum Ratings .....	1388
42.2	DC Characteristics .....	1389
42.2.1	Standard I/O Pin Output Characteristics (1) .....	1400
42.2.2	Standard I/O Pin Output Characteristics (2) .....	1402
42.2.3	Standard I/O Pin Output Characteristics (3) .....	1404
42.3	AC Characteristics .....	1406
42.3.1	Clock Timing .....	1406
42.3.2	Reset Timing .....	1410
42.3.3	Timing of Recovery from Low Power Consumption Modes .....	1411
42.3.4	Control Signal Timing .....	1414
42.3.5	Timing of On-Chip Peripheral Modules .....	1415
42.4	USB Characteristics .....	1428
42.5	A/D Conversion Characteristics .....	1430
42.6	D/A Conversion Characteristics .....	1435
42.7	Temperature Sensor Characteristics .....	1438
42.8	Comparator Characteristics .....	1438
42.9	LCD Characteristics .....	1440
42.9.1	External Resistance Division Method .....	1440
42.9.2	Internal Voltage Boosting Method .....	1441
42.9.3	Capacitor Split Method .....	1442
42.10	CTSU Characteristics .....	1443
42.11	Power-On Reset Circuit and Voltage Detection Circuit Characteristics .....	1444
42.12	Oscillation Stop Detection Timing .....	1447
42.13	ROM (Flash Memory for Code Storage) Characteristics .....	1448
42.14	E2 DataFlash Characteristics .....	1449
Appendix 1.	Port States in Each Processing Mode .....	1450
Appendix 2.	Package Dimensions .....	1452
REVISION HISTORY	.....	1455

32 MHz 32-bit RX MCUs, 50 DMIPS, up to 512 Kbytes of flash memory,  
USB 2.0 full-speed host/function/OTG, up to 12 comms channels, serial sound interface,  
LCD controller/driver, capacitive touch sensing unit, 12-bit A/D, 12-bit D/A, RTC

## Features

### ■ 32-bit RX CPU core

- 32 MHz maximum operating frequency  
Capable of 50 DMIPS when operating at 32 MHz
- Accumulator handles 64-bit results (for a single instruction) from 32-bit × 32-bit operations
- Multiplication and division unit handles 32-bit × 32-bit operations (multiplication instructions take one CPU clock cycle)
- Fast interrupt
- CISC Harvard architecture with five-stage pipeline
- Variable-length instruction format, ultra-compact code
- On-chip debugging circuit

### ■ Low power consumption functions

- Operation from a single 1.8 to 3.6 V supply
- Three low power consumption modes
- Supply current  
High-speed operating mode: 0.11 μm/MHz  
Software standby mode: 0.44 μA
- Recovery time from software standby mode: 4.8 μs

### ■ On-chip flash memory for code, no wait states

- Operation at 32 MHz, read cycle of 31.25 ns
- No wait states for reading at full CPU speed
- 128 to 512 Kbyte capacities
- Programmable at 1.8 V
- For instructions and operands

### ■ On-chip data flash memory

- 8 Kbytes  
1,000,000 Erase/Write cycles (typ.)
- BGO (Background Operation)

### ■ On-chip SRAM, no wait states

- 32 and 64 Kbyte capacities

### ■ Data transfer controller (DTC)

- Four transfer modes
- Transfer can be set for each interrupt source.

### ■ Event link controller (ELC)

- Module operation can be initiated by event signals without going through interrupts.
- Link operation between modules is possible while the CPU is sleeping.

### ■ Reset and power supply voltage management

- Six types including Power-On Reset (POR)
- Low voltage detection (LVD) with voltage settings

### ■ Clock functions

- External clock input frequency: Up to 20 MHz
- Main clock oscillator frequency: 1 to 20 MHz
- Sub-clock oscillator frequency: 32.768 kHz
- PLL circuit input: 4 to 8 MHz
- Low-speed on-chip oscillator: 4 MHz
- High-speed on-chip oscillator: 32 MHz ±1% (-20 to 85°C)
- IWDT-dedicated on-chip oscillator: 15 kHz
- USB-dedicated PLL circuit: 6 and 8 MHz
- Generate a dedicated 32.768-kHz clock for the RTC
- On-chip clock frequency accuracy measurement circuit (CAC)

### ■ Realtime clock (RTC)

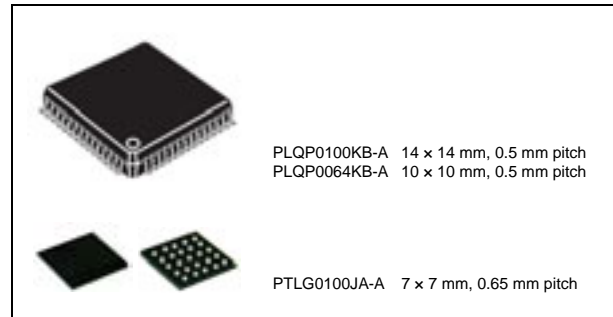
- 30-second, leap year, and error adjustment functions
- Calendar count mode or binary count mode selectable
- Capable of initiating exit from software standby mode

### ■ Independent watchdog timer (IWDT)

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

### ■ On-chip functions for IEC 60730 compliance

- Clock frequency accuracy measurement circuit, IWDT, functions to assist in RAM testing, etc.



### ■ Up to 12 channels for communication

- USB: USB 2.0 host/function/On-The-Go (OTG) (one channel), full-speed = 12 Mbps, low-speed = 1.5 Mbps, isochronous transfer, and BC (Battery Charger) supported
- SCI: Asynchronous mode, clock synchronous mode, smart card interface (up to eight channels)
- IrDA interface (one channel, in cooperation with SCI5)
- I<sup>2</sup>C bus interface: Transfer at up to 400 kbps, capable of SMBus operation (one channel)
- RSPI: Up to 16 Mbps (one channel)
- Serial sound interface (SSI) (one channel)

### ■ Up to 14 extended-function timers

- 16-bit MTU: Input capture/output compare, complementary PWM output, phase counting mode (six channels)
- 8-bit TMR (four channels)
- 16-bit CMT (four channels)

### ■ LCD controller/driver

- Segment signal output × common signal output: 40 × 4, 36 × 8
- On-chip voltage boost circuit, contrast adjustment, and 5-V panel supported
- Blinking function

### ■ 12-bit A/D converter

- Up to 17 channels
- 1.0 μs minimum conversion speed
- Double trigger (data duplication) function for motor control

### ■ 12-bit D/A converter

- Two channels

### ■ Comparator B

- Two channels

### ■ Capacitive touch sensing unit (CTSUS)

- Detection pins: 12 channels (for 100 pins only)
- High-sensitive electrostatic capacitance detection using self-capacitance and mutual capacitance methods
- On-chip noise canceller that enables high tolerance to disturbance noise
- Also supports a mutual capacitance method that allows touch channels to be increased with low pin counts

### ■ Temperature sensor

### ■ General I/O ports

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

### ■ Multi-function pin controller (MPC)

- Multiple I/O pins can be selected for peripheral functions.

### ■ Unique ID

- 16-byte ID code for the MCU

### ■ Operating temperature range

- -40 to +85°C
- -40 to +105°C

# 1. Overview

## 1.1 Outline of Specifications

Table 1.1 lists the specifications, and Table 1.2 gives a comparison of the functions of the products in different packages.

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

**Table 1.1 Outline of Specifications (1/3)**

Classification	Module/Function	Description
CPU	CPU	<ul style="list-style-type: none"> <li>Maximum operating frequency: 32 MHz</li> <li>32-bit RX CPU</li> <li>Minimum instruction execution time: One instruction per clock cycle</li> <li>Address space: 4-Gbyte linear</li> <li>Register set               <ul style="list-style-type: none"> <li>General purpose: Sixteen 32-bit registers</li> <li>Control: Eight 32-bit registers</li> <li>Accumulator: One 64-bit register</li> </ul> </li> <li>Basic instructions: 73</li> <li>DSP instructions: 9</li> <li>Addressing modes: 10</li> <li>Data arrangement               <ul style="list-style-type: none"> <li>Instructions: Little endian</li> <li>Data: Selectable as little endian or big endian</li> </ul> </li> <li>On-chip 32-bit multiplier: 32-bit × 32-bit → 64-bit</li> <li>On-chip divider: 32-bit ÷ 32-bit → 32 bits</li> <li>Barrel shifter: 32 bits</li> </ul>
Memory	ROM	<ul style="list-style-type: none"> <li>Capacity: 128 K /256 K /384 K /512 Kbytes</li> <li>32 MHz, no-wait memory access</li> <li>Programming/erasing method:               <ul style="list-style-type: none"> <li>Serial programming (asynchronous serial communication/USB communication), self-programming</li> </ul> </li> </ul>
	RAM	<ul style="list-style-type: none"> <li>Capacity: 32 K /64 Kbytes</li> <li>32 MHz, no-wait memory access</li> </ul>
	E2 DataFlash	<ul style="list-style-type: none"> <li>Capacity: 8 Kbytes</li> <li>Number of erase/write cycles: 1,000,000 (typ)</li> </ul>
MCU operating mode		Single-chip mode
Clock	Clock generation circuit	<ul style="list-style-type: none"> <li>Main clock oscillator, sub-clock oscillator, low-speed on-chip oscillator, high-speed on-chip oscillator, PLL frequency synthesizer, USB-dedicated PLL frequency synthesizer, and IWDWT-dedicated on-chip oscillator</li> <li>Oscillation stop detection: Available</li> <li>Clock frequency accuracy measurement circuit (CAC)</li> <li>Independent settings for the system clock (ICLK), peripheral module clock (PCLK), and FlashIF clock (FCLK)               <ul style="list-style-type: none"> <li>The CPU and system sections such as other bus masters run in synchronization with the system clock (ICLK): 32 MHz (at max.)</li> <li>Peripheral modules run in synchronization with the PCLK: 32 MHz (at max.)</li> <li>The flash peripheral circuit runs in synchronization with the FCLK: 32 MHz (at max.)</li> </ul> </li> <li>The ICLK frequency can only be set to FCLK, PCLKB, or PCLKD multiplied by n (n: 1, 2, 4, 8, 16, 32, 64).</li> </ul>
Resets		RES# pin reset, power-on reset, voltage monitoring reset, independent watchdog timer reset, and software reset
Voltage detection	Voltage detection circuit (LVDAa)	<ul style="list-style-type: none"> <li>When the voltage on VCC falls below the voltage detection level, an internal reset or internal interrupt is generated.</li> <li>Voltage detection circuit 1 is capable of selecting the detection voltage from 10 levels</li> <li>Voltage detection circuit 2 is capable of selecting the detection voltage from 4 levels</li> </ul>
Low power consumption	Low power consumption functions	<ul style="list-style-type: none"> <li>Module stop function</li> <li>Three low power consumption modes               <ul style="list-style-type: none"> <li>Sleep mode, deep sleep mode, and software standby mode</li> </ul> </li> </ul>
	Function for lower operating power consumption	<ul style="list-style-type: none"> <li>Operating power control modes               <ul style="list-style-type: none"> <li>High-speed operating mode, middle-speed operating mode, and low-speed operating mode</li> </ul> </li> </ul>
Interrupt	Interrupt controller (ICUb)	<ul style="list-style-type: none"> <li>Interrupt vectors: 120</li> <li>External interrupts: 9 (NMI, IRQ0 to IRQ7 pins)</li> <li>Non-maskable interrupts: 4 (NMI pin, voltage monitoring 1 interrupt, voltage monitoring 2 interrupt, and IWDWT interrupt)</li> <li>16 levels specifiable for the order of priority</li> </ul>

**Table 1.1 Outline of Specifications (2/3)**

Classification	Module/Function	Description
DMA	Data transfer controller (DTCa)	<ul style="list-style-type: none"> <li>Transfer modes: Normal transfer, repeat transfer, and block transfer</li> <li>Activation sources: Interrupts</li> <li>Chain transfer function</li> </ul>
I/O ports	General I/O ports	<ul style="list-style-type: none"> <li>100-pin /64-pin</li> <li>I/O: 82/46</li> <li>Input: 2/2</li> <li>Pull-up resistors: 69/38</li> <li>Open-drain outputs: 61/34</li> <li>5-V tolerance: 4/4</li> </ul>
	Event link controller (ELC)	<ul style="list-style-type: none"> <li>Event signals of 44 types can be directly connected to the module</li> <li>Operations of timer modules are selectable at event input</li> <li>Capable of event link operation for port B</li> </ul>
	Multi-function pin controller (MPC)	Capable of selecting the input/output function from multiple pins
Timers	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 six 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 or seven counter-input clock signals for each channel (PCLK/1, PCLK/4, PCLK/16, PCLK/64, PCLK/256, PCLK/1024, 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>Pulse output mode</li> <li>Complementary PWM output mode</li> <li>Reset synchronous PWM mode</li> <li>Phase-counting mode</li> <li>Capable of generating conversion start triggers for the A/D converter</li> </ul>
	Port output enable 2 (POE2a)	Controls the high-impedance state of the MTU's waveform output pins
	Compare match timer (CMT)	<ul style="list-style-type: none"> <li>(16 bits × 2 channels) × 2 units</li> <li>Select from among four clock signals (PCLK/8, PCLK/32, PCLK/128, PCLK/512)</li> </ul>
	Independent watchdog timer (IWDTa)	<ul style="list-style-type: none"> <li>14 bits × 1 channel</li> <li>Count clock: Dedicated low-speed on-chip oscillator for the IWDT</li> <li>Frequency divided by 1, 16, 32, 64, 128, or 256</li> </ul>
	Realtime clock (RTCA)	<ul style="list-style-type: none"> <li>Clock source: Sub-clock</li> <li>Calendar count mode or binary count mode selectable</li> <li>Interrupts: Alarm interrupt, periodic interrupt, and carry interrupt</li> </ul>
	8-bit timer (TMR)	<ul style="list-style-type: none"> <li>(8 bits × 2 channels) × 2 units</li> <li>Seven internal clocks (PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, and PCLK/8192) and an external clock can be selected</li> <li>Pulse output and PWM output with any duty cycle are available</li> <li>Two channels can be cascaded and used as a 16-bit timer</li> </ul>
Communication functions	Serial communications interfaces (SC1e, SC1f)	<ul style="list-style-type: none"> <li>8 channels (channel 0, 1, 2, 5, 6, 8, and 9: SC1e, channel 12: SC1f)</li> <li>Serial communications modes: Asynchronous, clock synchronous, and smart card interface</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 MTU2 timers</li> <li>Simple I<sup>2</sup>C</li> <li>Simple SPI</li> <li>Master/slave mode supported (SC1f only)</li> <li>Start frame and information frame are included (SC1f only)</li> <li>Start-bit detection in asynchronous mode: Low level or falling edge is selectable</li> </ul>
	IrDA interface (IRDA)	<ul style="list-style-type: none"> <li>1 channel (SC15 used)</li> <li>Supports encoding/decoding of waveforms conforming to IrDA standard 1.0</li> </ul>
	I <sup>2</sup> C bus interface (RIIC)	<ul style="list-style-type: none"> <li>1 channel</li> <li>Communications formats: I<sup>2</sup>C bus format/SMBus format</li> <li>Master mode or slave mode selectable</li> <li>Supports fast mode</li> </ul>

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

Classification	Module/Function	Description
Communication functions	Serial peripheral interface (RSPI)	<ul style="list-style-type: none"> <li>• 1 channel</li> <li>• Transfer facility Using the MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPI clock (RSPCK) 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> <li>• Data formats</li> <li>• Choice of LSB-first or MSB-first transfer The number of bits in each transfer can be changed to 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits. 128-bit buffers for transmission and reception Up to four frames can be transmitted or received in a single transfer operation (with each frame having up to 32 bits)</li> <li>• Double buffers for both transmission and reception</li> </ul>
	USB 2.0 host/function module (USBc)	<ul style="list-style-type: none"> <li>• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated.</li> <li>• Host/function module: 1 port</li> <li>• Compliant with USB version 2.0</li> <li>• Transfer speed: Full-speed (12 Mbps), low-speed (1.5 Mbps)</li> <li>• OTG (ON-The-Go) is supported.</li> <li>• Isochronous transfer is supported.</li> <li>• BC (Battery Charger) is supported.</li> </ul>
	Serial Sound Interface (SSI)	<ul style="list-style-type: none"> <li>• 1 channel</li> <li>• Capable of duplex communications</li> <li>• Various serial audio formats supported</li> <li>• Master/slave function supported</li> <li>• Programmable word clock or bit clock generation function</li> <li>• 8/16/18/20/22/24/32-bit data formats supported</li> <li>• On-chip 8-stage FIFO for transmission/reception</li> <li>• Supports WS continue mode in which the SSIWS signal is not stopped.</li> </ul>
LCD controller/driver (LCDC)		<ul style="list-style-type: none"> <li>• Internal voltage boosting method, capacitor split method, and external resistance division method are switchable.</li> <li>• Segment signal output x common signal output: 40 x 4, 36 x 8</li> </ul>
12-bit A/D converter (S12ADb)		<ul style="list-style-type: none"> <li>• 1 unit (1 unit x 17 channels)</li> <li>• 12-bit resolution</li> <li>• Minimum conversion time: 1.0 μs per channel when the ADCLK is operating at 32 MHz</li> <li>• Operating modes Scan mode (single scan mode, continuous scan mode, and group scan mode)</li> <li>• Double trigger mode (duplication of A/D conversion data)</li> <li>• A/D conversion start conditions A software trigger, a trigger from a timer (MTU), an external trigger signal, or ELC</li> </ul>
Temperature sensor (TEMPSA)		<ul style="list-style-type: none"> <li>• 1 channel</li> <li>• The voltage of the temperature is converted into a digital value by the 12-bit A/D converter.</li> </ul>
12-bit D/A converter (R12DAA)		<ul style="list-style-type: none"> <li>• 2 channels</li> <li>• 12-bit resolution</li> <li>• Output voltage: 0.35 to AVCC - 0.47 V</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>
Comparator B (CMPBa)		<ul style="list-style-type: none"> <li>• 2 channels</li> <li>• Function to compare the reference voltage and the analog input voltage</li> <li>• Window comparator operation or standard comparator operation is selectable</li> </ul>
Capacitive touch sensing unit (CTSU)		Detection pin: 12 channels
Data operation circuit (DOC)		Comparison, addition, and subtraction of 16-bit data
Unique ID		16-byte ID code for the MCU
Power supply voltages/Operating frequencies		VCC = 1.8 to 2.4 V: 8 MHz, VCC = 2.4 to 2.7 V: 16 MHz, VCC = 2.7 to 3.6 V: 32 MHz
Supply current		3.6 mA at 32 MHz (typ.)
Operating temperature range		D version: -40 to +85°C, G version: -40 to +105°C
Packages		100-pin LQFP (PLQP0100KB-A) 14 x 14 mm, 0.50 mm pitch 100-pin TFLGA (PTLG0100JA-A) 7 x 7 mm, 0.65 mm pitch 64-pin LQFP (PLQP0064KB-A) 10 x 10 mm, 0.50 mm pitch
On-chip debugging system		E1 emulator (FINE interface)

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

Module/Functions		RX113 Group	
		100 Pins	64 Pins
Interrupts	External interrupts	NMI, IRQ0 to IRQ7	
DMA	Data transfer controller	Supported	
Timers	Multi-function timer pulse unit 2	6 channels (MTU0 to MTU5)	
	Port output enable 2	Supported	
	Compare match timer	2 channels × 2 units	
	Realtime clock	Supported	
	8-bit timer	2 channels × 2 units	
	Independent watchdog timer	Supported	
Communication functions	Serial communications interfaces (SCle) [simple I <sup>2</sup> C, simple SPI]	7 channels (SCI0, 1, 2, 5, 6, 8, 9)	5 channels (SCI1, 5, 6, 8, 9)
	IrDA interface	1 channel (SCI5)	
	Serial communications interface (SCIf) [simple I <sup>2</sup> C, simple SPI]	1 channel (SCI12)	
	I <sup>2</sup> C bus interface	1 channel	
	Serial peripheral interface	1 channel	
	USB 2.0 host/function module (USBc)	1 channel	
	Serial sound interface	1 channel	
12-bit A/D converter (including high-precision channels)	17 channels (9 channels)	11 channels (3 channels)	
Temperature sensor	Supported		
Comparator B	2 channels		
12-bit D/A converter	2 channels		
CRC calculator	Supported		
Event link controller	Supported		
Capacitive touch sensing unit	12 channels	Not supported	
LCD	40 SEG × 4 COM 36 SEG × 8 COM	20 SEG × 4 COM 16 SEG × 8 COM	
Packages	100-pin LFQFP 100-pin TFLGA	64-pin LFQFP	

## 1.2 List of Products

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

**Table 1.3 List of Products**

Group	Part No.	Orderable Part No.	Package	ROM Capacity	RAM Capacity	E2 DataFlash	Maximum Operating Frequency	Operating Temperature
RX113	R5F51138ADFP	R5F51138ADFP#30	PLQP0100KB-A	512 Kbytes	64 Kbytes	8 Kbytes	32MHz	-40 to +85°C
	R5F51138ADFM	R5F51138ADFM#30	PLQP0064KB-A					
	R5F51138ADLJ	R5F51138ADLJ#20	PTLG0100JA-A					
	R5F51137ADFP	R5F51137ADFP#30	PLQP0100KB-A	384 Kbytes				
	R5F51137ADFM	R5F51137ADFM#30	PLQP0064KB-A					
	R5F51137ADLJ	R5F51137ADLJ#20	PTLG0100JA-A					
	R5F51136ADFP	R5F51136ADFP#30	PLQP0100KB-A	256 Kbytes	32 Kbytes			
	R5F51136ADFM	R5F51136ADFM#30	PLQP0064KB-A					
	R5F51136ADLJ	R5F51136ADLJ#20	PTLG0100JA-A					
	R5F51135ADFP	R5F51135ADFP#30	PLQP0100KB-A	128 Kbytes				
	R5F51135ADFM	R5F51135ADFM#30	PLQP0064KB-A					
	R5F51135ADLJ	R5F51135ADLJ#20	PTLG0100JA-A					
	R5F51138AGFP	R5F51138AGFP#30	PLQP0100KB-A*1	512 Kbytes	64 Kbytes	8 Kbytes	32MHz	-40 to +85°C
	R5F51138AGFM	R5F51138AGFM#30	PLQP0064KB-A*1					
	R5F51137AGFP	R5F51137AGFP#30	PLQP0100KB-A*1					
	R5F51137AGFM	R5F51137AGFM#30	PLQP0064KB-A*1	384 Kbytes				
R5F51136AGFP	R5F51136AGFP#30	PLQP0100KB-A*1						
R5F51136AGFM	R5F51136AGFM#30	PLQP0064KB-A*1						
R5F51135AGFP	R5F51135AGFP#30	PLQP0100KB-A*1	256 Kbytes	32 Kbytes				
R5F51135AGFM	R5F51135AGFM#30	PLQP0064KB-A*1						
R5F51135AGFP	R5F51135AGFP#30	PLQP0100KB-A*1			128 Kbytes			
R5F51135AGFM	R5F51135AGFM#30	PLQP0064KB-A*1						

Note 1. Planned

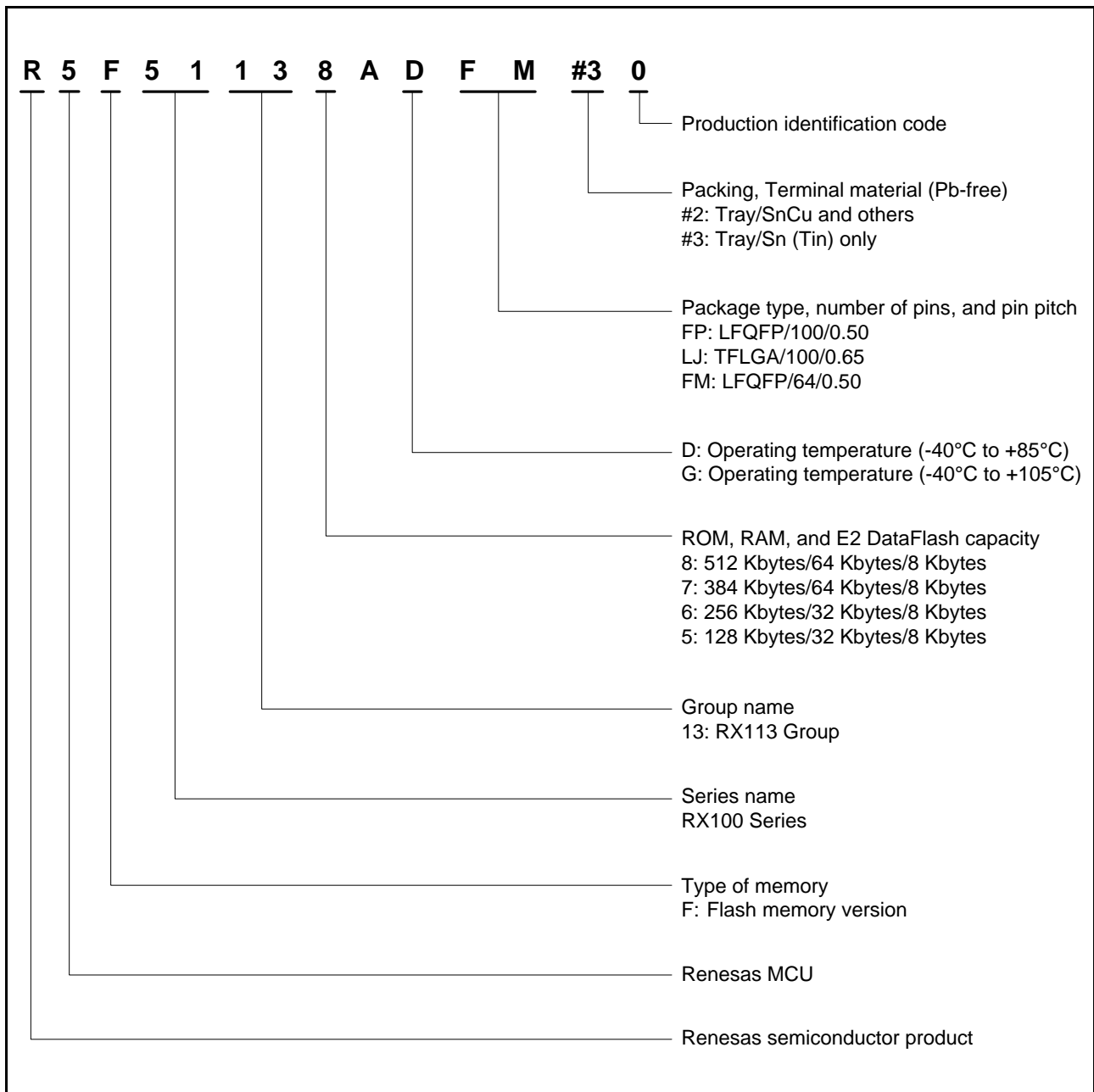


Figure 1.1 How to Read the Product Part No., Memory Capacity, and Package Type



### 1.3 Block Diagram

Figure 1.2 shows a block diagram.

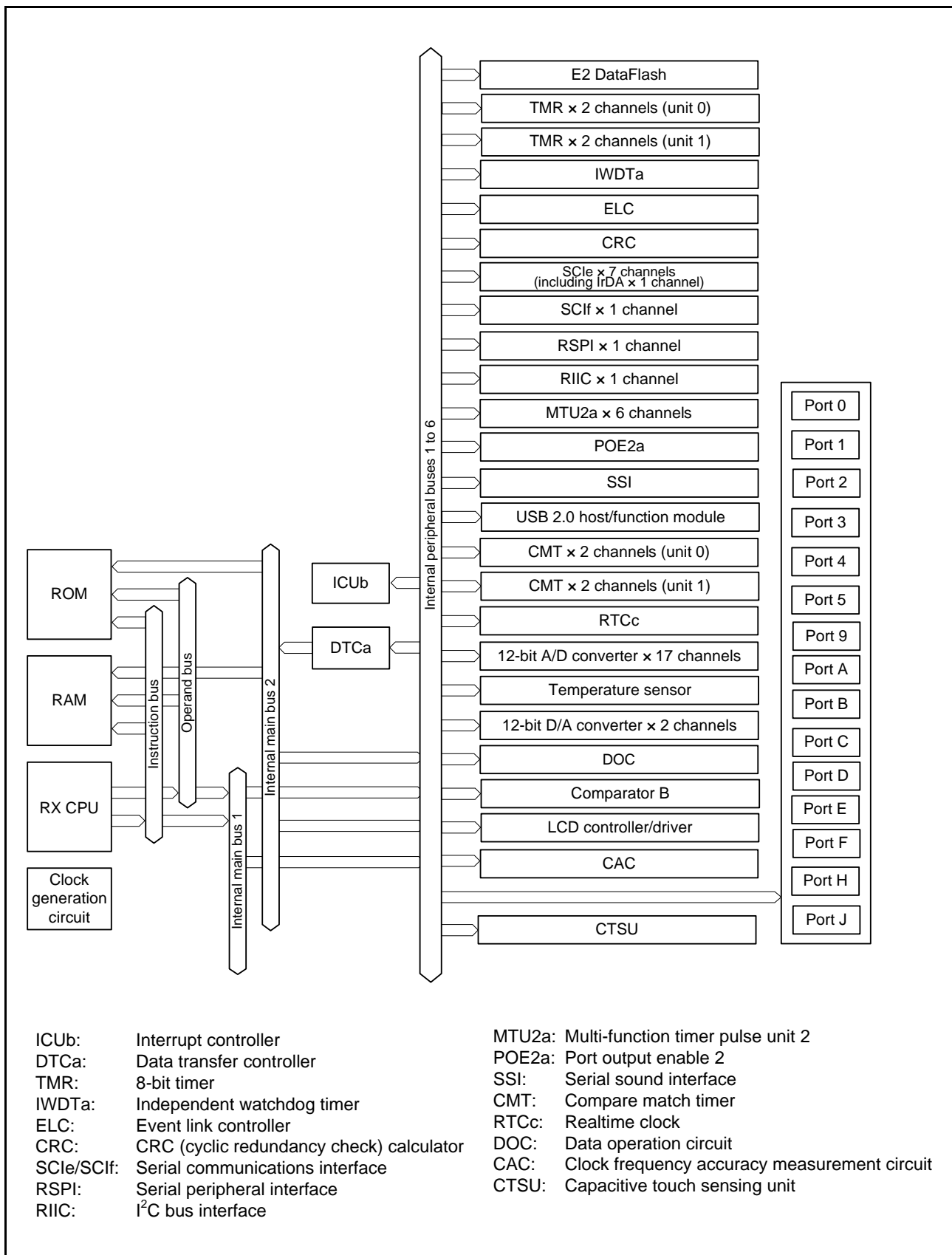


Figure 1.2 Block Diagram

## 1.4 Pin Functions

Table 1.4 lists the pin functions.

**Table 1.4 Pin Functions (1/4)**

Classifications	Pin Name	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply.
	VCL	—	Connect this pin to the VSS pin via the 4.7 $\mu$ F smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
Clock	XTAL	Output/ Input *1	Pins for connecting a crystal. An external clock can be input through the XTAL pin.
	EXTAL	Input	
	XCIN	Input	Input/output pins for the sub-clock oscillator. Connect a crystal between XCIN and XCOU.
	XCOU	Output	
	CLKOUT	Output	Clock output pin.
Operating mode control	MD	Input	Pin for setting the operating mode. The signal levels on this pin must not be changed during operation.
	UB#	Input	Pin used for boot mode (USB interface).
	UPSEL	Input	Pin used for boot mode (USB interface).
System control	RES#	Input	Reset pin. This MCU enters the reset state when this signal goes low.
CAC	CACREF	Input	Input pin for the clock frequency accuracy measurement circuit.
On-chip emulator	FINED	I/O	FINE interface pin.
LVD	CMPA2	Input	Detection target voltage pin for voltage detection 2.
Interrupts	NMI	Input	Non-maskable interrupt request pin.
	IRQ0 to IRQ7	Input	Interrupt request pins.
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/external pulse input pins.
Port output enable 2	MTCLKA, MTCLKB, MTCLKC, MTCLKD	Input	Input pins for the external clock.
	POE0# to POE3#, POE8#	Input	Input pins for request signals to place the MTU pins in the high impedance state.
Realtime clock	RTCOUT	Output	Output pin for the 1-Hz/64-Hz clock.
8-bit timer	TMO0 to TMO3	Output	Compare match output pins.
	TMCI0 to TMCI3	Input	Input pins for the external clock to be input to the counter.
	TMRI0 to TMRI3	Input	Counter reset input pins.

**Table 1.4 Pin Functions (2/4)**

Classifications	Pin Name	I/O	Description
Serial communications interface (SCle)	• Asynchronous mode/clock synchronous mode		
	SCK0, SCK1, SCK2, SCK5, SCK6, SCK8, SCK9	I/O	Input/output pins for the clock.
	RXD0, RXD1, RXD2, RXD5, RXD6, RXD8, RXD9	Input	Input pins for received data.
	TXD0, TXD1, TXD2, TXD5, TXD6, TXD8, TXD9	Output	Output pins for transmitted data.
	CTS0#, CTS1#, CTS2#, CTS5#, CTS6#, CTS8#, CTS9#	Input	Input pins for controlling the start of transmission and reception.
	RTS0#, RTS1#, RTS2#, RTS5#, RTS6#, RTS8#, RTS9#	Output	Output pins for controlling the start of transmission and reception.
	• Simple I <sup>2</sup> C mode		
	SSCL0, SSCL1, SSCL2, SSCL5, SSCL6, SSCL8, SSCL9	I/O	Input/output pins for the I <sup>2</sup> C clock.
	SSDA0, SSDA1, SSDA2, SSDA5, SSDA6, SSDA8, SSDA9	I/O	Input/output pins for the I <sup>2</sup> C data.
	• Simple SPI mode		
	SCK0, SCK1, SCK2, SCK5, SCK6, SCK8, SCK9	I/O	Input/output pins for the clock.
	SMISO0, SMISO1, SMISO2, SMISO5, SMISO6, SMISO8, SMISO9	I/O	Input/output pins for slave transmit data.
	SMOSI0, SMOSI1, SMOSI2, SMOSI5, SMOSI6, SMOSI8, SMOSI9	I/O	Input/output pins for master transmit data.
	SS0#, SS1#, SS2#, SS5#, SS6#, SS8#, SS9#	Input	Chip-select input pins.
IrDA interface	IRTXD5	Output	Data output pin in the IrDA format.
	IRRXD5	Input	Data input pin in the IrDA format.
Serial communications interface (SCIf)	• Asynchronous mode/clock synchronous mode		
	SCK12	I/O	Input/output pin for the clock.
	RXD12	Input	Input pin for receiving data.
	TXD12	Output	Output pin for transmitting 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 transmit data.
	SMOSI12	I/O	Input/output pin for master transmit data.
	SS12#	Input	Chip-select input pin.
	• Extended serial mode		
	RDX12	Input	Input pin for data reception by SCIf.
	TDX12	Output	Output pin for data transmission by SCIf.
	SIOX12	I/O	Input/output pin for data reception or transmission by SCIf.

**Table 1.4 Pin Functions (3/4)**

Classifications	Pin Name	I/O	Description
I <sup>2</sup> C bus interface	SCL0	I/O	Input/output pin for I <sup>2</sup> C bus interface clocks. Bus can be directly driven by the N-channel open drain output.
	SDA0	I/O	Input/output pin for I <sup>2</sup> C bus interface data. Bus can be directly driven by the N-channel open drain output.
Serial peripheral interface	RSPCKA	I/O	Input/output pin for the RSPI clock.
	MOSIA	I/O	Input/output pin for transmitting data from the RSPI master.
	MISOA	I/O	Input/output pin for transmitting data from the RSPI slave.
	SSLA0	I/O	Input/output pin to select the slave for the RSPI.
	SSLA1 to SSLA3	Output	Output pins to select the slave for the RSPI.
Serial sound interface	SSISCK0	I/O	SSI serial bit clock pin.
	SSIWS	I/O	Word selection pin.
	SSITXD0	Output	Serial data output pin.
	SSIRXD0	Input	Serial data input pin.
	AUDIO_MCLK	Input	Master clock pin for audio.
USB 2.0 host/function module	VCC_USB	Input	Power supply pin for USB. Connect this pin to VCC.
	VSS_USB	Input	Ground pin for USB. Connect this pin to VSS.
	USB0_DP	I/O	D+ I/O pin of the USB on-chip transceiver.
	USB0_DM	I/O	D- I/O pin of the USB on-chip transceiver.
	USB0_VBUS	Input	USB cable connection monitor pin.
	USB0_EXICEN	Output	Low-power control signal for the OTG chip.
	USB0_VBUSEN	Output	VBUS (5 V) supply enable signal for the OTG chip.
	USB0_OVRCURA, USB0_OVRCURB	Input	External overcurrent detection pins.
	USB0_ID	Input	Mini-AB connector ID input pin during operation in OTG mode.
12-bit A/D converter	AN000 to AN015, AN021	Input	Input pins for the analog signals to be processed by the A/D converter.
	ADTRG0#	Input	Input pin for the external trigger signals that start the A/D conversion.
12-bit D/A converter	DA0, DA1	Output	Output pins for the analog signals to be processed by the D/A converter.
Comparator B	CMPB0	Input	Input pin for the analog signal to be processed by comparator B0.
	CVREFB0	Input	Analog reference voltage supply pin for comparator B0.
	CMPB1	Input	Input pin for the analog signal to be processed by comparator B1.
	CVREFB1	Input	Analog reference voltage supply pin for comparator B1.
	CMPOB0	Output	Output pin for comparator B0.
	CMPOB1	Output	Output pin for comparator B1.
LCD	VL1, VL2, VL3, VL4	I/O	Voltage pin for driving the LCD.
	CAPH, CAPL	I/O	Capacitor connection pin for the LCD controller/driver.
	COM0 to COM7	Output	Common signal output pins for the LCD controller/driver.
	SEG00 to SEG39	Output	Segment signal output pins for the LCD controller/driver.
CTSU	TS0 to TS11	Input	Capacitive touch detection pins (touch pins).
	TSCAP	I/O	Secondary power supply pin for the touch driver.

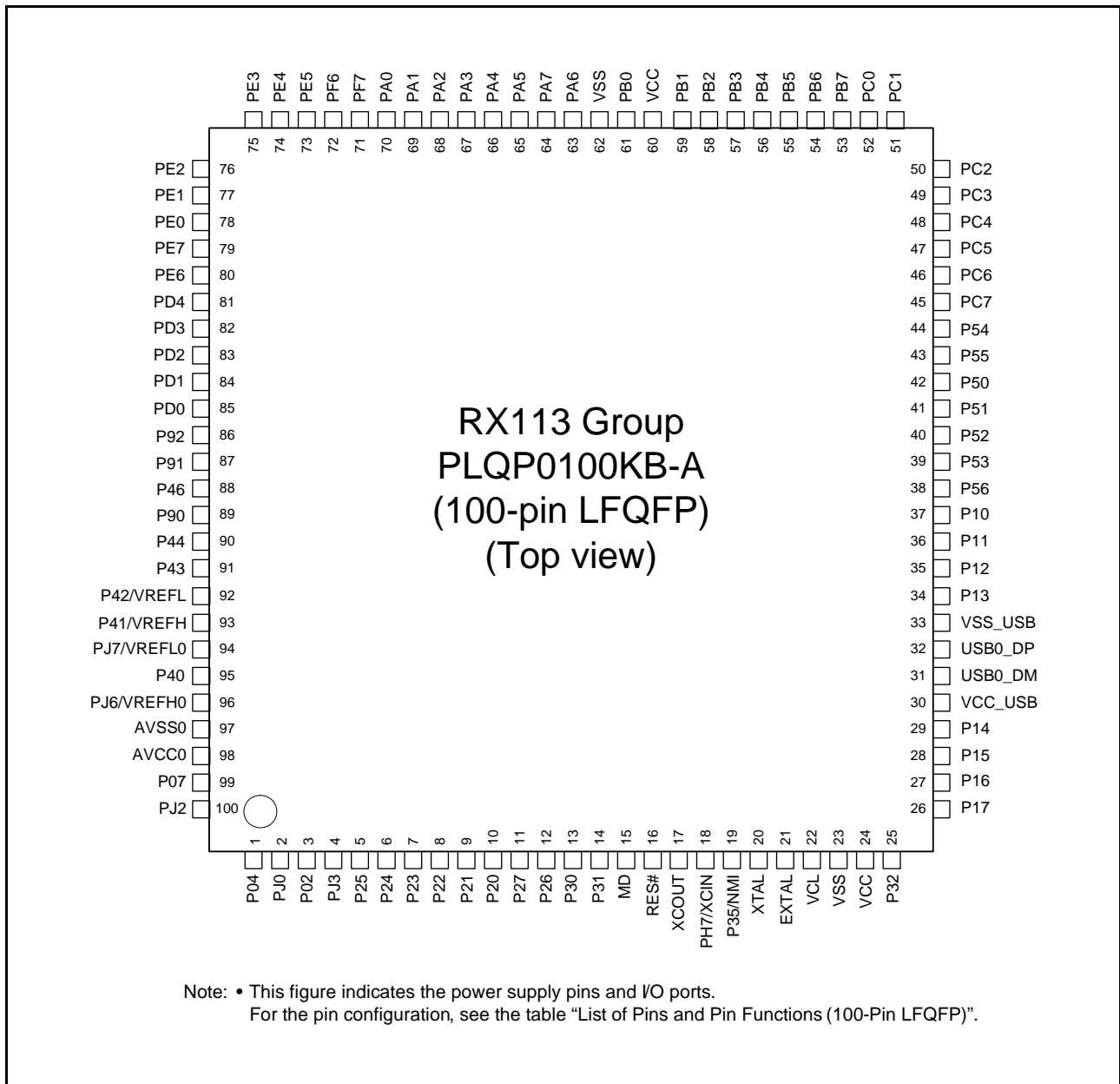
**Table 1.4 Pin Functions (4/4)**

Classifications	Pin Name	I/O	Description
Analog power supply	AVCC0	Input	Analog voltage supply pin for the 12-bit A/D converter and D/A converter. Connect this pin to VCC when not using the 12-bit A/D converter and D/A converter.
	AVSS0	Input	Analog ground pin for the 12-bit A/D converter and D/A converter. Connect this pin to VSS when not using the 12-bit A/D converter and D/A converter.
	VREFH0	Input	Analog reference voltage supply pin for the 12-bit A/D converter.
	VREFL0	Input	Analog reference ground pin for the 12-bit A/D converter.
	VREFH	Input	Analog reference voltage supply pin for the 12-bit D/A converter.
	VREFL	Input	Analog reference ground pin for the 12-bit D/A converter.
I/O ports	P02, P04, P07	I/O	3-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 P32, P35	I/O	4-bit input/output pins (P35 input pin).
	P40 to P44, P46	I/O	6-bit input/output pins.
	P50 to P56	I/O	7-bit input/output pins.
	P90 to P92	I/O	3-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 PD4	I/O	5-bit input/output pins.
	PE0 to PE7	I/O	8-bit input/output pins.
	PF6, PF7	I/O	2-bit input/output pins.
	PH7	Input	1-bit input pin.
	PJ0, PJ2, PJ3, PJ6, PJ7	I/O	5-bit input/output pins.

Note 1. For external clock input.

### 1.5 Pin Assignments

Figure 1.3 to Figure 1.5 show the pin assignments. Table 1.5 to Table 1.7 show the lists of pins and pin functions.



**Figure 1.3 Pin Assignments of the 100-Pin LQFP**

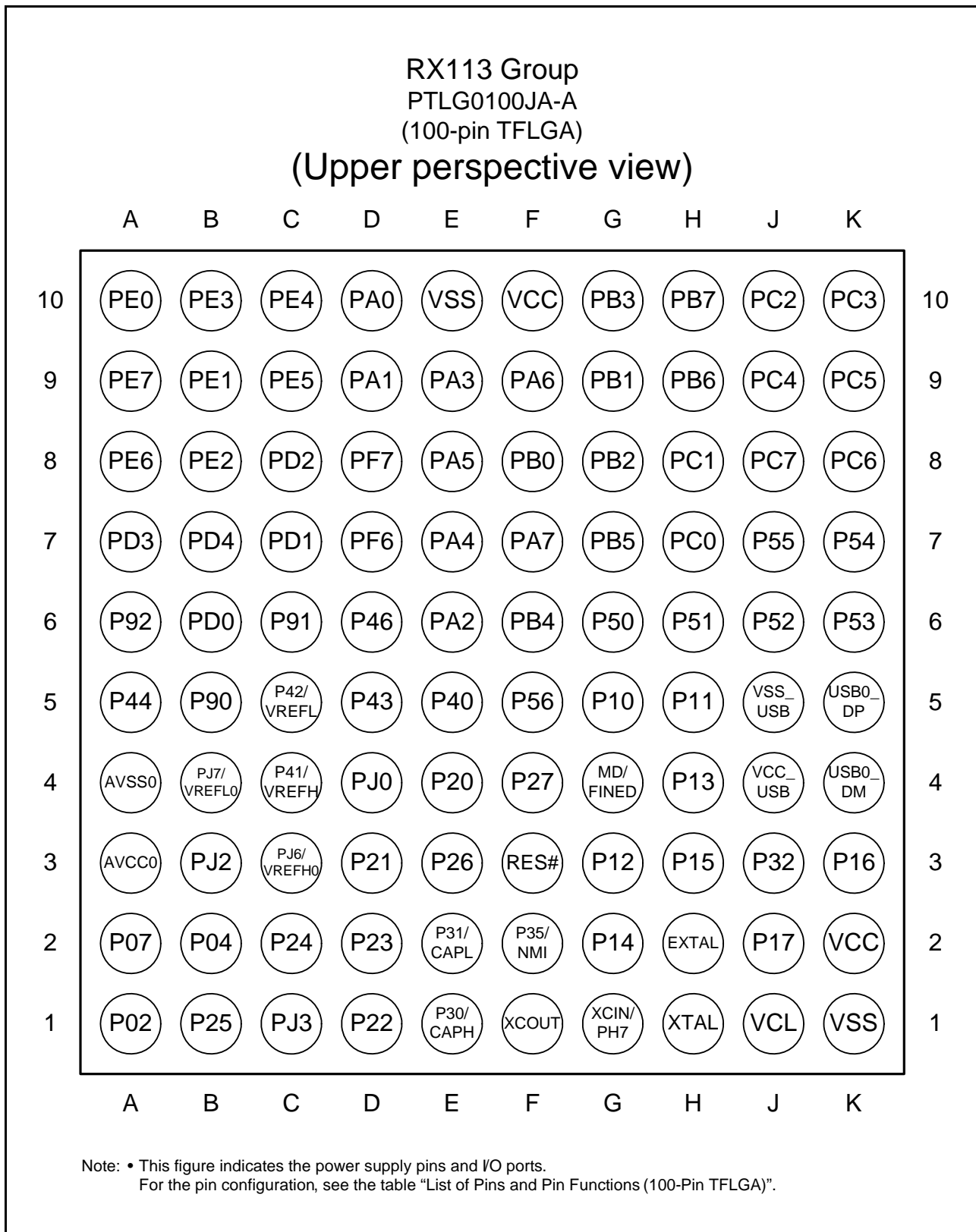


Figure 1.4 Pin Assignments of the 100-Pin TFLGA

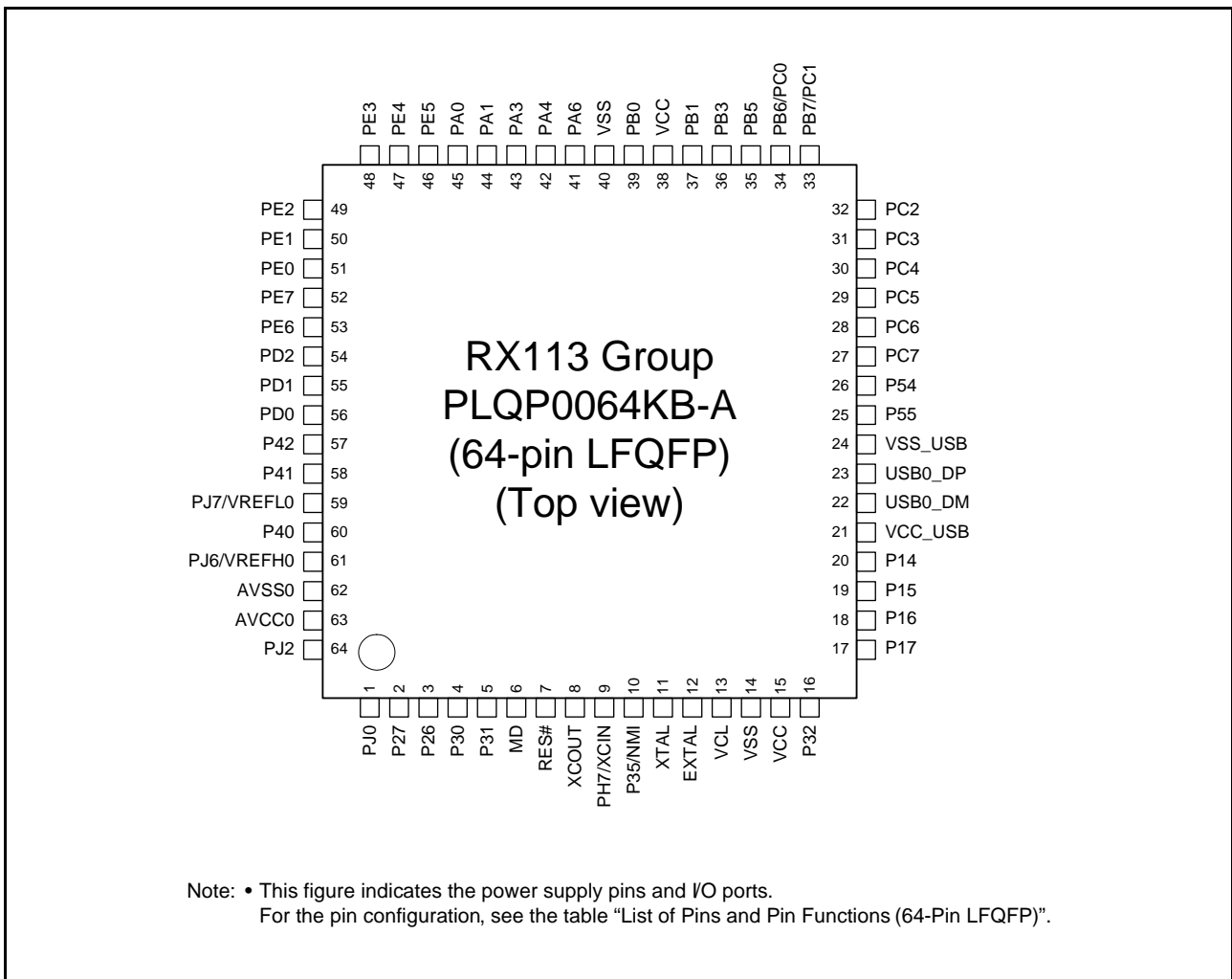


Figure 1.5 Pin Assignments of the 64-Pin LQFP



**Table 1.5 List of Pins and Pin Functions (100-Pin LQFP) (1/3)**

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, POE, RTC, TMR)	Communication (SCIE, SCIf, RSPI, RIIC, USB, SSI)	LCD, Touch	Others
1		P04	MTIOC0A/POE2#/TMC13	SCK6	TS1	
2		PJ0				DA0
3		P02	MTIOC0D/POE3#/TMR13	RXD6/SMISO6/SSCL6	TS2	
4		PJ3	MTIOC3C	CTS6#/RTS6#/SS6#	TS3	
5		P25	MTIOC4C/MTCLKB		TS4	ADTRG0#
6		P24	MTIOC4A/MTCLKA/TMR11		TS5	
7		P23	MTIOC3D/MTCLKD	CTS0#/RTS0#/SS0#	TS6	
8		P22	MTIOC3B/MTCLKC/TMO0	SCK0	TS7	
9		P21	MTIOC1B/TMC10	RXD0/SMISO0/SSCL0	TS8	
10		P20	MTIOC1A/TMR10	TXD0/SMOSI0/SSDA0	TS9	
11		P27	MTIOC2B/TMC13	SCK12/SCK1/RXD6/SMISO6/SSCL6	TS10	IRQ3/ADTRG0#/CACREF/CMPA2
12		P26	MTIOC2A/TMO1	TXD1/SMOSI1/SSDA1/USB0_VBUSEN/TXD6/SMOSI6/SSDA6	TSCAP	
13		P30	MTIOC4B/POE8#/TMR13	RXD1/SMISO1/SSCL1	CAPH	IRQ0
14		P31	MTIOC4D/TMC12	CTS1#/RTS1#/SS1#	CAPL	IRQ1
15	MD					FINED
16	RES#					
17	XCOUT					
18	XCIN	PH7				
19	UPSEL	P35				NMI
20	XTAL					
21	EXTAL					
22	VCL					
23	VSS					
24	VDD					
25		P32	MTIOC0C/RTCOUT/TMO3	TXD6/SMOSI6/SSDA6/CTS6#/RTS6#/SS6#	TS11	IRQ2
26		P17	MTIOC0C/MTIOC3A/MTIOC3B/POE8#/TMO1	SCK1/MISOA/SDA0/RXD12/RXDX12/SMISO12/SSCL12		IRQ7
27		P16	MTIOC3C/MTIOC3D/RTCOUT/TMO2	TXD1/SMOSI1/SSDA1/MOSIA/SCL0/USB0_VBUS/USB0_VBUSEN/USB0_OVRCURB		IRQ6/ADTRG0#
28		P15	MTIOC0B/MTCLKB/TMC12	RXD1/SMISO1/SSCL1/RSPCKA		IRQ5/CLKOUT/CACREF
29	UB#	P14	MTIOC0A/MTIOC3A/MTCLKA/TMR12	CTS1#/RTS1#/SS1#/SSLA0/TXD12/TXDX12/SIOX12/SMOSI12/SSDA12/USB0_OVRCURA		IRQ4
30	VCC_USB					
31				USB0_DM		
32				USB0_DP		
33	VSS_USB					
34		P13	MTIOC0B/TMO3	CTS12#/RTS12#/SS12#/CTS0#/RTS0#/SS0#	SEG00	IRQ3
35		P12	TMC11	SCK12/SCK0	SEG01	IRQ2

Table 1.5 List of Pins and Pin Functions (100-Pin LQFP) (2/3)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, POE, RTC, TMR)	Communication (SCIE, SCIF, RSPI, RIIC, USB, SSI)	LCD, Touch	Others
36		P11	MTIC5U/POE0#	RXD12/RXD12/SMISO12/SSCL12/ RXD0/SMISO0/SSCL0	SEG02	IRQ7
37		P10	MTIC5V/POE1#	TXD12/TXD12/SIOX12/SMOSI12/SSDA12/TXD0/SMOSI0/SSDA0	SEG03	IRQ6
38		P56	MTIOC1A/MTIC5W/POE2#	TXD1/SMOSI1/SSDA1	SEG04	IRQ5
39		P53	MTIOC2B	SSLA0/CTS2#/RTS2#/SS2#	SEG05	
40		P52		MISOA/RXD2/SMISO2/SSCL2	SEG06	
41		P51	MTIOC4C	RSPCKA/SCK2	SEG07	
42		P50	MTIOC2A	MOSIA/TXD2/SMOSI2/SSDA2	SEG08	
43		P55	MTIOC4D/TMO3		VL1	
44		P54	MTIOC4B/TMCI1		VL2	
45		PC7	MTIOC3A/MTCLKB/TMO2	TXD1/SMOSI1/SSDA1/MISOA/TXD8/SMOSI8/SSDA8/USB0_OVRCURB	VL3	CACREF
46		PC6	MTIOC3C/MTCLKA/TMCI2	RXD1/SMISO1/SSCL1/MOSIA/RXD8/SMISO8/SSCL8/USB0_EXICEN	VL4	
47		PC5	MTIOC3B/MTCLKD/TMRI2	SCK1/RSPCKA/SCK8/USB0_ID	COM0	
48		PC4	MTIOC3D/MTCLKC/POE0#/TMCI1	SSLA0/CTS8#/RTS8#/SS8#/SCK5/USB0_VBUSEN/USB0_VBUS *1	COM1	IRQ2/CLKOUT
49		PC3	MTIOC4D	TXD5/SMOSI5/SSDS5/IRTXD5	COM2	
50		PC2	MTIOC4B	RXD5/SMOSI5/SSCL5/IRRXD5/SSLA3	COM3	
51		PC1	MTIOC3A	SCK5/SSLA2	SEG09	
52		PC0	MTIOC3C	CTS5#/RTS5#/SS5#/SSLA1	SEG10	
53		PB7	MTIOC3B	TXD9/SMOSI9/SSDA9/SSITXD0	SEG11/COM4	
54		PB6	MTIOC3D	RXD9/SMISO9/SSCL9/SSIRXD0	SEG12/COM5	
55		PB5	MTIOC1B/MTIOC2A/POE1#/TMRI1	SCK9/SSISCK0	SEG13/COM6	
56		PB4		CTS9#/RTS9#/SS9#	SEG14	
57		PB3	MTIOC0A/MTIOC3B/MTIOC4A/POE3#/TMO0	SCK6/AUDIO_MCLK/USB0_OVRCURA	SEG15/COM7	
58		PB2		CTS6#/RTS6#/SS6#	SEG16	
59		PB1	MTIOC0C/MTIOC4C/TMCI0	TXD6/SMOSI6/SSDA6/SSIWS0	SEG17	IRQ4
60	VCC					
61		PB0	MTIOC0C/MTIC5W/RTCOUT	SCL0/RSPCKA/RXD6/SMISO6/SSCL6		IRQ2/ADTRG0#
62	VSS					
63		PA6	MTIC5V/MTCLKB/MTIOC2A/POE2#/TMCI3	CTS5#/RTS5#/SS5#/SDA0/MOSIA/RXD8/SMISO8/SSCL8		IRQ3
64		PA7		TXD8/SMOSI8/SSDS8	SEG18	
65		PA5		SCK8	SEG19	
66		PA4	MTIOC2B/MTIC5U/MTCLKA/TMRI0	TXD5/SMOSI5/SSDA5/IRTXD5/SSLA0/CTS8#/RTS8#/SS8#	SEG20	IRQ5/CVREFB1
67		PA3	MTIOC0D/MTIOC1B/MTCLKD/POE0#	RXD5/SMISO5/SSCL5/IRRXD5/MISOA	SEG21	IRQ6/CMPB1

**Table 1.5 List of Pins and Pin Functions (100-Pin LQFP) (3/3)**

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, POE, RTC, TMR)	Communication (SCIE, SCIF, RSPI, RIIC, USB, SSI)	LCD, Touch	Others
68		PA2		RXD5/SMISO5/SSCL5/IRRXD5/SSLA3	SEG22	
69		PA1	MTIOC0B/MTCLKC/RTCOUT	SCK5/SSLA2	SEG23	
70		PA0	MTIOC4A	SSLA1	SEG24	CACREF
71		PF7	MTIOC3A		SEG25	
72		PF6	MTIOC3C		SEG26	
73		PE5	MTIOC2B/MTIOC4C	MISOA/TXD9/SMOSI9/SSDA9	SEG27	IRQ5/AN013/ CMPOB1
74		PE4	MTIOC1A/MTIOC3A/MTIOC4D	MOSIA/RXD9/SMISO9/SSCL9/SSIWS0	SEG28	IRQ4/AN012
75		PE3	MTIOC0A/MTIOC1B/MTIOC4B/POE8#	CTS12#/RTS12#/SS12#/RSPCKA/SCK9/AUDIO_MCLK	SEG29	IRQ3/AN011
76		PE2	MTIOC4A	RXD12/RDX12/SMISO12/SSCL12/SSIRXD0	SEG30	IRQ7/AN010/ CVREFB0
77		PE1	MTIOC4C	TXD12/TDX12/SIOX12/SMOSI12/SSDA12/SSITXD0	SEG31	IRQ1/AN009/ CMPB0
78		PE0	MTIOC2A/POE3#	SCK12/CTS9#/RTS9#/SS9#/SSISCK0	SEG32	IRQ0/AN008
79		PE7			SEG33	IRQ7/AN015/ CMPOB0
80		PE6			SEG34	IRQ6/AN014
81		PD4	POE3#		SEG35	IRQ4
82		PD3	POE8#		SEG36	IRQ3
83		PD2	MTIOC4D		SEG37	IRQ2
84		PD1	MTIOC4B		SEG38	IRQ1
85		PD0			SEG39	IRQ0
86		P92				AN021
87		P91				AN007
88		P46				AN006
89		P90				AN005
90		P44				AN004
91		P43				AN003
92	VREFL	P42				AN002
93	VREFH	P41				AN001
94	VREFL0	PJ7				
95		P40				AN000
96	VREFH0	PJ6				
97	AVSS0					
98	AVCC0					
99		P07		TXD6/SMOSI6/SSDA6	TS0	ADTRG0#
100		PJ2				DA1

Note 1. Not 5 V tolerant.

**Table 1.6 List of Pins and Pin Functions (100-Pin TFLGA) (1/3)**

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, POE, RTC, TMR)	Communication (SCIE, SCIF, RSPI, RIIC, USB, SSI)	LCD, Touch	Others
A1		P02	MTIOC0D/POE3#/TMR13	RXD6/SMISO6/SSCL6	TS2	
A2		P07		TXD6/SMOSI6/SSDA6	TS0	ADTRG0#
A3	AVCC0					
A4	AVSS0					
A5		P44				AN004
A6		P92				AN021
A7		PD3	POE8#		SEG36	IRQ3
A8		PE6			SEG34	IRQ6/AN014
A9		PE7			SEG33	IRQ7/AN015/ CMPOB0
A10		PE0	MTIOC2A/POE3#	SCK12/CTS9#/RTS9#/SS9#/ SSISCK0	SEG32	IRQ0/AN008
B1		P25	MTIOC4C/MTCLKB		TS4	ADTRG0#
B2		P04	MTIOC0A/POE2#/TMC13	SCK6	TS1	
B3		PJ2				DA1
B4	VREFL0	PJ7				
B5		P90				AN005
B6		PD0			SEG39	IRQ0
B7		PD4	POE3#		SEG35	IRQ4
B8		PE2	MTIOC4A	RXD12/RDX12/SMISO12/ SSCL12/SSIRXD0	SEG30	IRQ7/AN010/ CVREFB0
B9		PE1	MTIOC4C	TXD12/TDX12/SIOX12/SMOSI12/ SSDA12/SSITXD0	SEG31	IRQ1/AN009/ CMPB0
B10		PE3	MTIOC0A/MTIOC1B/ MTIOC4B/POE8#	CTS12#/RTS12#/SS12#/RSPCKA/ SCK9/AUDIO_MCLK	SEG29	IRQ3/AN011
C1		PJ3	MTIOC3C	CTS6#/RTS6#/SS6#	TS3	
C2		P24	MTIOC4A/MTCLKA/ TMR11		TS5	
C3	VREFH0	PJ6				
C4	VREFH	P41				AN001
C5	VREFL	P42				AN002
C6		P91				AN007
C7		PD1	MTIOC4B		SEG38	IRQ1
C8		PD2	MTIOC4D		SEG37	IRQ2
C9		PE5	MTIOC2B/MTIOC4C	MISOA/TXD9/SMOSI9/SSDA9	SEG27	IRQ5/AN013/ CMPOB1
C10		PE4	MTIOC1A/MTIOC3A/ MTIOC4D	MOSIA/RXD9/SMISO9/SSCL9/ SSIWS0	SEG28	IRQ4/AN012
D1		P22	MTIOC3B/MTCLKC/ TMO0	SCK0	TS7	
D2		P23	MTIOC3D/MTCLKD	CTS0#/RTS0#/SS0#	TS6	
D3		P21	MTIOC1B/TMC10	RXD0/SMISO0/SSCL0	TS8	
D4		PJ0				DA0
D5		P43				AN003
D6		P46				AN006
D7		PF6	MTIOC3C		SEG26	
D8		PF7	MTIOC3A		SEG25	

**Table 1.6 List of Pins and Pin Functions (100-Pin TFLGA) (2/3)**

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, POE, RTC, TMR)	Communication (SCIE, SCIF, RSPI, RIIC, USB, SSI)	LCD, Touch	Others
D9		PA1	MTIOC0B/MTCLKC/RTCOUT	SCK5/SSLA2	SEG23	
D10		PA0	MTIOC4A	SSLA1	SEG24	CACREF
E1		P30	MTIOC4B/POE8#/TMR13	RXD1/SMISO1/SSCL1	CAPH	IRQ0
E2		P31	MTIOC4D/TMCI2	CTS1#/RTS1#/SS1#	CAPL	IRQ1
E3		P26	MTIOC2A/TMO1	TXD1/SMOSI1/SSDA1/ USB0_VBUSEN/TXD6/SMOSI6/ SSDA6	TSCAP	
E4		P20	MTIOC1A/TMRI0	TXD0/SMOSI0/SSDA0	TS9	
E5		P40				AN000
E6		PA2		RXD5/SMISO5/SSCL5/IRRXD5/ SSLA3	SEG22	
E7		PA4	MTIOC2B/MTIC5U/ MTCLKA/TMRI0	TXD5/SMOSI5/SSDA5/IRTXD5/ SSLA0/CTS8#/RTS8#/SS8#	SEG20	IRQ5/CVREFB1
E8		PA5		SCK8	SEG19	
E9		PA3	MTIOC0D/MTIOC1B/ MTCLKD/POE0#	RXD5/SMISO5/SSCL5/IRRXD5/ MISOA	SEG21	IRQ6/CMPB1
E10	VSS					
F1	XCOU					
F2	UPSEL	P35				NMI
F3	RES#					
F4		P27	MTIOC2B/TMCI3	SCK12/SCK1/RXD6/SMISO6/ SSCL6	TS10	IRQ3/ ADTRG0#/ CACREF/ CMPA2
F5		P56	MTIOC1A/MTIC5W/ POE2#	TXD1/SMOSI1/SSDA1	SEG4	IRQ5
F6		PB4		CTS9#/RTS9#/SS9#	SEG14	
F7		PA7		TXD8/SMOSI8/SSDS8	SEG18	
F8		PB0	MTIOC0C/MTIC5W/ RTCOUT	SCL0/RSPCKA/RXD6/SMISO6/ SSCL6		IRQ2/ADTRG0#
F9		PA6	MTIC5V/MTCLKB/ MTIOC2A/POE2#/TMCI3	CTS5#/RTS5#/SS5#/SDA0/MOSIA/ RXD8/SMOSI8/SSCL8		IRQ3
F10	VCC					
G1	XCIN	PH7				
G2	UB#	P14	MTIOC0A/MTIOC3A/ MTCLKA/TMRI2	CTS1#/RTS1#/SS1#/SSLA0/ TXD12/TXD12/SIOX12/SMOSI12/ SSDA12/USB0_OVRCURA		IRQ4
G3		P12	TMCI1	SCK12/SCK0	SEG01	IRQ2
G4	MD					FINED
G5		P10	MTIC5V/POE1#	TXD12/TXD12/SIOX12/SMOSI12/ SSDA12/TXD0/SMOSI0/SSDA0	SEG03	IRQ6
G6		P50	MTIOC2A	MOSIA/TXD2/SMOSI2/SSDA2	SEG08	
G7		PB5	MTIOC1B/MTIOC2A/ POE1#/TMR11	SCK9/SSISCK0	SEG13/ COM6	
G8		PB2		CTS6#/RTS6#/SS6#	SEG16	
G9		PB1	MTIOC0C/MTIOC4C/ TMCI0	TXD6/SMOSI6/SSDA6/SSIWS0	SEG17	IRQ4
G10		PB3	MTIOC0A/MTIOC3B/ MTIOC4A/POE3#/TMO0	SCK6/AUDIO_MCLK/ USB0_OVRCURA	SEG15/ COM7	

**Table 1.6 List of Pins and Pin Functions (100-Pin TFLGA) (3/3)**

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, POE, RTC, TMR)	Communication (SCIE, SCIf, RSPI, RIIC, USB, SSI)	LCD, Touch	Others
H1	XTAL					
H2	EXTAL					
H3		P15	MTIOC0B/MTCLKB/TMCI2	RXD1/SMISO1/SSCL1/RSPCKA		IRQ5/CLKOUT/CACREF
H4		P13	MTIOC0B/TMO3	CTS12#/RTS12#/SS12#/CTS0#/RTS0#/SS0#	SEG00	IRQ3
H5		P11	MTIC5U/POE0#	RXD12/RDX12/SMISO12/SSCL12/RXD0/SMISO0/SSCL0	SEG02	IRQ7
H6		P51	MTIOC4C	RSPCKA/SCK2	SEG07	
H7		PC0	MTIOC3C	CTS5#/RTS5#/SS5#/SSLA1	SEG10	
H8		PC1	MTIOC3A	SCK5/SSLA2	SEG09	
H9		PB6	MTIOC3D	RXD9/SMISO9/SSCL9/SSIRXD0	SEG12/COM5	
H10		PB7	MTIOC3B	TXD9/SMOSI9/SSDA9/SSITXD0	SEG11/COM4	
J1	VCL					
J2		P17	MTIOC0C/MTIOC3A/MTIOC3B/POE8#/TMO1	SCK1/MISOA/SDA0/RXD12/RDX12/SMISO12/SSCL12		IRQ7
J3		P32	MTIOC0C/RTCOU/ TMO3	TXD6/SMOSI6/SSDA6/CTS6#/RTS6#/SS6#	TS11	IRQ2
J4	VCC_USB					
J5	VSS_USB					
J6		P52		MISOA/RXD2/SMISO2/SSCL2	SEG06	
J7		P55	MTIOC4D/TMO3		VL1	
J8		PC7	MTIOC3A/MTCLKB/TMO2	TXD1/SMOSI1/SSDA1/MISOA/ TXD8/SMOSI8/SSDA8/ USB0_OVRCURB	VL3	CACREF
J9		PC4	MTIOC3D/MTCLKC/ POE0#/TMCI1	SSLA0/CTS8#/RTS8#/SS8#/SCK5/ USB0_VBUSEN/USB0_VBUS *1	COM1	IRQ2/CLKOUT
J10		PC2	MTIOC4B	RXD5/SMOSI5/SSCL5/IRRXD5/SSLA3	COM3	
K1	VSS					
K2	VDD					
K3		P16	MTIOC3C/MTIOC3D/ RTCOU/TMO2	TXD1/SMOSI1/SSDA1/MOSIA/ SCL0/USB0_VBUS/ USB0_VBUSEN/USB0_OVRCURB		IRQ6/ADTRG0#
K4				USB0_DM		
K5				USB0_DP		
K6		P53	MTIOC2B	SSLA0/CTS2#/RTS2#/SS2#	SEG05	
K7		P54	MTIOC4B/TMCI1		VL2	
K8		PC6	MTIOC3C/MTCLKA/ TMCI2	RXD1/SMISO1/SSCL1/MOSIA/ RXD8/SMISO8/SSCL8/ USB0_EXICEN	VL4	
K9		PC5	MTIOC3B/MTCLKD/ TMRI2	SCK1/RSPCKA/SCK8/USB0_ID	COM0	
K10		PC3	MTIOC4D	TXD5/SMOSI5/SSDS5/IRTXD5	COM2	

Note 1. Not 5 V tolerant.

**Table 1.7 List of Pins and Pin Functions (64-Pin LQFP) (1/2)**

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, POE, RTC, TMR)	Communication (SCIE, SCIF, RSPI, RIIC, USB, SSI)	LCD, Touch	Others
1		PJ0				DA0
2		P27	MTIOC2B/TMCI3	SCK1/SCK12/RXD6/SMISO6/SSCL6		IRQ3/CMPA2/CACREF/ADTRG0#
3		P26	MTIOC2A/TMO1	TXD1/SMOSI1/SSDA1/USB0_VBUSEN/TXD6/SMOSI6/SSDA6		
4		P30	MTIOC4B/POE8#/TMRI3	RXD1/SMISO1/SSCL1	CAPH	IRQ0
5		P31	MTIOC4D/TMCI2	CTS1#/RTS1#/SS1#	CAPL	IRQ1
6	MD					FINED
7	RES#					
8	XCOUT					
9	XCIN	PH7				
10	UPSEL	P35				NMI
11	XTAL					
12	EXTAL					
13	VCL					
14	VSS					
15	VCC					
16		P32	MTIOC0C/RTCOUT/TMO3	TXD6/SMOSI6/SSDA6/CTS6#/RTS6#/SS6#		IRQ2
17		P17	MTIOC0C/MTIOC3A/MTIOC3B/POE8#/TMO1	SCK1/MISOA/SDA0/RXD12/RXD12/SMISO12/SSCL12		IRQ7
18		P16	MTIOC3C/MTIOC3D/RTCOUT/TMO2	TXD1/SMOSI1/SSDA1/MOSIA/SCL0/USB0_VBUS/USB0_VBUSEN/USB0_OVRCURB		IRQ6/ADTRG0#
19		P15	MTIOC0B/MTCLKB/TMCI2	RXD1/SMISO1/SSCL1/RSPCKA		IRQ5/CLKOUT/CACREF
20	UB#	P14	MTIOC0A/MTIOC3A/MTCLKA/TMRI2	CTS1#/RTS1#/SS1#/SSLA0/TXD12/TXD12/SIOX12/SMOSI12/SSDA12/USB0_OVRCURA		IRQ4
21	VCC_USB					
22				USB0_DM		
23				USB0_DP		
24	VSS_USB					
25		P55	MTIOC4D/TMO3		VL1	
26		P54	MTIOC4B/TMCI1		VL2	
27		PC7	MTIOC3A/MTCLKB/TMO2	TXD1/SMOSI1/SSDA1/MISOA/TXD8/SMOSI8/SSDA8/USB0_OVRCURB	VL3	CACREF
28		PC6	MTIOC3C/MTCLKA/TMCI2	RXD1/SMISO1/SSCL1/MOSIA/RXD8/SMISO8/SSCL8/USB0_EXICEN	VL4	
29		PC5	MTIOC3B/MTCLKD/TMRI2	SCK1/RSPCKA/SCK8/USB0_ID	COM0	
30		PC4	MTIOC3D/MTCLKC/POE0#/TMCI1	SSLA0/CTS8#/RTS8#/SS8#/SCK5/USB0_VBUSEN/USB0_VBUS *1	COM1	IRQ2/CLKOUT
31		PC3	MTIOC4D	TXD5/SMOSI5/SSDA5/IRTXD5	COM2	
32		PC2	MTIOC4B	RXD5/SMISO5/SSCL5/SSLA3/IRRXD5	COM3	

**Table 1.7 List of Pins and Pin Functions (64-Pin LQFP) (2/2)**

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, POE, RTC, TMR)	Communication (SCIE, SCIF, RSPI, RIIC, USB, SSI)	LCD, Touch	Others
33		PB7/ PC1	MTIOC3B	TXD9/SMOSI9/SSDA9/SSITXD0	SEG11/ COM4	
34		PB6/ PC0	MTIOC3D	RXD9/SMOSI9/SSCL9/SSIRXD0	SEG12/ COM5	
35		PB5	MTIOC2A/MTIOC1B/ POE1#/TMRI1	SCK9/SSISCK0	SEG13/ COM6	
36		PB3	MTIOC0A/MTIOC3B/ MTIOC4A/POE3#/TMO0	SCK6/AUDIO_MCLK/ USB0_OVRCURA	SEG15/ COM7	
37		PB1	MTIOC0C/MTIOC4C/ TMCIO	TXD6/SMOSI6/SSDA6/SSIWS0	SEG17	IRQ4
38	VCC					
39		PB0	MTIC5W/MTIOC0C/ RTCOUT	SCL0/RSPCKA/RXD6/SMOSI6/ SSCL6		IRQ2/ADTRG0#
40	VSS					
41		PA6	MTIC5V/MTCLKB/ MTIOC2A/POE2#/TMC13	CTS5#/RTS5#/SS5#/SDA0/MOSIA		IRQ3
42		PA4	MTIC5U/MTCLKA/ MTIOC2B/TMRI0	TXD5/SMOSI5/SSDA5/IRTXD5/ SSLA0	SEG20	IRQ5/CVREFB1
43		PA3	MTIOC0D/MTCLKD/ MTIOC1B/POE0#	RXD5/SMISO5/SSCL5/IRRXD5/ MISOA	SEG21	IRQ6/CMPB1
44		PA1	MTIOC0B/MTCLKC/ RTCOUT	SCK5/SSLA2	SEG23	
45		PA0	MTIOC4A	SSLA1	SEG24	CACREF
46		PE5	MTIOC4C/MTIOC2B	MISOA/TXD9/SMOSI9/SSDA9	SEG27	IRQ5/AN013/ CMPOB1
47		PE4	MTIOC4D/MTIOC1A/ MTIOC3A	MOSIA/RXD9/SMISO9/SSCL9/ SSIWS0	SEG28	IRQ4/AN012
48		PE3	MTIOC0A/MTIOC1B/ MTIOC4B/POE8#	CTS12#/RTS12#/SS12#/RSPCKA/ SCK9/AUDIO_MCLK	SEG29	IRQ3/AN011
49		PE2	MTIOC4A	RXD12/RDX12/SMISO12/ SSCL12/RDX12/SSIRXD0	SEG30	IRQ7/AN010/ CVREFB0
50		PE1	MTIOC4C	TXD12/TDX12/SIOX12/SMOSI12/ SSDA12/SSITXD0	SEG31	IRQ1/AN009/ CMPB0
51		PE0	MTIOC2A/POE3#	SCK12/CTS9#/RTS9#/SS6#/ SSISCK0	SEG32	IRQ0/AN008
52		PE7			SEG33	IRQ7/AN015/ CMPOB0
53		PE6			SEG34	IRQ6/AN014
54		PD2	MTIOC4D		SEG37	IRQ2
55		PD1	MTIOC4B		SEG38	IRQ1
56		PD0			SEG39	IRQ0
57	VREFL	P42				AN002
58	VREFH	P41				AN001
59	VREFL0	PJ7				
60		P40				AN000
61	VREFH0	PJ6				
62	AVSS0					
63	AVCC0					
64		PJ2				DA1

Note 1. Not 5 V tolerant.



## 2. CPU

This MCU has the RX CPU as its core.

A variable-length instruction format has been adopted for the RX CPU. Allocating the more frequently used instructions to the shorter instruction lengths facilitates the development of efficient programs that take up less memory.

The CPU has 73 basic instructions and nine DSP instructions, for a total of 82 instructions. It has 10 addressing modes and caters to register-to-register operations, register-to-memory operations, immediate-to-register operations, immediate-to-memory operations, memory-to-memory transfer, and bitwise operations. In a single cycle, high-speed calculation is attained for not just register-to-register operations, but also for other types of combined instructions. The CPU includes an internal multiplier and an internal divider for high-speed multiplication and division.

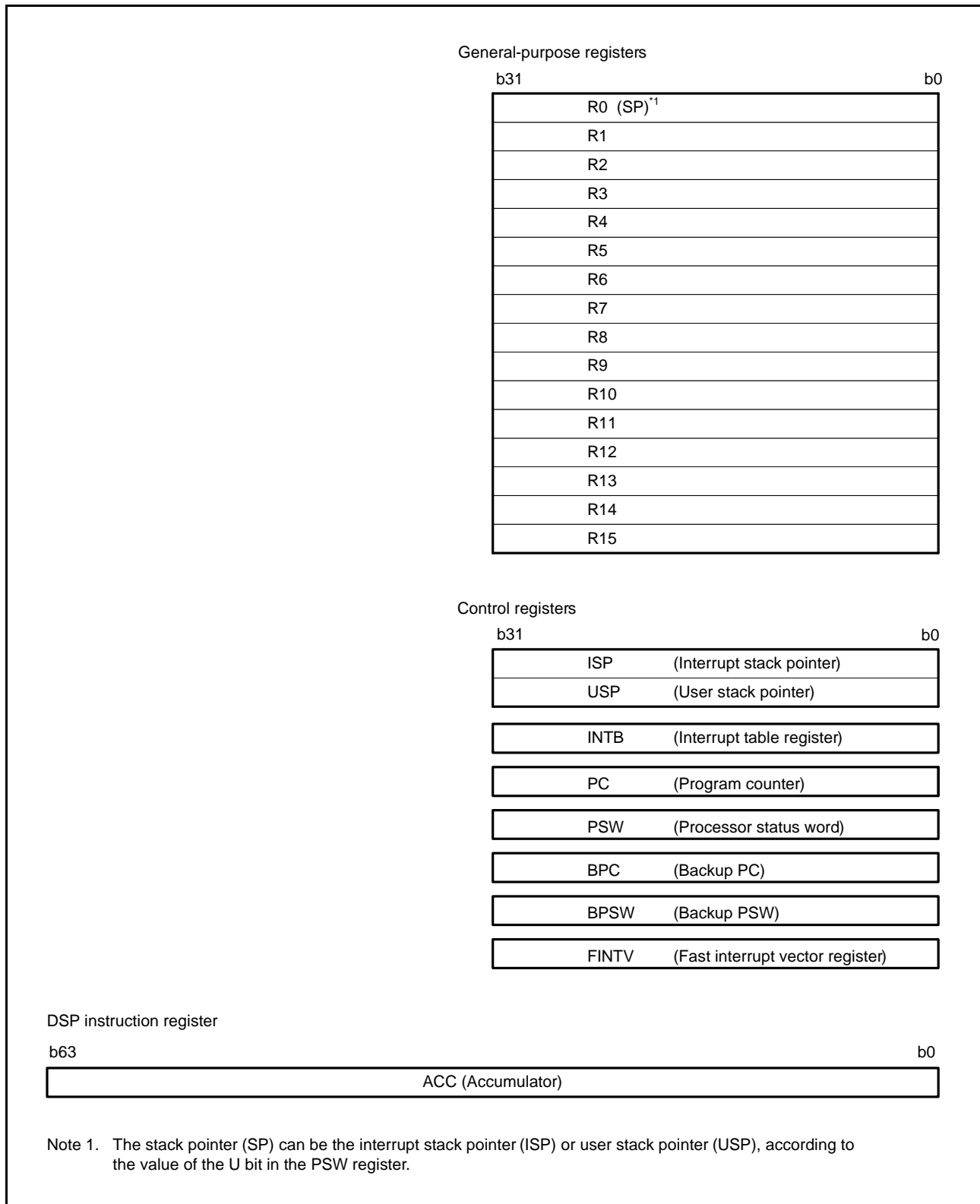
The RX CPU has a five-stage pipeline for processing instructions. The stages are instruction fetching, decoding, execution, memory access, and write-back. In cases where pipeline processing is drawn-out by memory access, subsequent operations may in fact be executed earlier. By adopting an “out-of-order completion” of this kind, instruction execution is controlled to optimize the number of clock cycles.

### 2.1 Features

- Minimum instruction execution rate: One instruction per clock cycle
- Address space: 4-Gbyte linear
- Register set of the CPU
  - General purpose: Sixteen 32-bit registers
  - Control: Eight 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 1 to 8 bytes)
  - Short formats for frequently used instructions
- DSP instructions: 9
  - Supports 16-bit × 16-bit multiplication and multiply-and-accumulate operations.
  - Rounds the data in the accumulator.
- Addressing modes: 10
- Five-stage pipeline
  - Adoption of “out-of-order completion”
- Processor modes
  - A supervisor mode and a user mode are supported.
- Data arrangement
  - Selectable as little endian or big endian

## 2.2 Register Set of the CPU

The RX CPU has 16 general-purpose registers, eight 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)

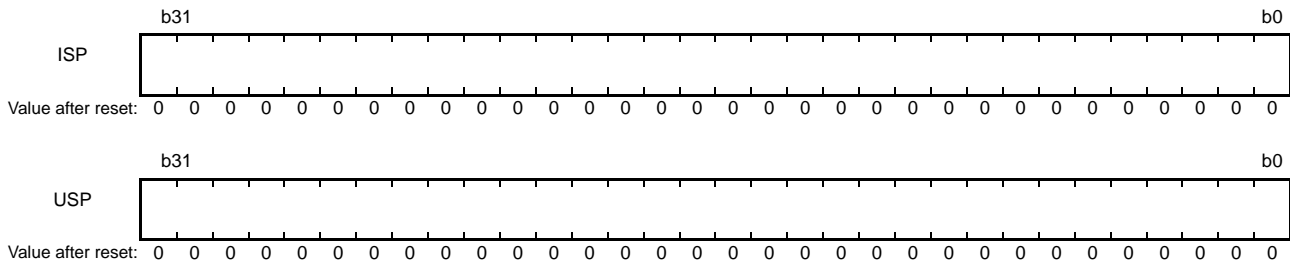
This CPU has 16 general-purpose registers (R0 to R15). R0 to R15 can be used as data registers or address registers. R0, a general-purpose register, also functions as the stack pointer (SP). The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (USP) by the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

### 2.2.2 Control Registers

This CPU has the following eight 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)

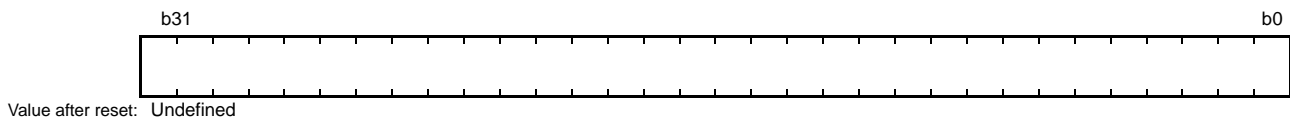
### 2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP)



The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).

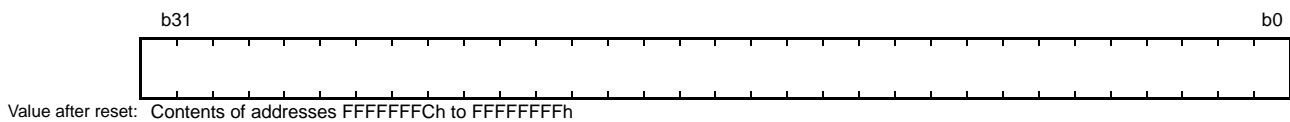
Set the ISP or USP to a multiple of 4, as this reduces the numbers of cycles required to execute interrupt sequences and instructions entailing stack manipulation.

### 2.2.2.2 Interrupt Table Register (INTB)



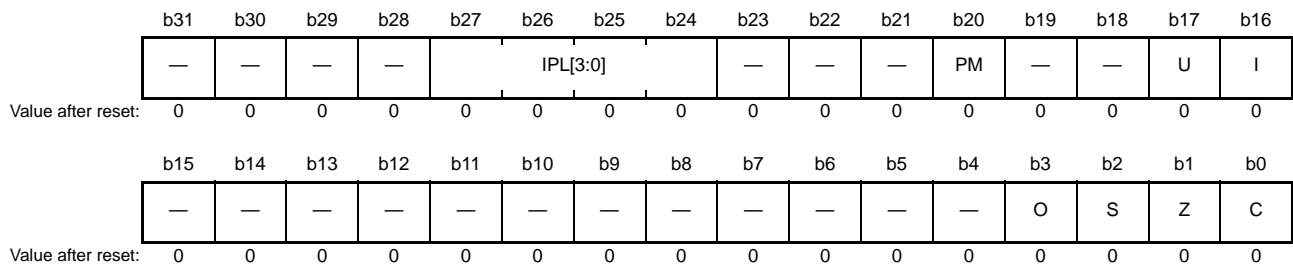
The interrupt table register (INTB) specifies the address where the relocatable vector table starts.

### 2.2.2.3 Program Counter (PC)



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

## 2.2.2.4 Processor Status Word (PSW)



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

Note 1. In user mode, writing to the IPL[3:0], PM, U, and I bits by an MVTC or a POPC instruction is ignored.

Writing to the IPL[3:0] bits by an MVTIPL instruction generates a privileged instruction exception.

Note 2. In supervisor mode, writing to the PM bit by an MVTC or a POPC instruction is ignored, but writing to the other bits is possible.

Note 3. Switching from supervisor mode to user mode requires execution of an RTE instruction after having set the PSW.PM bit saved on the stack to 1 or executing an RTFI instruction after having set the BPSW.PM bit to 1.

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

### C Flag (Carry Flag)

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

### Z Flag (Zero Flag)

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

### S Flag (Sign Flag)

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

### O Flag (Overflow Flag)

This flag indicates that an overflow occurred during an operation.

### I Bit (Interrupt Enable)

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

### U Bit (Stack Pointer Select)

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

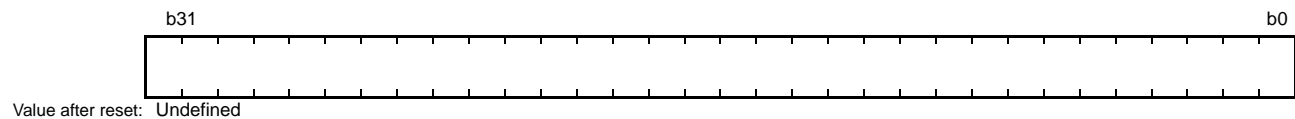
### PM Bit (Processor Mode Select)

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

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

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

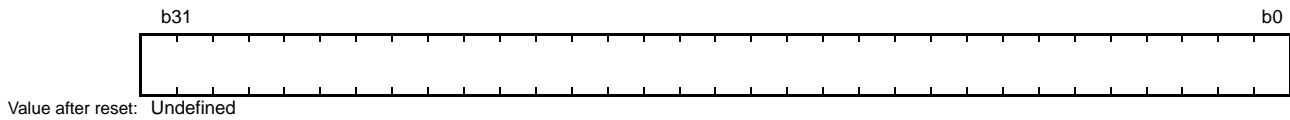
#### 2.2.2.5 Backup PC (BPC)



The backup PC (BPC) is provided to speed up response to interrupts.

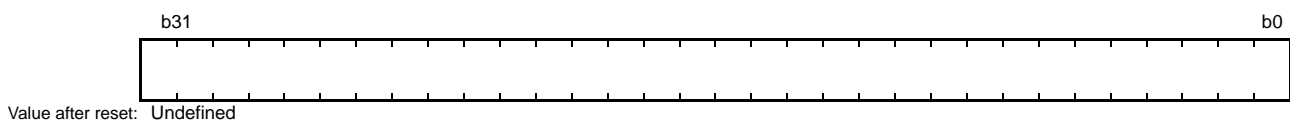
After a fast interrupt has been generated, the contents of the program counter (PC) are saved in the BPC register.

### 2.2.2.6 Backup PSW (BPSW)



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.

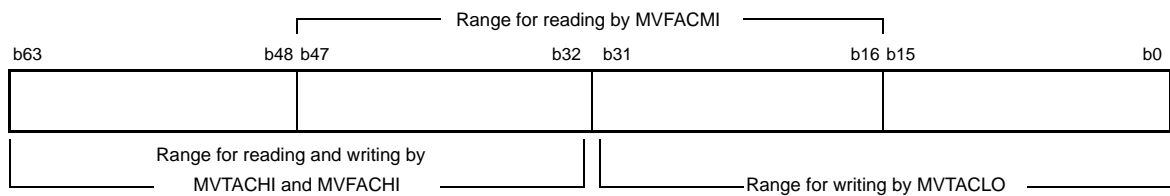
### 2.2.2.7 Fast Interrupt Vector Register (FINTV)



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.

## 2.2.3 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, MUL, and RMPA, in which case the prior value in the accumulator is modified by execution of the instruction.

Use the MVTACHI and MVTACLO instructions for writing to the accumulator. The MVTACHI and MVTACLO instructions write data to the higher-order 32 bits (bits 63 to 32) and the lower-order 32 bits (bits 31 to 0), respectively. Use the MVFACHI and MVFACMI instructions for reading data from the accumulator. The MVFACHI and MVFACMI instructions read data from the higher-order 32 bits (bits 63 to 32) and the middle 32 bits (bits 47 to 16), respectively.

## 2.3 Processor Mode

The RX CPU supports two processor modes, supervisor and user. These processor modes enable the realization of a hierarchical CPU resource protection.

Each processor mode imposes a level on rights of access to the CPU resources and the instructions that can be executed. Supervisor mode carries greater rights than those of user mode.

The initial state after a reset is supervisor mode.

### 2.3.1 Supervisor Mode

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

### 2.3.2 User Mode

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

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

### 2.3.3 Privileged Instruction

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

### 2.3.4 Switching between Processor Modes

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

#### (1) Switching from user mode to supervisor mode

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

#### (2) Switching from supervisor mode to user mode

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



## 2.4 Data Types

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

## 2.5 Endian

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

### 2.5.1 Switching the Endian

As arrangements of bytes, this MCU supports both big endian, where the higher-order byte (MSB) is at location 0, and little endian, where the lower-order byte (LSB) is at location 0.

For details on the endian setting, refer to section 3, Operating Modes.

Operations for access differ according to the endian setting and, depending on the instruction, whether 8-, 16- or 32-bit access has been selected. Operations for access in the various possible cases are described in Table 2.1 to Table 2.12.

In the tables,

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

	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 LL	—	—	—	—	—	—
Address 1	Transfer from LH	Transfer from LL	—	—	—	—	—
Address 2	—	Transfer from LH	Transfer from LL	—	—	—	—
Address 3	—	—	Transfer from LH	Transfer from LL	—	—	—
Address 4	—	—	—	Transfer from LH	Transfer from LL	—	—
Address 5	—	—	—	—	Transfer from LH	Transfer from LL	—
Address 6	—	—	—	—	—	Transfer from LH	Transfer from LL
Address 7	—	—	—	—	—	—	Transfer from LH

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

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

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

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

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

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

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

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

## 2.5.2 Access to I/O Registers

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

## 2.5.3 Notes on Access to I/O Registers

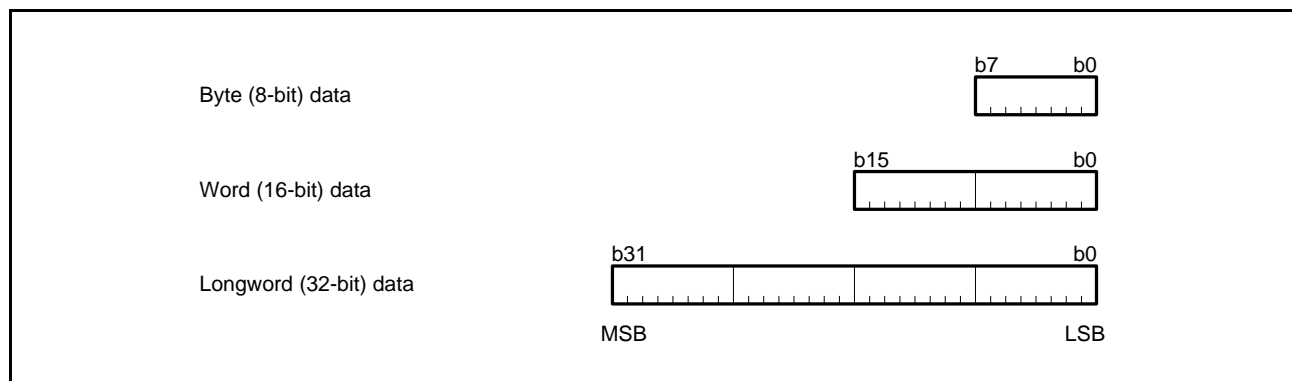
Ensure that access to I/O registers is in accord with the following rules.

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

## 2.5.4 Data Arrangement

### 2.5.4.1 Data Arrangement in Registers

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



**Figure 2.2** Data Arrangement in Registers

### 2.5.4.2 Data Arrangement in Memory

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

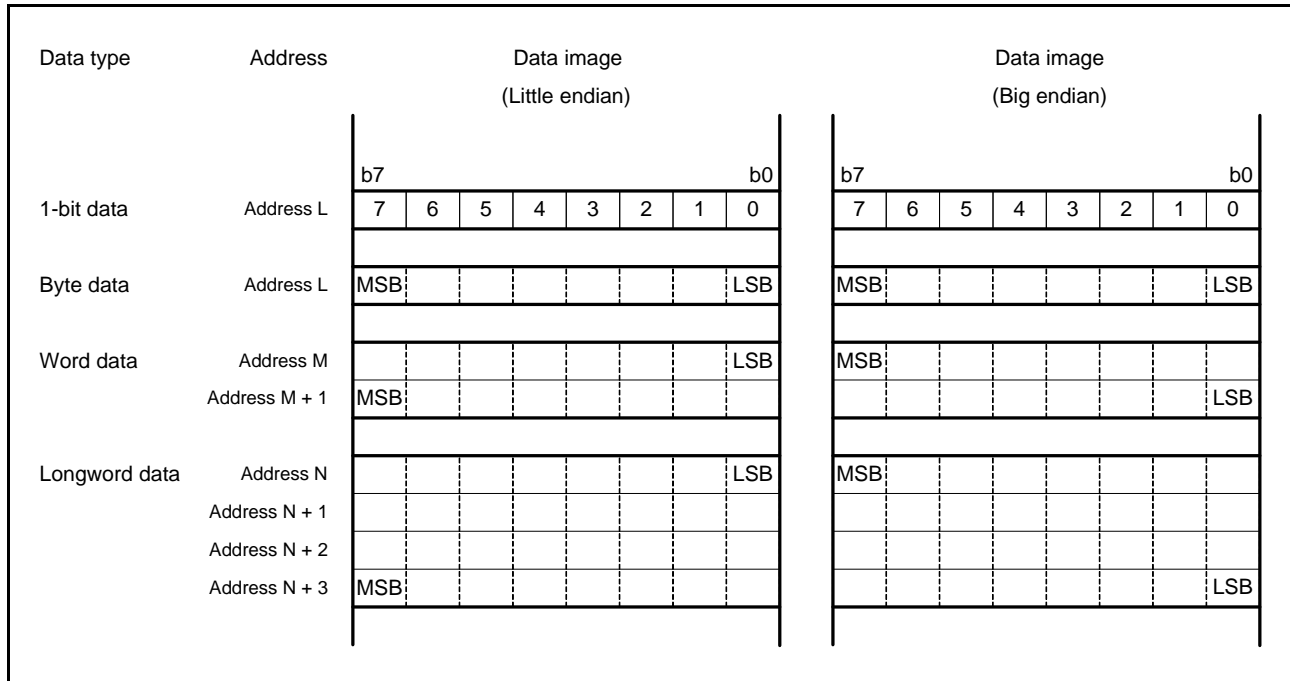


Figure 2.3 Data Arrangement in Memory

### 2.5.5 Notes on the Allocation of Instruction Codes

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

## 2.6 Vector Table

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

### 2.6.1 Fixed Vector Table

The fixed vector table is allocated to a fixed address range. The individual vectors for the privileged instruction exception, undefined instruction exception, non-maskable interrupt, and reset are allocated to addresses in the range from FFFFFFF80h to FFFFFFFFh. Figure 2.4 shows the fixed vector table.

	MSB	LSB
FFFFFFF80h	(Reserved)	
⋮	⋮	
FFFFFFFCCh	(Reserved)	
FFFFFFFD0h	Privileged instruction exception	
FFFFFFFD4h	(Reserved)	
FFFFFFFD8h	(Reserved)	
FFFFFFFDCh	Undefined instruction exception	
FFFFFFFE0h	(Reserved)	
FFFFFFFE4h	(Reserved)	
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, refer to section 14.3.1, Interrupt Vector Table.

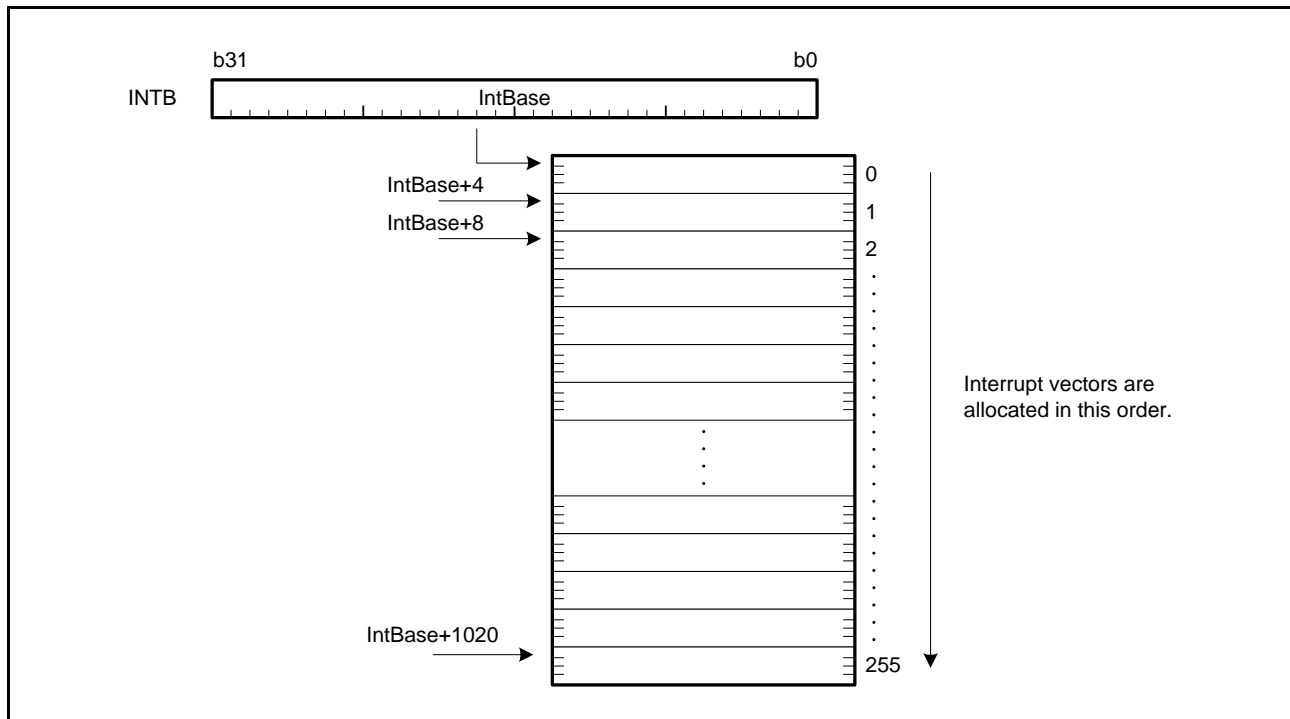


Figure 2.5 Relocatable Vector Table



## 2.7 Operation of Instructions

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

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

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

## 2.8 Pipeline

### 2.8.1 Overview

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

The operation of pipeline and respective stages is described below.

#### (1) IF stage (instruction fetch stage)

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

#### (2) D stage (decoding stage)

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

#### (3) E stage (execution stage)

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

#### (4) M stage (memory access stage)

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

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

(5) WB stage (write-back stage)

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

Figure 2.6 shows the pipeline configuration and its operation.

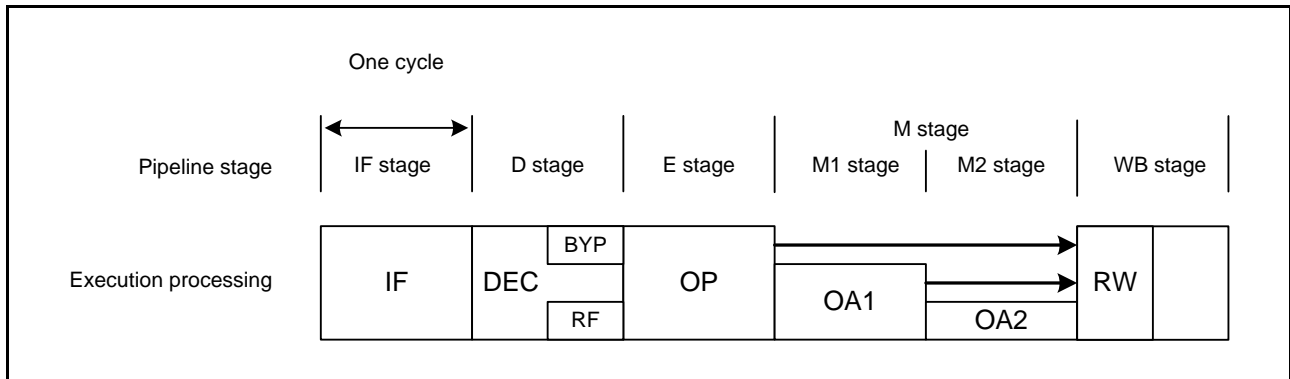


Figure 2.6 Pipeline Configuration and its Operation

## 2.8.2 Instructions and Pipeline Processing

The operands in the table below indicate the following meaning.

#IMM: Immediate

flag: bit, flag

Rs, Rs2, Rd, Rd2, Ri, Rb: General-purpose register

CR: Control register

dsp: displacement

pcdsp: displacement

### 2.8.2.1 Instructions Converted into Single Micro-Operation and Pipeline Processing

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

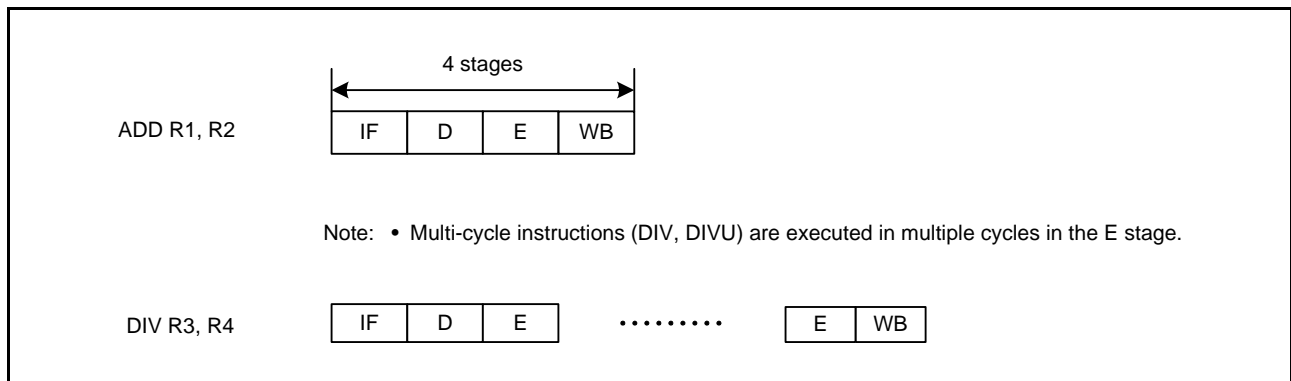
**Table 2.13 Instructions that are Converted into a Single Micro-Operation**

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
Arithmetic/logic instructions (register-register, immediate-register) Except EMUL, EMULU, RMPA, DIV, DIVU and SATR	<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, 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 2.17	Branch taken: 3 Branch not taken: 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

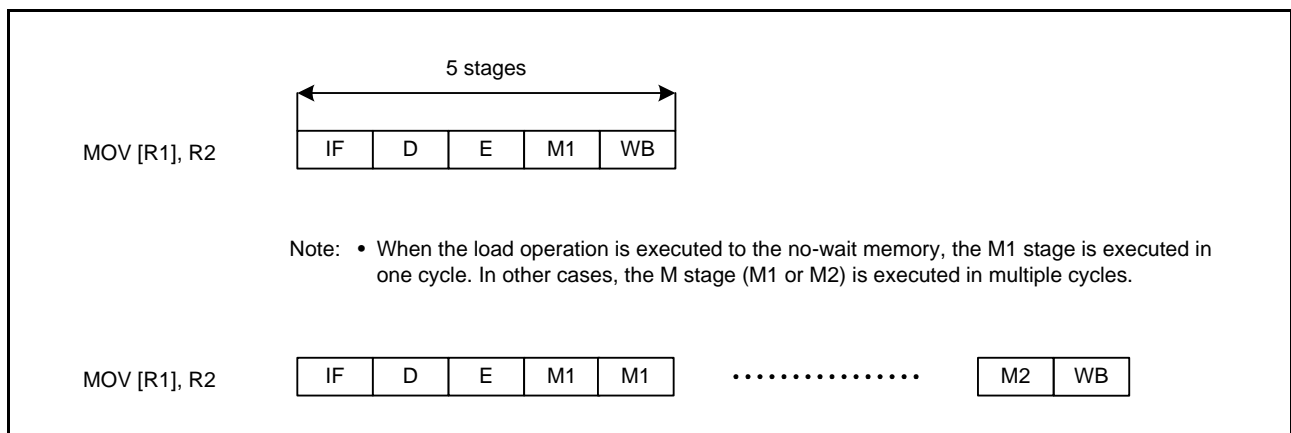
Note 1. The number of cycles for the dividing instruction varies according to the divisor and dividend.

Note 2. For the number of cycles for throughput and latency, refer to section 2.8.3, Calculation of the Instruction Processing Time.

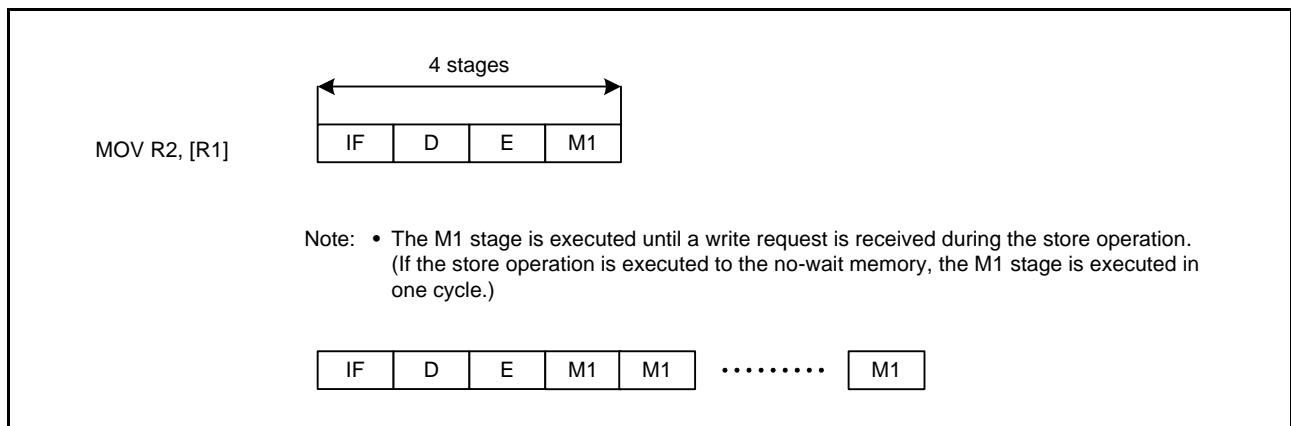
Figure 2.7 to Figure 2.9 show the operation of instructions that are converted into a basic single micro-operation.



**Figure 2.7 Operation for Register-Register, Immediate-Register**



**Figure 2.8 Load Operation**



**Figure 2.9 Store Operation**

### 2.8.2.2 Instructions Converted into Multiple Micro-Operations and Pipeline Processing

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

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

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
Arithmetic/logic instructions (memory source operand)	<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 4 to 20
Arithmetic/logic instruction (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 instruction (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 6+5×floor(n/2)+4×(n%2) n: Number of processing words*1 6+4n n: Number of processing longwords*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 instruction (restore operation of multiple registers)	<ul style="list-style-type: none"> <li>POPM “Rs-Rs2”</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 instruction (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*5	• SCMPU	—	$2+4 \times \text{floor}(n/4)+4 \times (n\%4)$ n: Number of comparison bytes*1
	• SMOVB	—	$n > 3?$ $6+3 \times \text{floor}(n/4)+3 \times (n\%4):$ $2+3n$ n: Number of transfer bytes*1
	• SMOVF, SMOVU	—	$2+3 \times \text{floor}(n/4)+3 \times (n\%4)$ n: Number of transfer bytes*1
	• SSTR.B	—	$2+\text{floor}(n/4)+n\%4$ n: Number of transfer bytes*1
	• SSTR.W	—	$2+\text{floor}(n/2)+n\%2$ n: Number of transfer words*1
	• SSTR.L	—	$2+n$ n: Number of transfer longwords
	• SUNTIL.B, SWHILE.B	—	$3+3 \times \text{floor}(n/4)+3 \times (n\%4)$ n: Number of comparison bytes*1
	• SUNTIL.W, SWHILE.W	—	$3+3 \times \text{floor}(n/2)+3 \times (n\%2)$ n: Number of comparison words*1
	• SUNTIL.L, SWHILE.L	—	$3+3 \times n$ n: Number of comparison longwords
System manipulation instructions	• RTE	—	6
	• RTFI	—	3

?: Conditional operator

Note 1. floor(x): Max. integer that is smaller than x

Note 2. For the number of cycles for throughput and latency, refer to section 2.8.3, Calculation of the Instruction Processing Time.

Note 3. The PUSHM instruction is converted into multiple store operations. The pipeline processing is the same as the one for the store operations of the MOV instruction, where the operation is repeated for the number of specified registers.

Note 4. The POPM instruction is converted into multiple load operations. The pipeline processing is the same as the one for the load operations of the MOV instruction, where the operation is repeated for the number of specified registers.

Note 5. Each of the SCMPU, SMOVU, SWHILE, and SUNTIL instructions ends the execution regardless of the specified cycles, if the end condition is satisfied during execution.

Figure 2.10 to Figure 2.14 show the operation of instructions that are converted into basic multiple micro-operations.

Note: mop: Micro-operation, stall: Pipeline stall

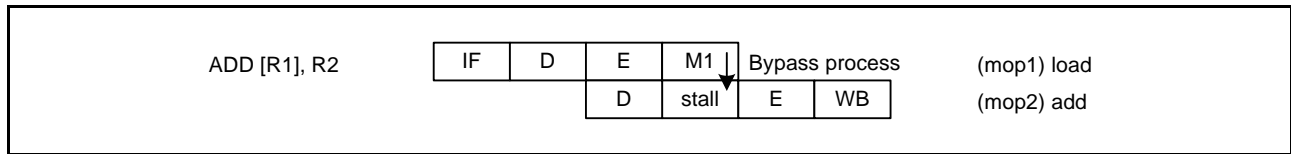


Figure 2.10 Arithmetic/Logic Instruction (Memory Source Operand)

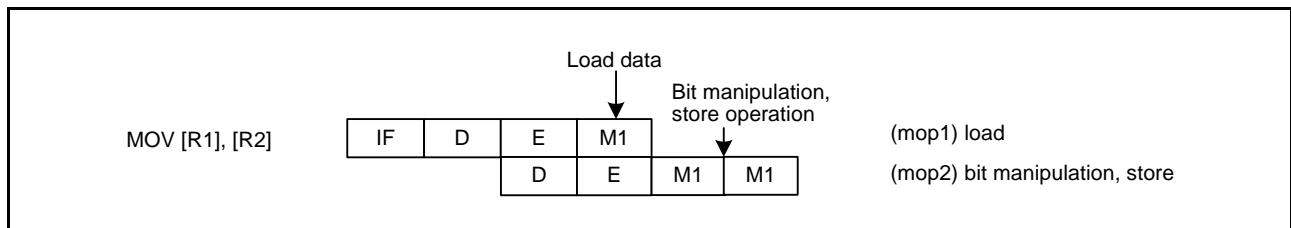


Figure 2.11 MOV Instruction (Memory-Memory), Bit Manipulation Instruction (Memory Source Operand)

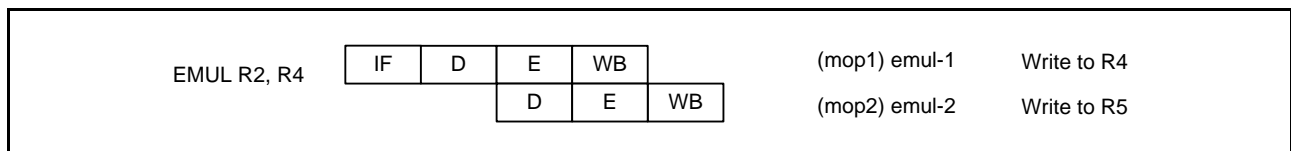


Figure 2.12 EMUL, EMULU Instructions (Register- Register, Register-Immediate)

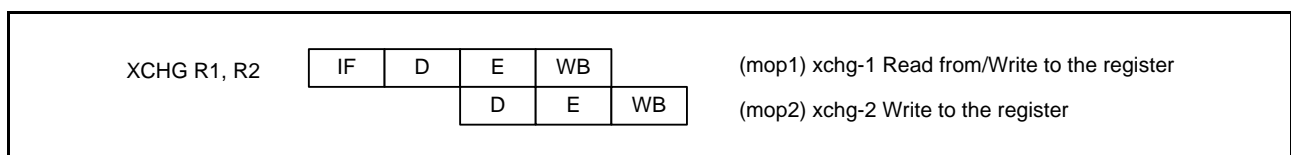


Figure 2.13 XCHG Instruction (Registers)

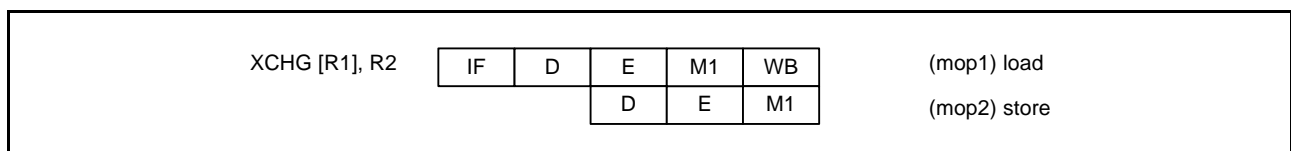


Figure 2.14 XCHG Instruction (Memory Source Operand)

### 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

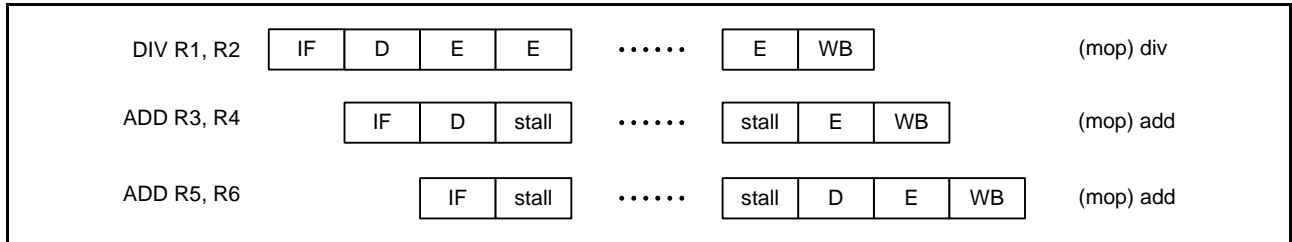


Figure 2.15 When an Instruction which Requires Multiple Cycles is Executed in the E Stage

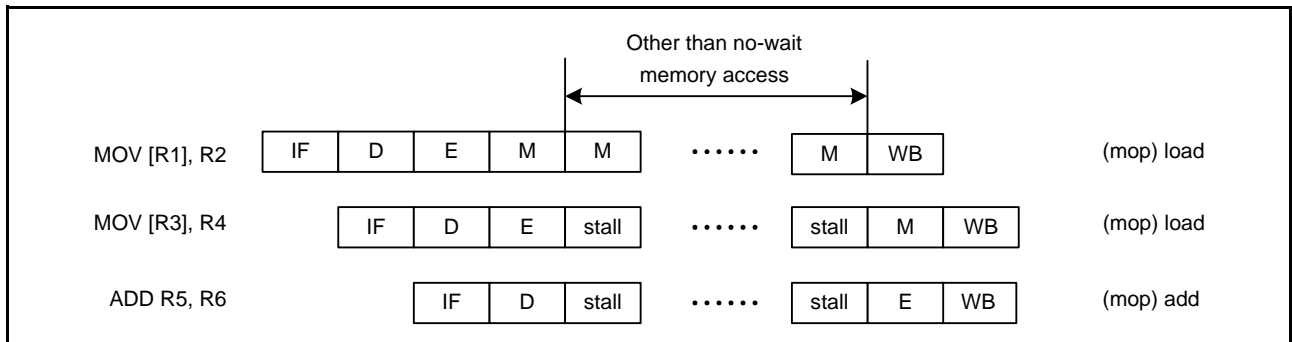


Figure 2.16 When an Instruction which Requires more than One Cycle for its Operand Access is Executed

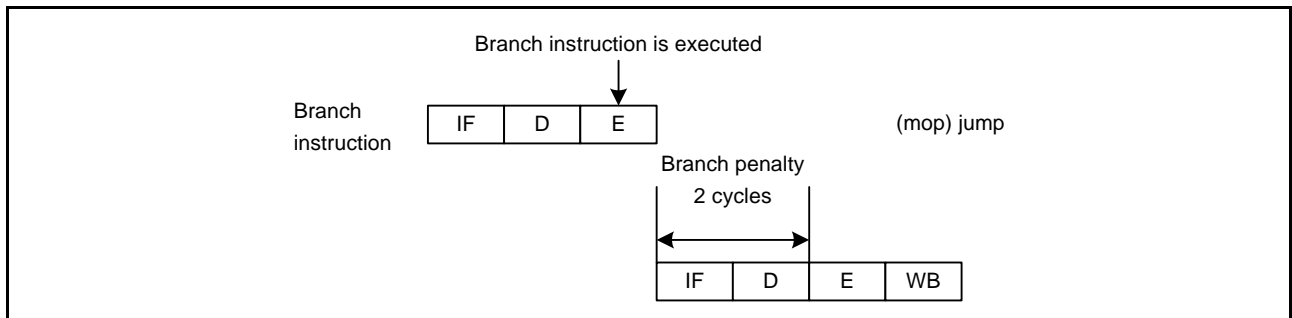


Figure 2.17 When a Branch Instruction is Executed (an Unconditional Branch Instruction is Executed or the Condition is Satisfied for a Conditional Branch Instruction)

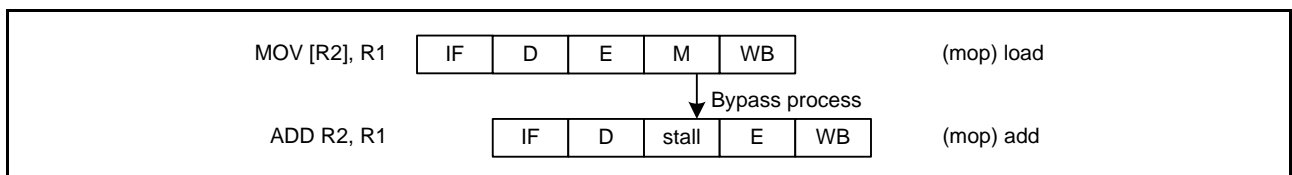


Figure 2.18 When the Subsequent Instruction Uses an Operand Read from the Memory



(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.

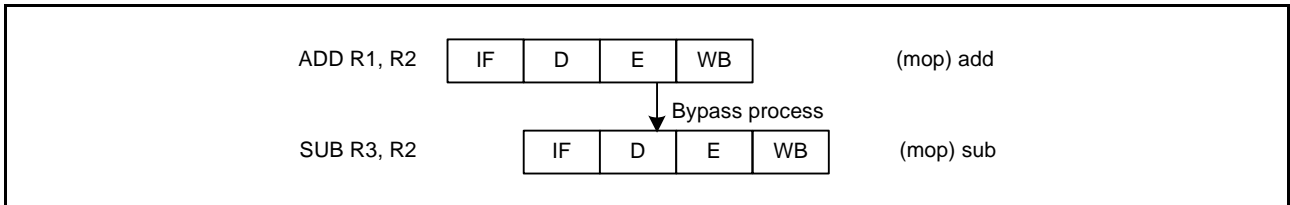


Figure 2.19 Bypass Process

(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.

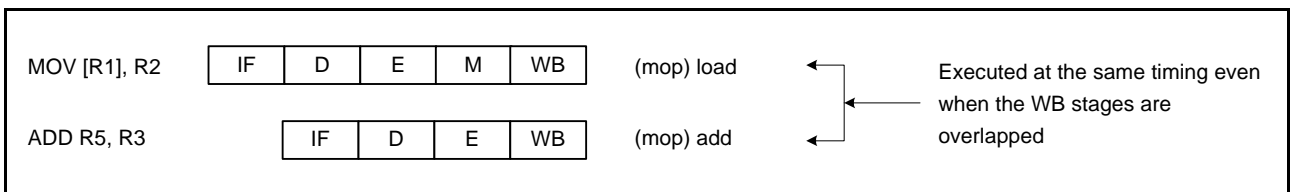


Figure 2.20 When WB Stages for the Memory Load and for the Operation are Overlapped

(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.

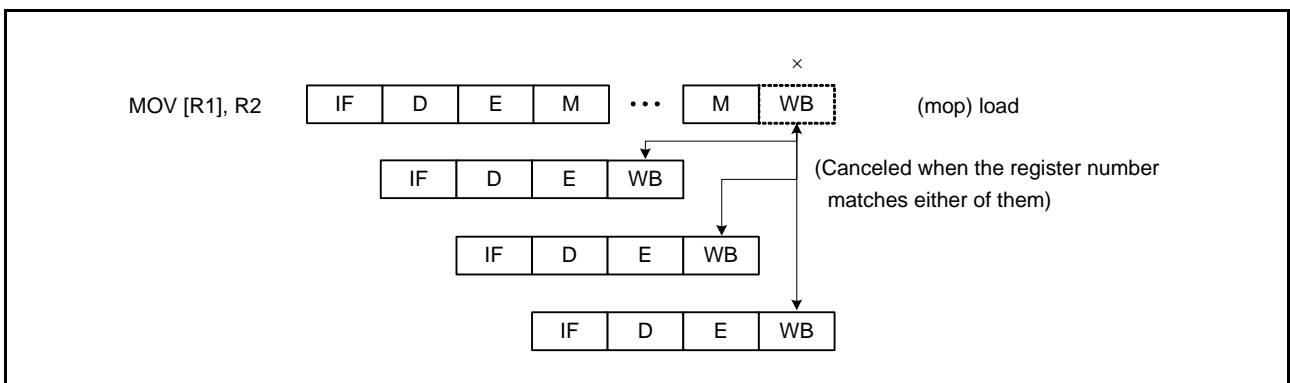
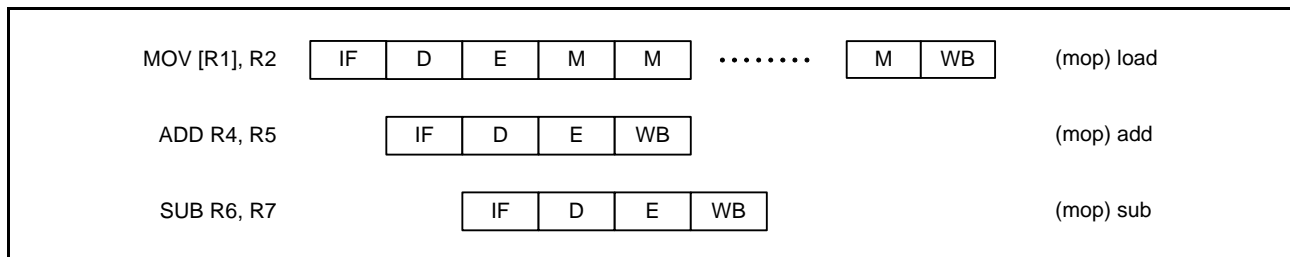


Figure 2.21 When Subsequent Instruction Writes to the Same Register before the End of Memory Load

## (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 2.22** When Load Data is not Used by the Subsequent Instruction

### 2.8.3 Calculation of the Instruction Processing Time

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

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

## 2.8.4 Numbers of Cycles for Response to Interrupts

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

**Table 2.15 Numbers of Cycles for Response to Interrupts**

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

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

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

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

## 3. Operating Modes

### 3.1 Operating Mode Types and Selection

Operating modes are selected by the pin level when a reset is released.

Table 3.1 shows the relationship between levels on the mode setting pins (MD and UB#) on release from the reset state and the operating mode selected at that time. For details on each of the operating modes, refer to section 3.3, Details of Operating Modes.

**Table 3.1 Selection of Operating Modes by the Mode Setting Pin**

Mode Setting Pin		Operating Mode
MD *1	UB#	
Low	Low	Boot mode (USB interface)
Low	High or open	Boot mode (SCI)
High	—	Single-chip mode

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

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

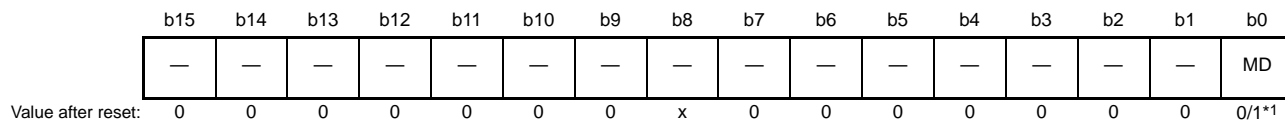
**Table 3.2 Endian Setting in Single-Chip Mode**

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

### 3.2 Register Descriptions

#### 3.2.1 Mode Monitor Register (MDMONR)

Address(es): 0008 0000h



Note 1. Depends on the setting of the mode pin (MD). When the MD pin is low, the bit value is 0; otherwise, the bit value is 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

### 3.2.2 System Control Register 1 (SYSCR1)

Address(es): 0008 0008h

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

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

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

#### RAME Bit (RAM Enable)

The RAME bit enables or disables the RAM.

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

Even when the RAME bit is set to 0, the RAM retains its value. To retain the value in the RAM, keep the specified RAM standby voltage (VRAM). For details, refer to section 42, Electrical Characteristics.

### 3.3 Details of Operating Modes

#### 3.3.1 Single-Chip Mode

In this mode, all I/O ports can be used as general input/output ports, peripheral function input/output, or interrupt input pins.

The chip starts up in single-chip mode if the high level is on the MD pin on release from the reset state.

#### 3.3.2 Boot Mode

In this mode, the on-chip flash memory modifying program (boot program) stored in a dedicated area within the MCU operates. The on-chip flash memory (ROM and E2 DataFlash) can be modified from outside the MCU by using a USB or universal asynchronous receiver/transmitter (SCI). For details, refer to section 41, Flash Memory.

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

##### 3.3.2.1 Boot Mode (USB Interface)

When a reset is released while pins MD and UB# are low, boot mode (USB interface) is selected. For details on boot mode (USB interface), refer to section 41.8.1, Boot Mode (USB Interface).

##### 3.3.2.2 Boot Mode (SCI)

When a reset is released while the MD pin is low and the UB# pin is high or left open, boot mode (SCI) is selected. For details on boot mode (SCI), refer to section 41.8.2, Boot Mode (SCI).

### 3.4 Transitions of Operating Modes

#### 3.4.1 Mode Setting Pin Levels and Operating Mode Transitions

Figure 3.1 shows operating mode transitions according to the setting of the MD and UB# pins.

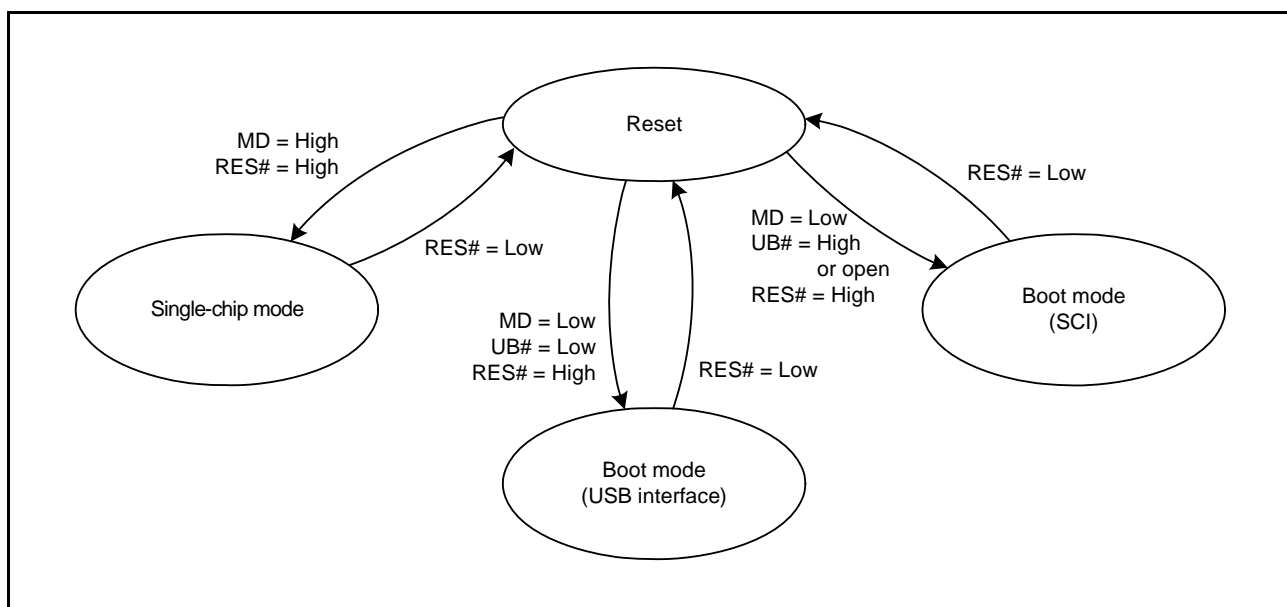


Figure 3.1 Mode Setting Pin Levels and Operating Modes

## 4. Address Space

### 4.1 Address Space

This MCU has a 4-Gbyte address space, consisting of the range of addresses from 0000 0000h to FFFF FFFFh. That is, linear access to an address space of up to 4 Gbytes is possible, and this contains program area.

Figure 4.1 shows the memory map.



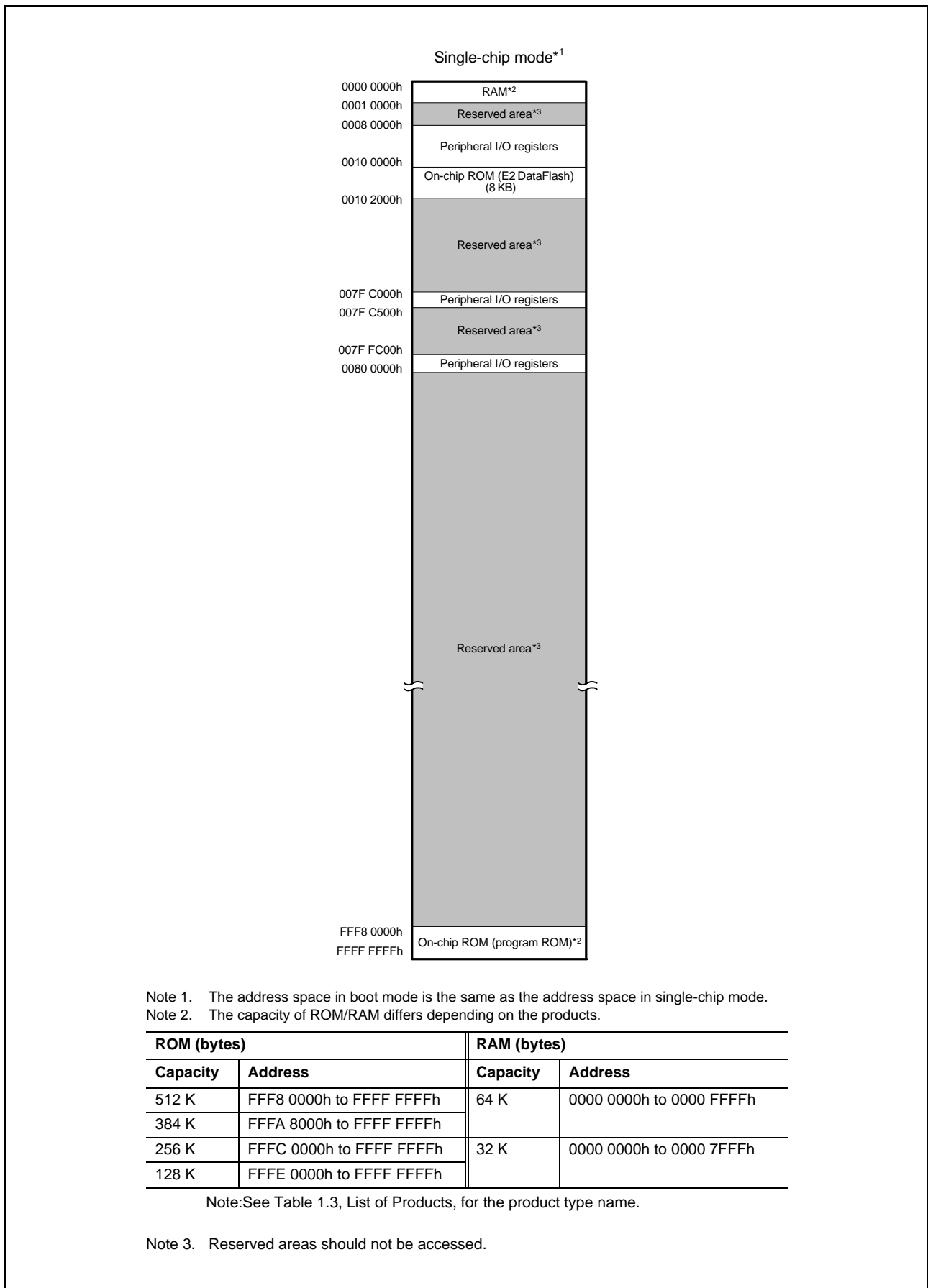


Figure 4.1 Memory Map

## 5. I/O Registers

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

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

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

### (2) Notes on writing to I/O registers

While writing to an I/O register, the CPU starts executing subsequent instructions before the I/O register write access is completed. This may cause the subsequent instructions to be executed before the write value is reflected in the operation. The examples below show how subsequent instructions must be executed after a write access to an I/O register is completed.

#### [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) set 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 in the I/O register and write it to a general register.
- Execute the operation using the value read.
- Execute the subsequent instruction.

#### Example of instructions

- 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
```

When executing an instruction after writing to multiple registers, only read the last I/O register written to and execute the instruction using that value; it is not necessary to execute the instruction using the values written to all the registers.

### (3) Number of cycles necessary for accessing I/O registers

See Table 5.1 for details on the number of clock cycles necessary for accessing I/O registers.

The number of access cycles to I/O registers is obtained by following equation.\*1

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

The number of bus cycles of internal peripheral buses 1 to 6 differs according to the register to be accessed.

When peripheral functions connected to internal peripheral buses 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) or bus access timing.

In the peripheral function unit, when the frequency ratio of ICLK is equal to or greater than that of PCLK (or FCLK), the sum of the number of bus cycles for internal main bus 1 and the number of the divided clock synchronization cycles will be one cycle of PCLK (or FCLK) at a maximum. Therefore, one PCLK (or FCLK) has been added to the number of access cycles shown in Table 5.1.

When the frequency ratio of ICLK is lower than that of PCLK (or FCLK), the subsequent bus access is started from the ICLK cycle following the completion of the access to the peripheral functions. Therefore, the access cycles are described on an ICLK basis.

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

### (4) Notes on sleep mode and mode transitions

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

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

Table 5.1 List of I/O Registers (Address Order) (1/23)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 0000h	SYSTEM	Mode Monitor Register	MDMONR	16	16	3 ICLK	section 3.
0008 0008h	SYSTEM	System Control Register 1	SYSCR1	16	16	3 ICLK	section 3.
0008 000Ch	SYSTEM	Standby Control Register	SBYCR	16	16	3 ICLK	section 11.
0008 0010h	SYSTEM	Module Stop Control Register A	MSTPCRA	32	32	3 ICLK	section 11.
0008 0014h	SYSTEM	Module Stop Control Register B	MSTPCRB	32	32	3 ICLK	section 11.
0008 0018h	SYSTEM	Module Stop Control Register C	MSTPCRC	32	32	3 ICLK	section 11.
0008 001Ch	SYSTEM	Module Stop Control Register D	MSTPCRD	32	32	3 ICLK	section 11.
0008 0020h	SYSTEM	System Clock Control Register	SCKCR	32	32	3 ICLK	section 9.
0008 0026h	SYSTEM	System Clock Control Register 3	SCKCR3	16	16	3 ICLK	section 9.
0008 0028h	SYSTEM	PLL Control Register	PLLCR	16	16	3 ICLK	section 9.
0008 002Ah	SYSTEM	PLL Control Register 2	PLLCR2	8	8	3 ICLK	section 9.
0008 002Ch	SYSTEM	USB-dedicated PLL Control Register	UPLLCR	16	16	3 ICLK	section 9.
0008 002Eh	SYSTEM	USB-dedicated PLL Control Register 2	UPLLCR2	8	8	3 ICLK	section 9.
0008 0032h	SYSTEM	Main Clock Oscillator Control Register	MOSCCR	8	8	3 ICLK	section 9.
0008 0033h	SYSTEM	Sub-Clock Oscillator Control Register	SOSCCR	8	8	3 ICLK	section 9.
0008 0034h	SYSTEM	Low-Speed On-Chip Oscillator Control Register	LOCOCR	8	8	3 ICLK	section 9.
0008 0035h	SYSTEM	IWDT-Dedicated On-Chip Oscillator Control Register	ILOCOCR	8	8	3 ICLK	section 9.
0008 0036h	SYSTEM	High-Speed On-Chip Oscillator Control Register	HOCOCR	8	8	3 ICLK	section 9.
0008 003Ch	SYSTEM	Oscillation Stabilization Flag Register	OSCOVFSR	8	8	3 ICLK	section 9.
0008 003Eh	SYSTEM	CLKOUT Output Control Register	CKOCR	16	16	3 ICLK	section 9.
0008 0040h	SYSTEM	Oscillation Stop Detection Control Register	OSTDCR	8	8	3 ICLK	section 9.
0008 0041h	SYSTEM	Oscillation Stop Detection Status Register	OSTDSR	8	8	3 ICLK	section 9.
0008 0050h	SYSTEM	LCD Source Clock Control Register	LCDSCLKCR	8	8	3 ICLK	section 9.
0008 0051h	SYSTEM	LCD Source Clock Control Register 2	LCDSCLKCR2	8	8	3 ICLK	section 9.
0008 00A0h	SYSTEM	Operating Power Control Register	OPCCR	8	8	3 ICLK	section 11.
0008 00A1h	SYSTEM	Sleep Mode Return Clock Source Switching Register	RSTCKCR	8	8	3 ICLK	section 11.
0008 00A2h	SYSTEM	Main Clock Oscillator Wait Control Register	MOSCWTCR	8	8	3 ICLK	section 9.
0008 00A5h	SYSTEM	High-Speed On-Chip Oscillator Wait Control Register	HOCOWTCR	8	8	3 ICLK	section 9.
0008 00AAh	SYSTEM	Sub Operating Power Control Register	SOPCCR	8	8	3 ICLK	section 11.
0008 00C0h	SYSTEM	Reset Status Register 2	RSTSR2	8	8	3 ICLK	section 6.
0008 00C2h	SYSTEM	Software Reset Register	SWRR	16	16	3 ICLK	section 6.
0008 00E0h	SYSTEM	Voltage Monitoring 1 Circuit Control Register 1	LVD1CR1	8	8	3 ICLK	section 8.
0008 00E1h	SYSTEM	Voltage Monitoring 1 Circuit Status Register	LVD1SR	8	8	3 ICLK	section 8.
0008 00E2h	SYSTEM	Voltage Monitoring 2 Circuit Control Register 1	LVD2CR1	8	8	3 ICLK	section 8.
0008 00E3h	SYSTEM	Voltage Monitoring 2 Circuit Status Register	LVD2SR	8	8	3 ICLK	section 8.
0008 03FEh	SYSTEM	Protect Register	PRCR	16	16	3 ICLK	section 12.
0008 1300h	BSC	Bus Error Status Clear Register	BERCLR	8	8	2 ICLK	section 15.
0008 1304h	BSC	Bus Error Monitoring Enable Register	BEREN	8	8	2 ICLK	section 15.
0008 1308h	BSC	Bus Error Status Register 1	BERSR1	8	8	2 ICLK	section 15.
0008 130Ah	BSC	Bus Error Status Register 2	BERSR2	16	16	2 ICLK	section 15.
0008 1310h	BSC	Bus Priority Control Register	BUSPRI	16	16	2 ICLK	section 15.
0008 2400h	DTC	DTC Control Register	DTCCR	8	8	2 ICLK	section 16.
0008 2404h	DTC	DTC Vector Base Register	DTCVBR	32	32	2 ICLK	section 16.
0008 2408h	DTC	DTC Address Mode Register	DTCADMOD	8	8	2 ICLK	section 16.
0008 240Ch	DTC	DTC Module Start Register	DTCST	8	8	2 ICLK	section 16.
0008 240Eh	DTC	DTC Status Register	DTCSTS	16	16	2 ICLK	section 16.
0008 7010h	ICU	Interrupt Request Register 016	IR016	8	8	2 ICLK	section 14.
0008 701Bh	ICU	Interrupt Request Register 027	IR027	8	8	2 ICLK	section 14.
0008 701Ch	ICU	Interrupt Request Register 028	IR028	8	8	2 ICLK	section 14.
0008 701Dh	ICU	Interrupt Request Register 029	IR029	8	8	2 ICLK	section 14.

**Table 5.1 List of I/O Registers (Address Order) (2/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 701Eh	ICU	Interrupt Request Register 030	IR030	8	8	2 ICLK	section 14.
0008 701Fh	ICU	Interrupt Request Register 031	IR031	8	8	2 ICLK	section 14.
0008 7020h	ICU	Interrupt Request Register 032	IR032	8	8	2 ICLK	section 14.
0008 7021h	ICU	Interrupt Request Register 033	IR033	8	8	2 ICLK	section 14.
0008 7022h	ICU	Interrupt Request Register 034	IR034	8	8	2 ICLK	section 14.
0008 7024h	ICU	Interrupt Request Register 036	IR036	8	8	2 ICLK	section 14.
0008 7025h	ICU	Interrupt Request Register 037	IR037	8	8	2 ICLK	section 14.
0008 7026h	ICU	Interrupt Request Register 038	IR038	8	8	2 ICLK	section 14.
0008 702Ch	ICU	Interrupt Request Register 044	IR044	8	8	2 ICLK	section 14.
0008 702Dh	ICU	Interrupt Request Register 045	IR045	8	8	2 ICLK	section 14.
0008 702Eh	ICU	Interrupt Request Register 046	IR046	8	8	2 ICLK	section 14.
0008 702Fh	ICU	Interrupt Request Register 047	IR047	8	8	2 ICLK	section 14.
0008 7039h	ICU	Interrupt Request Register 057	IR057	8	8	2 ICLK	section 14.
0008 703Ah	ICU	Interrupt Request Register 058	IR058	8	8	2 ICLK	section 14.
0008 703Bh	ICU	Interrupt Request Register 059	IR059	8	8	2 ICLK	section 14.
0008 703Ch	ICU	Interrupt Request Register 060	IR060	8	8	2 ICLK	section 14.
0008 703Dh	ICU	Interrupt Request Register 061	IR061	8	8	2 ICLK	section 14.
0008 703Eh	ICU	Interrupt Request Register 062	IR062	8	8	2 ICLK	section 14.
0008 703Fh	ICU	Interrupt Request Register 063	IR063	8	8	2 ICLK	section 14.
0008 7040h	ICU	Interrupt Request Register 064	IR064	8	8	2 ICLK	section 14.
0008 7041h	ICU	Interrupt Request Register 065	IR065	8	8	2 ICLK	section 14.
0008 7042h	ICU	Interrupt Request Register 066	IR066	8	8	2 ICLK	section 14.
0008 7043h	ICU	Interrupt Request Register 067	IR067	8	8	2 ICLK	section 14.
0008 7044h	ICU	Interrupt Request Register 068	IR068	8	8	2 ICLK	section 14.
0008 7045h	ICU	Interrupt Request Register 069	IR069	8	8	2 ICLK	section 14.
0008 7046h	ICU	Interrupt Request Register 070	IR070	8	8	2 ICLK	section 14.
0008 7047h	ICU	Interrupt Request Register 071	IR071	8	8	2 ICLK	section 14.
0008 7058h	ICU	Interrupt Request Register 088	IR088	8	8	2 ICLK	section 14.
0008 7059h	ICU	Interrupt Request Register 089	IR089	8	8	2 ICLK	section 14.
0008 705Ah	ICU	Interrupt Request Register 090	IR090	8	8	2 ICLK	section 14.
0008 705Ch	ICU	Interrupt Request Register 092	IR092	8	8	2 ICLK	section 14.
0008 705Dh	ICU	Interrupt Request Register 093	IR093	8	8	2 ICLK	section 14.
0008 7066h	ICU	Interrupt Request Register 102	IR102	8	8	2 ICLK	section 14.
0008 7067h	ICU	Interrupt Request Register 103	IR103	8	8	2 ICLK	section 14.
0008 706Ah	ICU	Interrupt Request Register 106	IR106	8	8	2 ICLK	section 14.
0008 706Ch	ICU	Interrupt Request Register 108	IR108	8	8	2 ICLK	section 14.
0008 706Dh	ICU	Interrupt Request Register 109	IR109	8	8	2 ICLK	section 14.
0008 706Eh	ICU	Interrupt Request Register 110	IR110	8	8	2 ICLK	section 14.
0008 7072h	ICU	Interrupt Request Register 114	IR114	8	8	2 ICLK	section 14.
0008 7073h	ICU	Interrupt Request Register 115	IR115	8	8	2 ICLK	section 14.
0008 7074h	ICU	Interrupt Request Register 116	IR116	8	8	2 ICLK	section 14.
0008 7075h	ICU	Interrupt Request Register 117	IR117	8	8	2 ICLK	section 14.
0008 7076h	ICU	Interrupt Request Register 118	IR118	8	8	2 ICLK	section 14.
0008 7077h	ICU	Interrupt Request Register 119	IR119	8	8	2 ICLK	section 14.
0008 7078h	ICU	Interrupt Request Register 120	IR120	8	8	2 ICLK	section 14.
0008 7079h	ICU	Interrupt Request Register 121	IR121	8	8	2 ICLK	section 14.
0008 707Ah	ICU	Interrupt Request Register 122	IR122	8	8	2 ICLK	section 14.
0008 707Bh	ICU	Interrupt Request Register 123	IR123	8	8	2 ICLK	section 14.
0008 707Ch	ICU	Interrupt Request Register 124	IR124	8	8	2 ICLK	section 14.
0008 707Dh	ICU	Interrupt Request Register 125	IR125	8	8	2 ICLK	section 14.
0008 707Eh	ICU	Interrupt Request Register 126	IR126	8	8	2 ICLK	section 14.
0008 707Fh	ICU	Interrupt Request Register 127	IR127	8	8	2 ICLK	section 14.

**Table 5.1 List of I/O Registers (Address Order) (3/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 7080h	ICU	Interrupt Request Register 128	IR128	8	8	2 ICLK	section 14.
0008 7081h	ICU	Interrupt Request Register 129	IR129	8	8	2 ICLK	section 14.
0008 7082h	ICU	Interrupt Request Register 130	IR130	8	8	2 ICLK	section 14.
0008 7083h	ICU	Interrupt Request Register 131	IR131	8	8	2 ICLK	section 14.
0008 7084h	ICU	Interrupt Request Register 132	IR132	8	8	2 ICLK	section 14.
0008 7085h	ICU	Interrupt Request Register 133	IR133	8	8	2 ICLK	section 14.
0008 7086h	ICU	Interrupt Request Register 134	IR134	8	8	2 ICLK	section 14.
0008 7087h	ICU	Interrupt Request Register 135	IR135	8	8	2 ICLK	section 14.
0008 7088h	ICU	Interrupt Request Register 136	IR136	8	8	2 ICLK	section 14.
0008 7089h	ICU	Interrupt Request Register 137	IR137	8	8	2 ICLK	section 14.
0008 708Ah	ICU	Interrupt Request Register 138	IR138	8	8	2 ICLK	section 14.
0008 708Bh	ICU	Interrupt Request Register 139	IR139	8	8	2 ICLK	section 14.
0008 708Ch	ICU	Interrupt Request Register 140	IR140	8	8	2 ICLK	section 14.
0008 708Dh	ICU	Interrupt Request Register 141	IR141	8	8	2 ICLK	section 14.
0008 70AAh	ICU	Interrupt Request Register 170	IR170	8	8	2 ICLK	section 14.
0008 70ABh	ICU	Interrupt Request Register 171	IR171	8	8	2 ICLK	section 14.
0008 70AEh	ICU	Interrupt Request Register 174	IR174	8	8	2 ICLK	section 14.
0008 70AFh	ICU	Interrupt Request Register 175	IR175	8	8	2 ICLK	section 14.
0008 70B0h	ICU	Interrupt Request Register 176	IR176	8	8	2 ICLK	section 14.
0008 70B1h	ICU	Interrupt Request Register 177	IR177	8	8	2 ICLK	section 14.
0008 70B2h	ICU	Interrupt Request Register 178	IR178	8	8	2 ICLK	section 14.
0008 70B3h	ICU	Interrupt Request Register 179	IR179	8	8	2 ICLK	section 14.
0008 70B4h	ICU	Interrupt Request Register 180	IR180	8	8	2 ICLK	section 14.
0008 70B5h	ICU	Interrupt Request Register 181	IR181	8	8	2 ICLK	section 14.
0008 70Bh6	ICU	Interrupt Request Register 182	IR182	8	8	2 ICLK	section 14.
0008 70B7h	ICU	Interrupt Request Register 183	IR183	8	8	2 ICLK	section 14.
0008 70B8h	ICU	Interrupt Request Register 184	IR184	8	8	2 ICLK	section 14.
0008 70B9h	ICU	Interrupt Request Register 185	IR185	8	8	2 ICLK	section 14.
0008 70BAh	ICU	Interrupt Request Register 186	IR186	8	8	2 ICLK	section 14.
0008 70BBh	ICU	Interrupt Request Register 187	IR187	8	8	2 ICLK	section 14.
0008 70BCh	ICU	Interrupt Request Register 188	IR188	8	8	2 ICLK	section 14.
0008 70BDh	ICU	Interrupt Request Register 189	IR189	8	8	2 ICLK	section 14.
0008 70D6h	ICU	Interrupt Request Register 214	IR214	8	8	2 ICLK	section 14.
0008 70D7h	ICU	Interrupt Request Register 215	IR215	8	8	2 ICLK	section 14.
0008 70D8h	ICU	Interrupt Request Register 216	IR216	8	8	2 ICLK	section 14.
0008 70D9h	ICU	Interrupt Request Register 217	IR217	8	8	2 ICLK	section 14.
0008 70DAh	ICU	Interrupt Request Register 218	IR218	8	8	2 ICLK	section 14.
0008 70DBh	ICU	Interrupt Request Register 219	IR219	8	8	2 ICLK	section 14.
0008 70DCh	ICU	Interrupt Request Register 220	IR220	8	8	2 ICLK	section 14.
0008 70DDh	ICU	Interrupt Request Register 221	IR221	8	8	2 ICLK	section 14.
0008 70DEh	ICU	Interrupt Request Register 222	IR222	8	8	2 ICLK	section 14.
0008 70DFh	ICU	Interrupt Request Register 223	IR223	8	8	2 ICLK	section 14.
0008 70E0h	ICU	Interrupt Request Register 224	IR224	8	8	2 ICLK	section 14.
0008 70E1h	ICU	Interrupt Request Register 225	IR225	8	8	2 ICLK	section 14.
0008 70E2h	ICU	Interrupt Request Register 226	IR226	8	8	2 ICLK	section 14.
0008 70E3h	ICU	Interrupt Request Register 227	IR227	8	8	2 ICLK	section 14.
0008 70E4h	ICU	Interrupt Request Register 228	IR228	8	8	2 ICLK	section 14.
0008 70E5h	ICU	Interrupt Request Register 229	IR229	8	8	2 ICLK	section 14.
0008 70E6h	ICU	Interrupt Request Register 230	IR230	8	8	2 ICLK	section 14.
0008 70E7h	ICU	Interrupt Request Register 231	IR231	8	8	2 ICLK	section 14.
0008 70E8h	ICU	Interrupt Request Register 232	IR232	8	8	2 ICLK	section 14.
0008 70E9h	ICU	Interrupt Request Register 233	IR233	8	8	2 ICLK	section 14.

**Table 5.1 List of I/O Registers (Address Order) (4/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 70EAh	ICU	Interrupt Request Register 234	IR234	8	8	2 ICLK	section 14.
0008 70EBh	ICU	Interrupt Request Register 235	IR235	8	8	2 ICLK	section 14.
0008 70ECh	ICU	Interrupt Request Register 236	IR236	8	8	2 ICLK	section 14.
0008 70EDh	ICU	Interrupt Request Register 237	IR237	8	8	2 ICLK	section 14.
0008 70EEh	ICU	Interrupt Request Register 238	IR238	8	8	2 ICLK	section 14.
0008 70EFh	ICU	Interrupt Request Register 239	IR239	8	8	2 ICLK	section 14.
0008 70F0h	ICU	Interrupt Request Register 240	IR240	8	8	2 ICLK	section 14.
0008 70F1h	ICU	Interrupt Request Register 241	IR241	8	8	2 ICLK	section 14.
0008 70F2h	ICU	Interrupt Request Register 242	IR242	8	8	2 ICLK	section 14.
0008 70F3h	ICU	Interrupt Request Register 243	IR243	8	8	2 ICLK	section 14.
0008 70F4h	ICU	Interrupt Request Register 244	IR244	8	8	2 ICLK	section 14.
0008 70F5h	ICU	Interrupt Request Register 245	IR245	8	8	2 ICLK	section 14.
0008 70F6h	ICU	Interrupt Request Register 246	IR246	8	8	2 ICLK	section 14.
0008 70F7h	ICU	Interrupt Request Register 247	IR247	8	8	2 ICLK	section 14.
0008 70F8h	ICU	Interrupt Request Register 248	IR248	8	8	2 ICLK	section 14.
0008 70F9h	ICU	Interrupt Request Register 249	IR249	8	8	2 ICLK	section 14.
0008 711Bh	ICU	DTC Activation Enable Register 027	DTCER027	8	8	2 ICLK	section 14.
0008 711Ch	ICU	DTC Activation Enable Register 028	DTCER028	8	8	2 ICLK	section 14.
0008 711Dh	ICU	DTC Activation Enable Register 029	DTCER029	8	8	2 ICLK	section 14.
0008 711Eh	ICU	DTC Activation Enable Register 030	DTCER030	8	8	2 ICLK	section 14.
0008 711Fh	ICU	DTC Activation Enable Register 031	DTCER031	8	8	2 ICLK	section 14.
0008 7124h	ICU	DTC Activation Enable Register 036	DTCER036	8	8	2 ICLK	section 14.
0008 7125h	ICU	DTC Activation Enable Register 037	DTCER037	8	8	2 ICLK	section 14.
0008 712Dh	ICU	DTC Activation Enable Register 045	DTCER045	8	8	2 ICLK	section 14.
0008 712Eh	ICU	DTC Activation Enable Register 046	DTCER046	8	8	2 ICLK	section 14.
0008 713Ah	ICU	DTC Activation Enable Register 058	DTCER058	8	8	2 ICLK	section 14.
0008 713Bh	ICU	DTC Activation Enable Register 059	DTCER059	8	8	2 ICLK	section 14.
0008 713Ch	ICU	DTC Activation Enable Register 060	DTCER060	8	8	2 ICLK	section 14.
0008 713Dh	ICU	DTC Activation Enable Register 061	DTCER061	8	8	2 ICLK	section 14.
0008 7140h	ICU	DTC Activation Enable Register 064	DTCER064	8	8	2 ICLK	section 14.
0008 7141h	ICU	DTC Activation Enable Register 065	DTCER065	8	8	2 ICLK	section 14.
0008 7142h	ICU	DTC Activation Enable Register 066	DTCER066	8	8	2 ICLK	section 14.
0008 7143h	ICU	DTC Activation Enable Register 067	DTCER067	8	8	2 ICLK	section 14.
0008 7144h	ICU	DTC Activation Enable Register 068	DTCER068	8	8	2 ICLK	section 14.
0008 7145h	ICU	DTC Activation Enable Register 069	DTCER069	8	8	2 ICLK	section 14.
0008 7146h	ICU	DTC Activation Enable Register 070	DTCER070	8	8	2 ICLK	section 14.
0008 7147h	ICU	DTC Activation Enable Register 071	DTCER071	8	8	2 ICLK	section 14.
0008 7166h	ICU	DTC Activation Enable Register 102	DTCER102	8	8	2 ICLK	section 14.
0008 7167h	ICU	DTC Activation Enable Register 103	DTCER103	8	8	2 ICLK	section 14.
0008 716Ah	ICU	DTC Activation Enable Register 106	DTCER106	8	8	2 ICLK	section 14.
0008 716Dh	ICU	DTC Activation Enable Register 109	DTCER109	8	8	2 ICLK	section 14.
0008 716Eh	ICU	DTC Activation Enable Register 110	DTCER110	8	8	2 ICLK	section 14.
0008 7172h	ICU	DTC Activation Enable Register 114	DTCER114	8	8	2 ICLK	section 14.
0008 7173h	ICU	DTC Activation Enable Register 115	DTCER115	8	8	2 ICLK	section 14.
0008 7174h	ICU	DTC Activation Enable Register 116	DTCER116	8	8	2 ICLK	section 14.
0008 7175h	ICU	DTC Activation Enable Register 117	DTCER117	8	8	2 ICLK	section 14.
0008 7179h	ICU	DTC Activation Enable Register 121	DTCER121	8	8	2 ICLK	section 14.
0008 717Ah	ICU	DTC Activation Enable Register 122	DTCER122	8	8	2 ICLK	section 14.
0008 717Dh	ICU	DTC Activation Enable Register 125	DTCER125	8	8	2 ICLK	section 14.
0008 717Eh	ICU	DTC Activation Enable Register 126	DTCER126	8	8	2 ICLK	section 14.
0008 7181h	ICU	DTC Activation Enable Register 129	DTCER129	8	8	2 ICLK	section 14.
0008 7182h	ICU	DTC Activation Enable Register 130	DTCER130	8	8	2 ICLK	section 14.

**Table 5.1 List of I/O Registers (Address Order) (5/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 7183h	ICU	DTC Activation Enable Register 131	DTCER131	8	8	2 ICLK	section 14.
0008 7184h	ICU	DTC Activation Enable Register 132	DTCER132	8	8	2 ICLK	section 14.
0008 7186h	ICU	DTC Activation Enable Register 134	DTCER134	8	8	2 ICLK	section 14.
0008 7187h	ICU	DTC Activation Enable Register 135	DTCER135	8	8	2 ICLK	section 14.
0008 7188h	ICU	DTC Activation Enable Register 136	DTCER136	8	8	2 ICLK	section 14.
0008 7189h	ICU	DTC Activation Enable Register 137	DTCER137	8	8	2 ICLK	section 14.
0008 718Ah	ICU	DTC Activation Enable Register 138	DTCER138	8	8	2 ICLK	section 14.
0008 718Bh	ICU	DTC Activation Enable Register 139	DTCER139	8	8	2 ICLK	section 14.
0008 718Ch	ICU	DTC Activation Enable Register 140	DTCER140	8	8	2 ICLK	section 14.
0008 718Dh	ICU	DTC Activation Enable Register 141	DTCER141	8	8	2 ICLK	section 14.
0008 71AEh	ICU	DTC Activation Enable Register 174	DTCER174	8	8	2 ICLK	section 14.
0008 71AFh	ICU	DTC Activation Enable Register 175	DTCER175	8	8	2 ICLK	section 14.
0008 71B1h	ICU	DTC Activation Enable Register 177	DTCER177	8	8	2 ICLK	section 14.
0008 71B2h	ICU	DTC Activation Enable Register 178	DTCER178	8	8	2 ICLK	section 14.
0008 71B4h	ICU	DTC Activation Enable Register 180	DTCER180	8	8	2 ICLK	section 14.
0008 71B5h	ICU	DTC Activation Enable Register 181	DTCER181	8	8	2 ICLK	section 14.
0008 71B7h	ICU	DTC Activation Enable Register 183	DTCER183	8	8	2 ICLK	section 14.
0008 71B8h	ICU	DTC Activation Enable Register 184	DTCER184	8	8	2 ICLK	section 14.
0008 71BBh	ICU	DTC Activation Enable Register 187	DTCER187	8	8	2 ICLK	section 14.
0008 71BCh	ICU	DTC Activation Enable Register 188	DTCER188	8	8	2 ICLK	section 14.
0008 71D7h	ICU	DTC Activation Enable Register 215	DTCER215	8	8	2 ICLK	section 14.
0008 71D8h	ICU	DTC Activation Enable Register 216	DTCER216	8	8	2 ICLK	section 14.
0008 71DBh	ICU	DTC Activation Enable Register 219	DTCER219	8	8	2 ICLK	section 14.
0008 71DCh	ICU	DTC Activation Enable Register 220	DTCER220	8	8	2 ICLK	section 14.
0008 71DFh	ICU	DTC Activation Enable Register 223	DTCER223	8	8	2 ICLK	section 14.
0008 71E0h	ICU	DTC Activation Enable Register 224	DTCER224	8	8	2 ICLK	section 14.
0008 71E3h	ICU	DTC Activation Enable Register 227	DTCER227	8	8	2 ICLK	section 14.
0008 71E4h	ICU	DTC Activation Enable Register 228	DTCER228	8	8	2 ICLK	section 14.
0008 71E7h	ICU	DTC Activation Enable Register 231	DTCER231	8	8	2 ICLK	section 14.
0008 71E8h	ICU	DTC Activation Enable Register 232	DTCER232	8	8	2 ICLK	section 14.
0008 71EBh	ICU	DTC Activation Enable Register 235	DTCER235	8	8	2 ICLK	section 14.
0008 71ECh	ICU	DTC Activation Enable Register 236	DTCER236	8	8	2 ICLK	section 14.
0008 71EFh	ICU	DTC Activation Enable Register 239	DTCER239	8	8	2 ICLK	section 14.
0008 71F0h	ICU	DTC Activation Enable Register 240	DTCER240	8	8	2 ICLK	section 14.
0008 71F7h	ICU	DTC Activation Enable Register 247	DTCER247	8	8	2 ICLK	section 14.
0008 71F8h	ICU	DTC Activation Enable Register 248	DTCER248	8	8	2 ICLK	section 14.
0008 7202h	ICU	Interrupt Request Enable Register 02	IER02	8	8	2 ICLK	section 14.
0008 7203h	ICU	Interrupt Request Enable Register 03	IER03	8	8	2 ICLK	section 14.
0008 7204h	ICU	Interrupt Request Enable Register 04	IER04	8	8	2 ICLK	section 14.
0008 7205h	ICU	Interrupt Request Enable Register 05	IER05	8	8	2 ICLK	section 14.
0008 7207h	ICU	Interrupt Request Enable Register 07	IER07	8	8	2 ICLK	section 14.
0008 7208h	ICU	Interrupt Request Enable Register 08	IER08	8	8	2 ICLK	section 14.
0008 720Bh	ICU	Interrupt Request Enable Register 0B	IER0B	8	8	2 ICLK	section 14.
0008 720Ch	ICU	Interrupt Request Enable Register 0C	IER0C	8	8	2 ICLK	section 14.
0008 720Dh	ICU	Interrupt Request Enable Register 0D	IER0D	8	8	2 ICLK	section 14.
0008 720Eh	ICU	Interrupt Request Enable Register 0E	IER0E	8	8	2 ICLK	section 14.
0008 720Fh	ICU	Interrupt Request Enable Register 0F	IER0F	8	8	2 ICLK	section 14.
0008 7210h	ICU	Interrupt Request Enable Register 10	IER10	8	8	2 ICLK	section 14.
0008 7211h	ICU	Interrupt Request Enable Register 11	IER11	8	8	2 ICLK	section 14.
0008 7215h	ICU	Interrupt Request Enable Register 15	IER15	8	8	2 ICLK	section 14.
0008 7216h	ICU	Interrupt Request Enable Register 16	IER16	8	8	2 ICLK	section 14.
0008 7217h	ICU	Interrupt Request Enable Register 17	IER17	8	8	2 ICLK	section 14.



**Table 5.1 List of I/O Registers (Address Order) (6/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 721Ah	ICU	Interrupt Request Enable Register 1A	IER1A	8	8	2 ICLK	section 14.
0008 721Bh	ICU	Interrupt Request Enable Register 1B	IER1B	8	8	2 ICLK	section 14.
0008 721Ch	ICU	Interrupt Request Enable Register 1C	IER1C	8	8	2 ICLK	section 14.
0008 721Dh	ICU	Interrupt Request Enable Register 1D	IER1D	8	8	2 ICLK	section 14.
0008 721Eh	ICU	Interrupt Request Enable Register 1E	IER1E	8	8	2 ICLK	section 14.
0008 721Fh	ICU	Interrupt Request Enable Register 1F	IER1F	8	8	2 ICLK	section 14.
0008 72E0h	ICU	Software Interrupt Activation Register	SWINTR	8	8	2 ICLK	section 14.
0008 72F0h	ICU	Fast Interrupt Set Register	FIR	16	16	2 ICLK	section 14.
0008 7300h	ICU	Interrupt Source Priority Register 000	IPR000	8	8	2 ICLK	section 14.
0008 7303h	ICU	Interrupt Source Priority Register 003	IPR003	8	8	2 ICLK	section 14.
0008 7304h	ICU	Interrupt Source Priority Register 004	IPR004	8	8	2 ICLK	section 14.
0008 7305h	ICU	Interrupt Source Priority Register 005	IPR005	8	8	2 ICLK	section 14.
0008 7306h	ICU	Interrupt Source Priority Register 006	IPR006	8	8	2 ICLK	section 14.
0008 7307h	ICU	Interrupt Source Priority Register 007	IPR007	8	8	2 ICLK	section 14.
0008 7320h	ICU	Interrupt Source Priority Register 032	IPR032	8	8	2 ICLK	section 14.
0008 7321h	ICU	Interrupt Source Priority Register 033	IPR033	8	8	2 ICLK	section 14.
0008 7322h	ICU	Interrupt Source Priority Register 034	IPR034	8	8	2 ICLK	section 14.
0008 7324h	ICU	Interrupt Source Priority Register 036	IPR036	8	8	2 ICLK	section 14.
0008 7325h	ICU	Interrupt Source Priority Register 037	IPR037	8	8	2 ICLK	section 14.
0008 7326h	ICU	Interrupt Source Priority Register 038	IPR038	8	8	2 ICLK	section 14.
0008 732Ch	ICU	Interrupt Source Priority Register 044	IPR044	8	8	2 ICLK	section 14.
0008 7339h	ICU	Interrupt Source Priority Register 057	IPR057	8	8	2 ICLK	section 14.
0008 733Ah	ICU	Interrupt Source Priority Register 058	IPR058	8	8	2 ICLK	section 14.
0008 733Bh	ICU	Interrupt Source Priority Register 059	IPR059	8	8	2 ICLK	section 14.
0008 733Ch	ICU	Interrupt Source Priority Register 060	IPR060	8	8	2 ICLK	section 14.
0008 733Fh	ICU	Interrupt Source Priority Register 063	IPR063	8	8	2 ICLK	section 14.
0008 7340h	ICU	Interrupt Source Priority Register 064	IPR064	8	8	2 ICLK	section 14.
0008 7341h	ICU	Interrupt Source Priority Register 065	IPR065	8	8	2 ICLK	section 14.
0008 7342h	ICU	Interrupt Source Priority Register 066	IPR066	8	8	2 ICLK	section 14.
0008 7343h	ICU	Interrupt Source Priority Register 067	IPR067	8	8	2 ICLK	section 14.
0008 7344h	ICU	Interrupt Source Priority Register 068	IPR068	8	8	2 ICLK	section 14.
0008 7345h	ICU	Interrupt Source Priority Register 069	IPR069	8	8	2 ICLK	section 14.
0008 7346h	ICU	Interrupt Source Priority Register 070	IPR070	8	8	2 ICLK	section 14.
0008 7347h	ICU	Interrupt Source Priority Register 071	IPR071	8	8	2 ICLK	section 14.
0008 7358h	ICU	Interrupt Source Priority Register 088	IPR088	8	8	2 ICLK	section 14.
0008 7359h	ICU	Interrupt Source Priority Register 089	IPR089	8	8	2 ICLK	section 14.
0008 735Ah	ICU	Interrupt Source Priority Register 090	IPR090	8	8	2 ICLK	section 14.
0008 735Ch	ICU	Interrupt Source Priority Register 092	IPR092	8	8	2 ICLK	section 14.
0008 735Dh	ICU	Interrupt Source Priority Register 093	IPR093	8	8	2 ICLK	section 14.
0008 7366h	ICU	Interrupt Source Priority Register 102	IPR102	8	8	2 ICLK	section 14.
0008 7367h	ICU	Interrupt Source Priority Register 103	IPR103	8	8	2 ICLK	section 14.
0008 736Ah	ICU	Interrupt Source Priority Register 106	IPR106	8	8	2 ICLK	section 14.
0008 7372h	ICU	Interrupt Source Priority Register 114	IPR114	8	8	2 ICLK	section 14.
0008 7376h	ICU	Interrupt Source Priority Register 118	IPR118	8	8	2 ICLK	section 14.
0008 7379h	ICU	Interrupt Source Priority Register 121	IPR121	8	8	2 ICLK	section 14.
0008 737Bh	ICU	Interrupt Source Priority Register 123	IPR123	8	8	2 ICLK	section 14.
0008 737Dh	ICU	Interrupt Source Priority Register 125	IPR125	8	8	2 ICLK	section 14.
0008 737Fh	ICU	Interrupt Source Priority Register 127	IPR127	8	8	2 ICLK	section 14.
0008 7381h	ICU	Interrupt Source Priority Register 129	IPR129	8	8	2 ICLK	section 14.
0008 7385h	ICU	Interrupt Source Priority Register 133	IPR133	8	8	2 ICLK	section 14.
0008 7386h	ICU	Interrupt Source Priority Register 134	IPR134	8	8	2 ICLK	section 14.
0008 738Ah	ICU	Interrupt Source Priority Register 138	IPR138	8	8	2 ICLK	section 14.

**Table 5.1 List of I/O Registers (Address Order) (7/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 738Bh	ICU	Interrupt Source Priority Register 139	IPR139	8	8	2 ICLK	section 14.
0008 73AAh	ICU	Interrupt Source Priority Register 170	IPR170	8	8	2 ICLK	section 14.
0008 73ABh	ICU	Interrupt Source Priority Register 171	IPR171	8	8	2 ICLK	section 14.
0008 73AEh	ICU	Interrupt Source Priority Register 174	IPR174	8	8	2 ICLK	section 14.
0008 73B1h	ICU	Interrupt Source Priority Register 177	IPR177	8	8	2 ICLK	section 14.
0008 73B4h	ICU	Interrupt Source Priority Register 180	IPR180	8	8	2 ICLK	section 14.
0008 73B7h	ICU	Interrupt Source Priority Register 183	IPR183	8	8	2 ICLK	section 14.
0008 73BAh	ICU	Interrupt Source Priority Register 186	IPR186	8	8	2 ICLK	section 14.
0008 73D6h	ICU	Interrupt Source Priority Register 214	IPR214	8	8	2 ICLK	section 14.
0008 73DAh	ICU	Interrupt Source Priority Register 218	IPR218	8	8	2 ICLK	section 14.
0008 73DEh	ICU	Interrupt Source Priority Register 222	IPR222	8	8	2 ICLK	section 14.
0008 73E2h	ICU	Interrupt Source Priority Register 226	IPR226	8	8	2 ICLK	section 14.
0008 73E6h	ICU	Interrupt Source Priority Register 230	IPR230	8	8	2 ICLK	section 14.
0008 73EAh	ICU	Interrupt Source Priority Register 234	IPR234	8	8	2 ICLK	section 14.
0008 73EEh	ICU	Interrupt Source Priority Register 238	IPR238	8	8	2 ICLK	section 14.
0008 73F2h	ICU	Interrupt Source Priority Register 242	IPR242	8	8	2 ICLK	section 14.
0008 73F3h	ICU	Interrupt Source Priority Register 243	IPR243	8	8	2 ICLK	section 14.
0008 73F4h	ICU	Interrupt Source Priority Register 244	IPR244	8	8	2 ICLK	section 14.
0008 73F5h	ICU	Interrupt Source Priority Register 245	IPR245	8	8	2 ICLK	section 14.
0008 73F6h	ICU	Interrupt Source Priority Register 246	IPR246	8	8	2 ICLK	section 14.
0008 73F7h	ICU	Interrupt Source Priority Register 247	IPR247	8	8	2 ICLK	section 14.
0008 73F8h	ICU	Interrupt Source Priority Register 248	IPR248	8	8	2 ICLK	section 14.
0008 73F9h	ICU	Interrupt Source Priority Register 249	IPR249	8	8	2 ICLK	section 14.
0008 7500h	ICU	IRQ Control Register 0	IRQCR0	8	8	2 ICLK	section 14.
0008 7501h	ICU	IRQ Control Register 1	IRQCR1	8	8	2 ICLK	section 14.
0008 7502h	ICU	IRQ Control Register 2	IRQCR2	8	8	2 ICLK	section 14.
0008 7503h	ICU	IRQ Control Register 3	IRQCR3	8	8	2 ICLK	section 14.
0008 7504h	ICU	IRQ Control Register 4	IRQCR4	8	8	2 ICLK	section 14.
0008 7505h	ICU	IRQ Control Register 5	IRQCR5	8	8	2 ICLK	section 14.
0008 7506h	ICU	IRQ Control Register 6	IRQCR6	8	8	2 ICLK	section 14.
0008 7507h	ICU	IRQ Control Register 7	IRQCR7	8	8	2 ICLK	section 14.
0008 7510h	ICU	IRQ Pin Digital Filter Enable Register 0	IRQFLTE0	8	8	2 ICLK	section 14.
0008 7514h	ICU	IRQ Pin Digital Filter Setting Register 0	IRQFLTC0	16	16	2 ICLK	section 14.
0008 7580h	ICU	Non-Maskable Interrupt Status Register	NMISR	8	8	2 ICLK	section 14.
0008 7581h	ICU	Non-Maskable Interrupt Enable Register	NMIER	8	8	2 ICLK	section 14.
0008 7582h	ICU	Non-Maskable Interrupt Status Clear Register	NMICLR	8	8	2 ICLK	section 14.
0008 7583h	ICU	NMI Pin Interrupt Control Register	NMICR	8	8	2 ICLK	section 14.
0008 7590h	ICU	NMI Pin Digital Filter Enable Register	NMIFLTE	8	8	2 ICLK	section 14.
0008 7594h	ICU	NMI Pin Digital Filter Setting Register	NMIFLTC	8	8	2 ICLK	section 14.
0008 8000h	CMT	Compare Match Timer Start Register 0	CMSTR0	16	16	2 or 3 PCLKB	section 23.
0008 8002h	CMT0	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	section 23.
0008 8004h	CMT0	Compare Match Timer Counter	CMCNT	16	16	2 or 3 PCLKB	section 23.
0008 8006h	CMT0	Compare Match Timer Constant Register	CMCOR	16	16	2 or 3 PCLKB	section 23.
0008 8008h	CMT1	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	section 23.
0008 800Ah	CMT1	Compare Match Timer Counter	CMCNT	16	16	2 or 3 PCLKB	section 23.
0008 800Ch	CMT1	Compare Match Timer Constant Register	CMCOR	16	16	2 or 3 PCLKB	section 23.
0008 8010h	CMT	Compare Match Timer Start Register1	CMSTR1	16	16	2 or 3 PCLKB	section 23.
0008 8012h	CMT2	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	section 23.
0008 8014h	CMT2	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	section 23.
0008 8016h	CMT2	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	section 23.
0008 8018h	CMT3	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	section 23.
0008 801Ah	CMT3	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	section 23.

**Table 5.1 List of I/O Registers (Address Order) (8/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 801Ch	CMT3	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	section 23.
0008 8030h	IWDT	IWDT Refresh Register	IWDTRR	8	8	2 or 3 PCLKB	section 25.
0008 8032h	IWDT	IWDT Control Register	IWDTCR	16	16	2 or 3 PCLKB	section 25.
0008 8034h	IWDT	IWDT Status Register	IWDTSR	16	16	2 or 3 PCLKB	section 25.
0008 8036h	IWDT	IWDT Reset Control Register	IWDTRCR	8	8	2 or 3 PCLKB	section 25.
0008 8038h	IWDT	IWDT Count Stop Control Register	IWDTCSTPR	8	8	2 or 3 PCLKB	section 25.
0008 8040h	R12DA	D/A Data Register 0	DADR0	16	16	2 or 3 PCLKB	section 36.
0008 8042h	R12DA	D/A Data Register 1	DADR1	16	16	2 or 3 PCLKB	section 36.
0008 8044h	R12DA	D/A Control Register	DACR	8	8	2 or 3 PCLKB	section 36.
0008 8045h	R12DA	DADRm Format Select Register	DADPR	8	8	2 or 3 PCLKB	section 36.
0008 8046h	R12DA	D/A A/D Synchronous Start Control Register	DAADSCR	8	8	2 or 3 PCLKB	section 36.
0008 8047h	R12DA	D/A VREF Control Register	DAVREFCR	8	8	2 or 3 PCLKB	section 36.
0008 8200h	TMR0	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 22.
0008 8201h	TMR1	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 22.
0008 8202h	TMR0	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	section 22.
0008 8203h	TMR1	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	section 22.
0008 8204h	TMR0	Time Constant Register A	TCORA	8	8	2 or 3 PCLKB	section 22.
0008 8205h	TMR1	Time Constant Register A	TCORA	8	8 <sup>*1</sup>	2 or 3 PCLKB	section 22.
0008 8206h	TMR0	Time Constant Register B	TCORB	8	8	2 or 3 PCLKB	section 22.
0008 8207h	TMR1	Time Constant Register B	TCORB	8	8 <sup>*1</sup>	2 or 3 PCLKB	section 22.
0008 8208h	TMR0	Timer Counter	TCNT	8	8	2 or 3 PCLKB	section 22.
0008 8209h	TMR1	Timer Counter	TCNT	8	8 <sup>*1</sup>	2 or 3 PCLKB	section 22.
0008 820Ah	TMR0	Timer Counter Control Register	TCCR	8	8	2 or 3 PCLKB	section 22.
0008 820Bh	TMR1	Timer Counter Control Register	TCCR	8	8 <sup>*1</sup>	2 or 3 PCLKB	section 22.
0008 820Ch	TMR0	Time Count Start Register	TCSTR	8	8	2 or 3 PCLKB	section 22.
0008 8210h	TMR2	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 22.
0008 8211h	TMR3	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 22.
0008 8212h	TMR2	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	section 22.
0008 8213h	TMR3	Timer Control/Status Register	TCSR	8	8	2 or 3 PCLKB	section 22.
0008 8214h	TMR2	Time Constant Register A	TCORA	8	8	2 or 3 PCLKB	section 22.
0008 8215h	TMR3	Time Constant Register A	TCORA	8	8 <sup>*1</sup>	2 or 3 PCLKB	section 22.
0008 8216h	TMR2	Time Constant Register B	TCORB	8	8	2 or 3 PCLKB	section 22.
0008 8217h	TMR3	Time Constant Register B	TCORB	8	8 <sup>*1</sup>	2 or 3 PCLKB	section 22.
0008 8218h	TMR2	Timer Counter	TCNT	8	8	2 or 3 PCLKB	section 22.
0008 8219h	TMR3	Timer Counter	TCNT	8	8 <sup>*1</sup>	2 or 3 PCLKB	section 22.
0008 821Ah	TMR2	Timer Control Register	TCCR	8	8	2 or 3 PCLKB	section 22.
0008 821Bh	TMR3	Timer Control Register	TCCR	8	8 <sup>*1</sup>	2 or 3 PCLKB	section 22.
0008 821Ch	TMR2	Time Count Start Register	TCSTR	8	8	2 or 3 PCLKB	section 22.
0008 8280h	CRC	CRC Control Register	CRCCR	8	8	2 or 3 PCLKB	section 32.
0008 8281h	CRC	CRC Data Input Register	CRCDIR	8	8	2 or 3 PCLKB	section 32.
0008 8282h	CRC	CRC Data Output Register	CRCDOR	16	16	2 or 3 PCLKB	section 32.
0008 8300h	RIIC0	I <sup>2</sup> C Bus Control Register 1	ICCR1	8	8	2 or 3 PCLKB	section 29.
0008 8301h	RIIC0	I <sup>2</sup> C Bus Control Register 2	ICCR2	8	8	2 or 3 PCLKB	section 29.
0008 8302h	RIIC0	I <sup>2</sup> C Bus Mode Register 1	ICMR1	8	8	2 or 3 PCLKB	section 29.
0008 8303h	RIIC0	I <sup>2</sup> C Bus Mode Register 2	ICMR2	8	8	2 or 3 PCLKB	section 29.
0008 8304h	RIIC0	I <sup>2</sup> C Bus Mode Register 3	ICMR3	8	8	2 or 3 PCLKB	section 29.
0008 8305h	RIIC0	I <sup>2</sup> C Bus Function Enable Register	ICFER	8	8	2 or 3 PCLKB	section 29.
0008 8306h	RIIC0	I <sup>2</sup> C Bus Status Enable Register	ICSER	8	8	2 or 3 PCLKB	section 29.
0008 8307h	RIIC0	I <sup>2</sup> C Bus Interrupt Enable Register	ICIER	8	8	2 or 3 PCLKB	section 29.
0008 8308h	RIIC0	I <sup>2</sup> C Bus Status Register 1	ICSR1	8	8	2 or 3 PCLKB	section 29.
0008 8309h	RIIC0	I <sup>2</sup> C Bus Status Register 2	ICSR2	8	8	2 or 3 PCLKB	section 29.
0008 830Ah	RIIC0	Slave Address Register L0	SARL0	8	8	2 or 3 PCLKB	section 29.

**Table 5.1 List of I/O Registers (Address Order) (9/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 830Ah	RIIC0	Timeout Internal Counter L	TMOCNTRL	8	8	2 or 3 PCLKB	section 29.
0008 830Bh	RIIC0	Slave Address Register U0	SARU0	8	8	2 or 3 PCLKB	section 29.
0008 830Bh	RIIC0	Timeout Internal Counter U	TMOCNTU	8	8 *2	2 or 3 PCLKB	section 29.
0008 830Ch	RIIC0	Slave Address Register L1	SARL1	8	8	2 or 3 PCLKB	section 29.
0008 830Dh	RIIC0	Slave Address Register U1	SARU1	8	8	2 or 3 PCLKB	section 29.
0008 830Eh	RIIC0	Slave Address Register L2	SARL2	8	8	2 or 3 PCLKB	section 29.
0008 830Fh	RIIC0	Slave Address Register U2	SARU2	8	8	2 or 3 PCLKB	section 29.
0008 8310h	RIIC0	I <sup>2</sup> C Bus Bit Rate Low-Level Register	ICBRL	8	8	2 or 3 PCLKB	section 29.
0008 8311h	RIIC0	I <sup>2</sup> C Bus Bit Rate High-Level Register	ICBRH	8	8	2 or 3 PCLKB	section 29.
0008 8312h	RIIC0	I <sup>2</sup> C Bus Transmit Data Register	ICDRT	8	8	2 or 3 PCLKB	section 29.
0008 8313h	RIIC0	I <sup>2</sup> C Bus Receive Data Register	ICDRR	8	8	2 or 3 PCLKB	section 29.
0008 8380h	RSPI0	RSPI Control Register	SPCR	8	8	2 or 3 PCLKB	section 31.
0008 8381h	RSPI0	RSPI Slave Select Polarity Register	SSLP	8	8	2 or 3 PCLKB	section 31.
0008 8382h	RSPI0	RSPI Pin Control Register	SPPCR	8	8	2 or 3 PCLKB	section 31.
0008 8383h	RSPI0	RSPI Status Register	SPSR	8	8	2 or 3 PCLKB	section 31.
0008 8384h	RSPI0	RSPI Data Register	SPDR	32	16, 32	2 or 3 PCLKB/2ICLK	section 31.
0008 8388h	RSPI0	RSPI Sequence Control Register	SPSCR	8	8	2 or 3 PCLKB	section 31.
0008 8389h	RSPI0	RSPI Sequence Status Register	SPSSR	8	8	2 or 3 PCLKB	section 31.
0008 838Ah	RSPI0	RSPI Bit Rate Register	SPBR	8	8	2 or 3 PCLKB	section 31.
0008 838Bh	RSPI0	RSPI Data Control Register	SPDCR	8	8	2 or 3 PCLKB	section 31.
0008 838Ch	RSPI0	RSPI Clock Delay Register	SPCKD	8	8	2 or 3 PCLKB	section 31.
0008 838Dh	RSPI0	RSPI Slave Select Negation Delay Register	SSLND	8	8	2 or 3 PCLKB	section 31.
0008 838Eh	RSPI0	RSPI Next-Access Delay Register	SPND	8	8	2 or 3 PCLKB	section 31.
0008 838Fh	RSPI0	RSPI Control Register 2	SPCR2	8	8	2 or 3 PCLKB	section 31.
0008 8390h	RSPI0	RSPI Command Register 0	SPCMD0	16	16	2 or 3 PCLKB	section 31.
0008 8392h	RSPI0	RSPI Command Register 1	SPCMD1	16	16	2 or 3 PCLKB	section 31.
0008 8394h	RSPI0	RSPI Command Register 2	SPCMD2	16	16	2 or 3 PCLKB	section 31.
0008 8396h	RSPI0	RSPI Command Register 3	SPCMD3	16	16	2 or 3 PCLKB	section 31.
0008 8398h	RSPI0	RSPI Command Register 4	SPCMD4	16	16	2 or 3 PCLKB	section 31.
0008 839Ah	RSPI0	RSPI Command Register 5	SPCMD5	16	16	2 or 3 PCLKB	section 31.
0008 839Ch	RSPI0	RSPI Command Register 6	SPCMD6	16	16	2 or 3 PCLKB	section 31.
0008 839Eh	RSPI0	RSPI Command Register 7	SPCMD7	16	16	2 or 3 PCLKB	section 31.
0008 8410h	IRDA	IrDA Control Register	IRCR	8	8	2 or 3 PCLKB	section 28.
0008 8600h	MTU3	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 20.
0008 8601h	MTU4	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 20.
0008 8602h	MTU3	Timer Mode Register	TMDR	8	8	2 or 3 PCLKB	section 20.
0008 8603h	MTU4	Timer Mode Register	TMDR	8	8	2 or 3 PCLKB	section 20.
0008 8604h	MTU3	Timer I/O Control Register H	TIORH	8	8	2 or 3 PCLKB	section 20.
0008 8605h	MTU3	Timer I/O Control Register L	TIORL	8	8	2 or 3 PCLKB	section 20.
0008 8606h	MTU4	Timer I/O Control Register H	TIORH	8	8	2 or 3 PCLKB	section 20.
0008 8607h	MTU4	Timer I/O Control Register L	TIORL	8	8	2 or 3 PCLKB	section 20.
0008 8608h	MTU3	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKB	section 20.
0008 8609h	MTU4	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKB	section 20.
0008 860Ah	MTU	Timer Output Master Enable Register	TOER	8	8	2 or 3 PCLKB	section 20.
0008 860Dh	MTU	Timer Gate Control Register	TGCR	8	8	2 or 3 PCLKB	section 20.
0008 860Eh	MTU	Timer Output Control Register 1	TOCR1	8	8	2 or 3 PCLKB	section 20.
0008 860Fh	MTU	Timer Output Control Register 2	TOCR2	8	8	2 or 3 PCLKB	section 20.
0008 8610h	MTU3	Timer Counter	TCNT	16	16	2 or 3 PCLKB	section 20.
0008 8612h	MTU4	Timer Counter	TCNT	16	16	2 or 3 PCLKB	section 20.
0008 8614h	MTU	Timer Cycle Data Register	TCDR	16	16	2 or 3 PCLKB	section 20.
0008 8616h	MTU	Timer Dead Time Data Register	TDDR	16	16	2 or 3 PCLKB	section 20.
0008 8618h	MTU3	Timer General Register A	TGRA	16	16	2 or 3 PCLKB	section 20.

**Table 5.1 List of I/O Registers (Address Order) (10/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 861Ah	MTU3	Timer General Register B	TGRB	16	16	2 or 3 PCLKB	section 20.
0008 861Ch	MTU4	Timer General Register A	TGRA	16	16	2 or 3 PCLKB	section 20.
0008 861Eh	MTU4	Timer General Register B	TGRB	16	16	2 or 3 PCLKB	section 20.
0008 8620h	MTU	Timer Subcounter	TCNTS	16	16	2 or 3 PCLKB	section 20.
0008 8622h	MTU	Timer Cycle Buffer Register	TCBR	16	16	2 or 3 PCLKB	section 20.
0008 8624h	MTU3	Timer General Register C	TGRC	16	16	2 or 3 PCLKB	section 20.
0008 8626h	MTU3	Timer General Register D	TGRD	16	16	2 or 3 PCLKB	section 20.
0008 8628h	MTU4	Timer General Register C	TGRC	16	16	2 or 3 PCLKB	section 20.
0008 862Ah	MTU4	Timer General Register D	TGRD	16	16	2 or 3 PCLKB	section 20.
0008 862Ch	MTU3	Timer Status Register	TSR	8	8	2 or 3 PCLKB	section 20.
0008 862Dh	MTU4	Timer Status Register	TSR	8	8	2 or 3 PCLKB	section 20.
0008 8630h	MTU	Timer Interrupt Skipping Set Register	TITCR	8	8	2 or 3 PCLKB	section 20.
0008 8631h	MTU	Timer Interrupt Skipping Counter	TITCNT	8	8	2 or 3 PCLKB	section 20.
0008 8632h	MTU	Timer Buffer Transfer Set Register	TBTER	8	8	2 or 3 PCLKB	section 20.
0008 8634h	MTU	Timer Dead Time Enable Register	TDER	8	8	2 or 3 PCLKB	section 20.
0008 8636h	MTU	Timer Output Level Buffer Register	TOLBR	8	8	2 or 3 PCLKB	section 20.
0008 8638h	MTU3	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	2 or 3 PCLKB	section 20.
0008 8639h	MTU4	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	2 or 3 PCLKB	section 20.
0008 8640h	MTU4	Timer A/D Converter Start Request Control Register	TADCR	16	16	2 or 3 PCLKB	section 20.
0008 8644h	MTU4	Timer A/D Converter Start Request Cycle Set Register A	TADCORA	16	16	2 or 3 PCLKB	section 20.
0008 8646h	MTU4	Timer A/D Converter Start Request Cycle Set Register B	TADCORB	16	16	2 or 3 PCLKB	section 20.
0008 8648h	MTU4	Timer A/D Converter Start Request Cycle Set Buffer Register A	TADCOBRA	16	16	2 or 3 PCLKB	section 20.
0008 864Ah	MTU4	Timer A/D Converter Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	2 or 3 PCLKB	section 20.
0008 8660h	MTU	Timer Waveform Control Register	TWCR	8	8, 16	2 or 3 PCLKB	section 20.
0008 8680h	MTU	Timer Start Register	TSTR	8	8, 16	2 or 3 PCLKB	section 20.
0008 8681h	MTU	Timer Synchronous Register	TSYR	8	8, 16	2 or 3 PCLKB	section 20.
0008 8684h	MTU	Timer Read/Write Enable Register	TRWER	8	8, 16	2 or 3 PCLKB	section 20.
0008 8690h	MTU0	Noise Filter Control Register	NFCR	8	8, 16	2 or 3 PCLKB	section 20.
0008 8691h	MTU1	Noise Filter Control Register	NFCR	8	8, 16	2 or 3 PCLKB	section 20.
0008 8692h	MTU2	Noise Filter Control Register	NFCR	8	8, 16	2 or 3 PCLKB	section 20.
0008 8693h	MTU3	Noise Filter Control Register	NFCR	8	8, 16	2 or 3 PCLKB	section 20.
0008 8694h	MTU4	Noise Filter Control Register	NFCR	8	8, 16	2 or 3 PCLKB	section 20.
0008 8695h	MTU5	Noise Filter Control Register	NFCR	8	8, 16	2 or 3 PCLKB	section 20.
0008 8700h	MTU0	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 20.
0008 8701h	MTU0	Timer Mode Register	TMDR	8	8	2 or 3 PCLKB	section 20.
0008 8702h	MTU0	Timer I/O Control Register H	TIORH	8	8	2 or 3 PCLKB	section 20.
0008 8703h	MTU0	Timer I/O Control Register L	TIORL	8	8	2 or 3 PCLKB	section 20.
0008 8704h	MTU0	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKB	section 20.
0008 8705h	MTU0	Timer Status Register	TSR	8	8	2 or 3 PCLKB	section 20.
0008 8706h	MTU0	Timer Counter	TCNT	16	16	2 or 3 PCLKB	section 20.
0008 8708h	MTU0	Timer General Register A	TGRA	16	16	2 or 3 PCLKB	section 20.
0008 870Ah	MTU0	Timer General Register B	TGRB	16	16	2 or 3 PCLKB	section 20.
0008 870Ch	MTU0	Timer General Register C	TGRC	16	16	2 or 3 PCLKB	section 20.
0008 870Eh	MTU0	Timer General Register D	TGRD	16	16	2 or 3 PCLKB	section 20.
0008 8720h	MTU0	Timer General Register E	TGRE	16	16	2 or 3 PCLKB	section 20.
0008 8722h	MTU0	Timer General Register F	TGRF	16	16	2 or 3 PCLKB	section 20.
0008 8724h	MTU0	Timer Interrupt Enable Register 2	TIER2	8	8	2 or 3 PCLKB	section 20.
0008 8726h	MTU0	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	2 or 3 PCLKB	section 20.
0008 8780h	MTU1	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 20.
0008 8781h	MTU1	Timer Mode Register	TMDR	8	8	2 or 3 PCLKB	section 20.
0008 8782h	MTU1	Timer I/O Control Register	TIOR	8	8	2 or 3 PCLKB	section 20.
0008 8784h	MTU1	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKB	section 20.

**Table 5.1 List of I/O Registers (Address Order) (11/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 8785h	MTU1	Timer Status Register	TSR	8	8	2 or 3 PCLKB	section 20.
0008 8786h	MTU1	Timer Counter	TCNT	16	16	2 or 3 PCLKB	section 20.
0008 8788h	MTU1	Timer General Register A	TGRA	16	16	2 or 3 PCLKB	section 20.
0008 878Ah	MTU1	Timer General Register B	TGRB	16	16	2 or 3 PCLKB	section 20.
0008 8790h	MTU1	Timer Input Capture Control Register	TICCR	8	8	2 or 3 PCLKB	section 20.
0008 8800h	MTU2	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 20.
0008 8801h	MTU2	Timer Mode Register	TMDR	8	8	2 or 3 PCLKB	section 20.
0008 8802h	MTU2	Timer I/O Control Register	TIOR	8	8	2 or 3 PCLKB	section 20.
0008 8804h	MTU2	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKB	section 20.
0008 8805h	MTU2	Timer Status Register	TSR	8	8	2 or 3 PCLKB	section 20.
0008 8806h	MTU2	Timer Counter	TCNT	16	16	2 or 3 PCLKB	section 20.
0008 8808h	MTU2	Timer General Register A	TGRA	16	16	2 or 3 PCLKB	section 20.
0008 880Ah	MTU2	Timer General Register B	TGRB	16	16	2 or 3 PCLKB	section 20.
0008 8880h	MTU5	Timer Counter U	TCNTU	16	16	2 or 3 PCLKB	section 20.
0008 8882h	MTU5	Timer General Register U	TGRU	16	16	2 or 3 PCLKB	section 20.
0008 8884h	MTU5	Timer Control Register U	TCRU	8	8	2 or 3 PCLKB	section 20.
0008 8886h	MTU5	Timer I/O Control Register U	TIORU	8	8	2 or 3 PCLKB	section 20.
0008 8890h	MTU5	Timer Counter V	TCNTV	16	16	2 or 3 PCLKB	section 20.
0008 8892h	MTU5	Timer General Register V	TGRV	16	16	2 or 3 PCLKB	section 20.
0008 8894h	MTU5	Timer Control Register V	TCRV	8	8	2 or 3 PCLKB	section 20.
0008 8896h	MTU5	Timer I/O Control Register V	TIORV	8	8	2 or 3 PCLKB	section 20.
0008 88A0h	MTU5	Timer Counter W	TCNTW	16	16	2 or 3 PCLKB	section 20.
0008 88A2h	MTU5	Timer General Register W	TGRW	16	16	2 or 3 PCLKB	section 20.
0008 88A4h	MTU5	Timer Control Register W	TCRW	8	8	2 or 3 PCLKB	section 20.
0008 88A6h	MTU5	Timer I/O Control Register W	TIORW	8	8	2 or 3 PCLKB	section 20.
0008 88B2h	MTU5	Timer Interrupt Enable Register	TIER	8	8	2 or 3 PCLKB	section 20.
0008 88B4h	MTU5	Timer Start Register	TSTR	8	8	2 or 3 PCLKB	section 20.
0008 88B6h	MTU5	Timer Compare Match Clear Register	TCNTCMPCLR	8	8	2 or 3 PCLKB	section 20.
0008 8900h	POE	Input Level Control/Status Register 1	ICSR1	16	8, 16	2 or 3 PCLKB	section 21.
0008 8902h	POE	Output Level Control/Status Register 1	OCSR1	16	8, 16	2 or 3 PCLKB	section 21.
0008 8908h	POE	Input Level Control/Status Register 2	ICSR2	16	8, 16	2 or 3 PCLKB	section 21.
0008 890Ah	POE	Software Port Output Enable Register	SPOER	8	8	2 or 3 PCLKB	section 21.
0008 890Bh	POE	Port Output Enable Control Register 1	POECR1	8	8	2 or 3 PCLKB	section 21.
0008 890Ch	POE	Port Output Enable Control Register 2	POECR2	8	8	2 or 3 PCLKB	section 21.
0008 890Eh	POE	Input Level Control/Status Register 3	ICSR3	16	8, 16	2 or 3 PCLKB	section 21.
0008 9000h	S12AD	A/D Control Register	ADCSR	16	16	2 or 3 PCLKB	section 35.
0008 9004h	S12AD	A/D Channel Select Register A	ADANSA	16	16	2 or 3 PCLKB	section 35.
0008 9006h	S12AD	A/D Channel Select Register A1	ADANSA1	16	16	2 or 3 PCLKB	section 35.
0008 9008h	S12AD	A/D-Converted Value Addition Mode Select Register	ADADS	16	16	2 or 3 PCLKB	section 35.
0008 900Ah	S12AD	A/D-Converted Value Addition Mode Select Register 1	ADADS1	16	16	2 or 3 PCLKB	section 35.
0008 900Ch	S12AD	A/D-Converted Value Addition Count Select Register	ADADC	8	8	2 or 3 PCLKB	section 35.
0008 900Eh	S12AD	A/D Control Extended Register	ADCER	16	16	2 or 3 PCLKB	section 35.
0008 9010h	S12AD	A/D Start Trigger Select Register	ADSTRGR	16	16	2 or 3 PCLKB	section 35.
0008 9012h	S12AD	A/D Converted Extended Input Control Register	ADEXICR	16	16	2 or 3 PCLKB	section 35.
0008 9014h	S12AD	A/D Channel Select Register B	ADANSB	16	16	2 or 3 PCLKB	section 35.
0008 9016h	S12AD	A/D Channel Select Register B1	ADANSB1	16	16	2 or 3 PCLKB	section 35.
0008 9018h	S12AD	A/D Data Duplication Register	ADDBLDR	16	16	2 or 3 PCLKB	section 35.
0008 901Ah	S12AD	A/D Temperature Sensor Data Register	ADTSDR	16	16	2 or 3 PCLKB	section 35.
0008 901Ch	S12AD	A/D Internal Reference Voltage Data Register	ADOCDR	16	16	2 or 3 PCLKB	section 35.
0008 9020h	S12AD	A/D Data Register 0	ADDR0	16	16	2 or 3 PCLKB	section 35.
0008 9022h	S12AD	A/D Data Register 1	ADDR1	16	16	2 or 3 PCLKB	section 35.
0008 9024h	S12AD	A/D Data Register 2	ADDR2	16	16	2 or 3 PCLKB	section 35.

**Table 5.1 List of I/O Registers (Address Order) (12/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 9026h	S12AD	A/D Data Register 3	ADDR3	16	16	2 or 3 PCLKB	section 35.
0008 9028h	S12AD	A/D Data Register 4	ADDR4	16	16	2 or 3 PCLKB	section 35.
0008 902Ah	S12AD	A/D Data Register 5	ADDR5	16	16	2 or 3 PCLKB	section 35.
0008 902Ch	S12AD	A/D Data Register 6	ADDR6	16	16	2 or 3 PCLKB	section 35.
0008 902Eh	S12AD	A/D Data Register 7	ADDR7	16	16	2 or 3 PCLKB	section 35.
0008 9030h	S12AD	A/D Data Register 8	ADDR8	16	16	2 or 3 PCLKB	section 35.
0008 9032h	S12AD	A/D Data Register 9	ADDR9	16	16	2 or 3 PCLKB	section 35.
0008 9034h	S12AD	A/D Data Register 10	ADDR10	16	16	2 or 3 PCLKB	section 35.
0008 9036h	S12AD	A/D Data Register 11	ADDR11	16	16	2 or 3 PCLKB	section 35.
0008 9038h	S12AD	A/D Data Register 12	ADDR12	16	16	2 or 3 PCLKB	section 35.
0008 903Ah	S12AD	A/D Data Register 13	ADDR13	16	16	2 or 3 PCLKB	section 35.
0008 903Ch	S12AD	A/D Data Register 14	ADDR14	16	16	2 or 3 PCLKB	section 35.
0008 903Eh	S12AD	A/D Data Register 15	ADDR15	16	16	2 or 3 PCLKB	section 35.
0008 904Ah	S12AD	A/D Data Register 21	ADDR21	16	16	2 or 3 PCLKB	section 35.
0008 9060h	S12AD	A/D Sampling State Register 0	ADSSTR0	8	8	2 or 3 PCLKB	section 35.
0008 9061h	S12AD	A/D Sampling State Register L	ADSSTRL	8	8	2 or 3 PCLKB	section 35.
0008 9070h	S12AD	A/D Sampling State Register T	ADSSTRT	8	8	2 or 3 PCLKB	section 35.
0008 9071h	S12AD	A/D Sampling State Register O	ADSSTRO	8	8	2 or 3 PCLKB	section 35.
0008 9073h	S12AD	A/D Sampling State Register 1	ADSSTR1	8	8	2 or 3 PCLKB	section 35.
0008 9074h	S12AD	A/D Sampling State Register 2	ADSSTR2	8	8	2 or 3 PCLKB	section 35.
0008 9075h	S12AD	A/D Sampling State Register 3	ADSSTR3	8	8	2 or 3 PCLKB	section 35.
0008 9076h	S12AD	A/D Sampling State Register 4	ADSSTR4	8	8	2 or 3 PCLKB	section 35.
0008 9077h	S12AD	A/D Sampling State Register 5	ADSSTR5	8	8	2 or 3 PCLKB	section 35.
0008 9078h	S12AD	A/D Sampling State Register 6	ADSSTR6	8	8	2 or 3 PCLKB	section 35.
0008 9079h	S12AD	A/D Sampling State Register 7	ADSSTR7	8	8	2 or 3 PCLKB	section 35.
0008 907Ch	S12AD	A/D High-Side Reference Voltage Control Register	ADHVREFCNT	8	8	2 or 3 PCLKB	section 35.
0008 9080h	S12AD	A/D Sampling State Register 21	ADSSTR21	8	8	2 or 3 PCLKB	section 35.
0008 A000h	SCI0	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 27.
0008 A001h	SCI0	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 27.
0008 A002h	SCI0	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 27.
0008 A003h	SCI0	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 27.
0008 A004h	SCI0	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 27.
0008 A005h	SCI0	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 27.
0008 A006h	SCI0	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 27.
0008 A007h	SCI0	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 27.
0008 A008h	SCI0	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 27.
0008 A009h	SCI0	I <sup>2</sup> C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 27.
0008 A00Ah	SCI0	I <sup>2</sup> C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 27.
0008 A00Bh	SCI0	I <sup>2</sup> C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 27.
0008 A00Ch	SCI0	I <sup>2</sup> C Status Register	SISR	8	8	2 or 3 PCLKB	section 27.
0008 A00Dh	SCI0	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 27.
0008 A020h	SCI1	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 27.
0008 A021h	SCI1	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 27.
0008 A022h	SCI1	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 27.
0008 A023h	SCI1	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 27.
0008 A024h	SCI1	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 27.
0008 A025h	SCI1	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 27.
0008 A026h	SCI1	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 27.
0008 A027h	SCI1	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 27.
0008 A028h	SCI1	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 27.
0008 A029h	SCI1	I <sup>2</sup> C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 27.
0008 A02Ah	SCI1	I <sup>2</sup> C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 27.

**Table 5.1 List of I/O Registers (Address Order) (13/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 A02Bh	SCI1	I <sup>2</sup> C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 27.
0008 A02Ch	SCI1	I <sup>2</sup> C Status Register	SISR	8	8	2 or 3 PCLKB	section 27.
0008 A02Dh	SCI1	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 27.
0008 A040h	SCI2	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 27.
0008 A041h	SCI2	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 27.
0008 A042h	SCI2	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 27.
0008 A043h	SCI2	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 27.
0008 A044h	SCI2	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 27.
0008 A045h	SCI2	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 27.
0008 A046h	SCI2	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 27.
0008 A047h	SCI2	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 27.
0008 A048h	SCI2	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 27.
0008 A049h	SCI2	I <sup>2</sup> C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 27.
0008 A04Ah	SCI2	I <sup>2</sup> C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 27.
0008 A04Bh	SCI2	I <sup>2</sup> C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 27.
0008 A04Ch	SCI2	I <sup>2</sup> C Status Register	SISR	8	8	2 or 3 PCLKB	section 27.
0008 A04Dh	SCI2	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 27.
0008 A0A0h	SCI5	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 27.
0008 A0A1h	SCI5	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 27.
0008 A0A2h	SCI5	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 27.
0008 A0A3h	SCI5	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 27.
0008 A0A4h	SCI5	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 27.
0008 A0A5h	SCI5	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 27.
0008 A0A6h	SCI5	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 27.
0008 A0A7h	SCI5	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 27.
0008 A0A8h	SCI5	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 27.
0008 A0A9h	SCI5	I <sup>2</sup> C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 27.
0008 A0AAh	SCI5	I <sup>2</sup> C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 27.
0008 A0ABh	SCI5	I <sup>2</sup> C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 27.
0008 A0ACh	SCI5	I <sup>2</sup> C Status Register	SISR	8	8	2 or 3 PCLKB	section 27.
0008 A0ADh	SCI5	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 27.
0008 A0C0h	SCI6	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 27.
0008 A0C1h	SCI6	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 27.
0008 A0C2h	SCI6	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 27.
0008 A0C3h	SCI6	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 27.
0008 A0C4h	SCI6	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 27.
0008 A0C5h	SCI6	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 27.
0008 A0C6h	SCI6	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 27.
0008 A0C7h	SCI6	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 27.
0008 A0C8h	SCI6	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 27.
0008 A0C9h	SCI6	I <sup>2</sup> C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 27.
0008 A0CAh	SCI6	I <sup>2</sup> C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 27.
0008 A0CBh	SCI6	I <sup>2</sup> C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 27.
0008 A0CCh	SCI6	I <sup>2</sup> C Status Register	SISR	8	8	2 or 3 PCLKB	section 27.
0008 A0CDh	SCI6	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 27.
0008 A100h	SCI8	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 27.
0008 A101h	SCI8	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 27.
0008 A102h	SCI8	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 27.
0008 A103h	SCI8	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 27.
0008 A104h	SCI8	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 27.
0008 A105h	SCI8	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 27.
0008 A106h	SCI8	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 27.



**Table 5.1 List of I/O Registers (Address Order) (14/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 A107h	SCI8	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 27.
0008 A108h	SCI8	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 27.
0008 A109h	SCI8	I <sup>2</sup> C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 27.
0008 A10Ah	SCI8	I <sup>2</sup> C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 27.
0008 A10Bh	SCI8	I <sup>2</sup> C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 27.
0008 A10Ch	SCI8	I <sup>2</sup> C Status Register	SISR	8	8	2 or 3 PCLKB	section 27.
0008 A10Dh	SCI8	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 27.
0008 A120h	SCI9	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 27.
0008 A121h	SCI9	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 27.
0008 A122h	SCI9	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 27.
0008 A123h	SCI9	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 27.
0008 A124h	SCI9	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 27.
0008 A125h	SCI9	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 27.
0008 A126h	SCI9	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 27.
0008 A127h	SCI9	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 27.
0008 A128h	SCI9	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 27.
0008 A129h	SCI9	I <sup>2</sup> C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 27.
0008 A12Ah	SCI9	I <sup>2</sup> C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 27.
0008 A12Bh	SCI9	I <sup>2</sup> C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 27.
0008 A12Ch	SCI9	I <sup>2</sup> C Status Register	SISR	8	8	2 or 3 PCLKB	section 27.
0008 A12Dh	SCI9	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 27.
0008 A500h	SSI0	Control Register	SSICR	32	32	2 or 3 PCLKB	section 35.
0008 A504h	SSI0	Status Register	SSISR	32	32	2 or 3 PCLKB	section 35.
0008 A510h	SSI0	FIFO Control Register	SSIFCR	32	32	2 or 3 PCLKB	section 35.
0008 A514h	SSI0	FIFO Status Register	SSIFSR	32	32	2 or 3 PCLKB	section 35.
0008 A518h	SSI0	Transmit FIFO Data Register	SSIFTDR	32	32	2 or 3 PCLKB	section 35.
0008 A51Ch	SSI0	Receive FIFO Data Register	SSIFRDR	32	32	2 or 3 PCLKB	section 35.
0008 A520h	SSI0	TDM Mode Register	SSITDMR	32	32	2 or 3 PCLKB	section 35.
0008 B000h	CAC	CAC Control Register 0	CACR0	8	8	2 or 3 PCLKB	section 10.
0008 B001h	CAC	CAC Control Register 1	CACR1	8	8	2 or 3 PCLKB	section 10.
0008 B002h	CAC	CAC Control Register 2	CACR2	8	8	2 or 3 PCLKB	section 10.
0008 B003h	CAC	CAC Interrupt Request Enable Register	CAICR	8	8	2 or 3 PCLKB	section 10.
0008 B004h	CAC	CAC Status Register	CASTR	8	8	2 or 3 PCLKB	section 10.
0008 B006h	CAC	CAC Upper-Limit Value Setting Register	CAULVR	16	16	2 or 3 PCLKB	section 10.
0008 B008h	CAC	CAC Lower-Limit Value Setting Register	CALLVR	16	16	2 or 3 PCLKB	section 10.
0008 B00Ah	CAC	CAC Counter Buffer Register	CACNTBR	16	16	2 or 3 PCLKB	section 10.
0008 B080h	DOC	DOC Control Register	DOCR	8	8	2 or 3 PCLKB	section 39.
0008 B082h	DOC	DOC Data Input Register	DODIR	16	16	2 or 3 PCLKB	section 39.
0008 B084h	DOC	DOC Data Setting Register	DODSR	16	16	2 or 3 PCLKB	section 39.
0008 B100h	ELC	Event Link Control Register	ELCR	8	8	2 or 3 PCLKB	section 17.
0008 B102h	ELC	Event Link Setting Register 1	ELSR1	8	8	2 or 3 PCLKB	section 17.
0008 B103h	ELC	Event Link Setting Register 2	ELSR2	8	8	2 or 3 PCLKB	section 17.
0008 B104h	ELC	Event Link Setting Register 3	ELSR3	8	8	2 or 3 PCLKB	section 17.
0008 B105h	ELC	Event Link Setting Register 4	ELSR4	8	8	2 or 3 PCLKB	section 17.
0008 B108h	ELC	Event Link Setting Register 7	ELSR7	8	8	2 or 3 PCLKB	section 17.
0008 B10Bh	ELC	Event Link Setting Register 10	ELSR10	8	8	2 or 3 PCLKB	section 17.
0008 B10Dh	ELC	Event Link Setting Register 12	ELSR12	8	8	2 or 3 PCLKB	section 17.
0008 B10Fh	ELC	Event Link Setting Register 14	ELSR14	8	8	2 or 3 PCLKB	section 17.
0008 B110h	ELC	Event Link Setting Register 15	ELSR15	8	8	2 or 3 PCLKB	section 17.
0008 B112h	ELC	Event Link Setting Register 17	ELSR17	8	8	2 or 3 PCLKB	section 17.
0008 B113h	ELC	Event Link Setting Register 18	ELSR18	8	8	2 or 3 PCLKB	section 17.
0008 B115h	ELC	Event Link Setting Register 20	ELSR20	8	8	2 or 3 PCLKB	section 17.

**Table 5.1 List of I/O Registers (Address Order) (15/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 B117h	ELC	Event Link Setting Register 22	ELSR22	8	8	2 or 3 PCLKB	section 17.
0008 B119h	ELC	Event Link Setting Register 24	ELSR24	8	8	2 or 3 PCLKB	section 17.
0008 B11Ah	ELC	Event Link Setting Register 25	ELSR25	8	8	2 or 3 PCLKB	section 17.
0008 B11Fh	ELC	Event Link Option Setting Register A	ELOPA	8	8	2 or 3 PCLKB	section 17.
0008 B120h	ELC	Event Link Option Setting Register B	ELOPB	8	8	2 or 3 PCLKB	section 17.
0008 B121h	ELC	Event Link Option Setting Register C	ELOPC	8	8	2 or 3 PCLKB	section 17.
0008 B122h	ELC	Event Link Option Setting Register D	ELOPD	8	8	2 or 3 PCLKB	section 17.
0008 B123h	ELC	Port Group Setting Register 1	PGR1	8	8	2 or 3 PCLKB	section 17.
0008 B125h	ELC	Port Group Control Register 1	PGC1	8	8	2 or 3 PCLKB	section 17.
0008 B127h	ELC	Port Buffer Register 1	PDBF1	8	8	2 or 3 PCLKB	section 17.
0008 B129h	ELC	Event Link Port Setting Register 0	PEL0	8	8	2 or 3 PCLKB	section 17.
0008 B12Ah	ELC	Event Link Port Setting Register 1	PEL1	8	8	2 or 3 PCLKB	section 17.
0008 B12Dh	ELC	Event Link Software Event Generation Register	ELSEGR	8	8	2 or 3 PCLKB	section 17.
0008 B300h	SCI12	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 27.
0008 B301h	SCI12	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 27.
0008 B302h	SCI12	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 27.
0008 B303h	SCI12	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 27.
0008 B304h	SCI12	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 27.
0008 B305h	SCI12	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 27.
0008 B306h	SCI12	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 27.
0008 B307h	SCI12	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 27.
0008 B308h	SCI12	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 27.
0008 B309h	SCI12	I <sup>2</sup> C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 27.
0008 B30Ah	SCI12	I <sup>2</sup> C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 27.
0008 B30Bh	SCI12	I <sup>2</sup> C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 27.
0008 B30Ch	SCI12	I <sup>2</sup> C Status Register	SISR	8	8	2 or 3 PCLKB	section 27.
0008 B30Dh	SCI12	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 27.
0008 B320h	SCI12	Extended Serial Mode Enable Register	ESMER	8	8	2 or 3 PCLKB	section 27.
0008 B321h	SCI12	Control Register 0	CR0	8	8	2 or 3 PCLKB	section 27.
0008 B322h	SCI12	Control Register 1	CR1	8	8	2 or 3 PCLKB	section 27.
0008 B323h	SCI12	Control Register 2	CR2	8	8	2 or 3 PCLKB	section 27.
0008 B324h	SCI12	Control Register 3	CR3	8	8	2 or 3 PCLKB	section 27.
0008 B325h	SCI12	Port Control Register	PCR	8	8	2 or 3 PCLKB	section 27.
0008 B326h	SCI12	Interrupt Control Register	ICR	8	8	2 or 3 PCLKB	section 27.
0008 B327h	SCI12	Status Register	STR	8	8	2 or 3 PCLKB	section 27.
0008 B328h	SCI12	Status Clear Register	STCR	8	8	2 or 3 PCLKB	section 27.
0008 B329h	SCI12	Control Field 0 Data Register	CF0DR	8	8	2 or 3 PCLKB	section 27.
0008 B32Ah	SCI12	Control Field 0 Compare Enable Register	CF0CR	8	8	2 or 3 PCLKB	section 27.
0008 B32Bh	SCI12	Control Field 0 Receive Data Register	CF0RR	8	8	2 or 3 PCLKB	section 27.
0008 B32Ch	SCI12	Primary Control Field 1 Data Register	PCF1DR	8	8	2 or 3 PCLKB	section 27.
0008 B32Dh	SCI12	Secondary Control Field 1 Data Register	SCF1DR	8	8	2 or 3 PCLKB	section 27.
0008 B32Eh	SCI12	Control Field 1 Compare Enable Register	CF1CR	8	8	2 or 3 PCLKB	section 27.
0008 B32Fh	SCI12	Control Field 1 Receive Data Register	CF1RR	8	8	2 or 3 PCLKB	section 27.
0008 B330h	SCI12	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 27.
0008 B331h	SCI12	Timer Mode Register	TMR	8	8	2 or 3 PCLKB	section 27.
0008 B332h	SCI12	Timer Prescaler Register	TPRE	8	8	2 or 3 PCLKB	section 27.
0008 B333h	SCI12	Timer Count Register	TCNT	8	8	2 or 3 PCLKB	section 27.
0008 C000h	PORT0	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C001h	PORT1	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C002h	PORT2	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C003h	PORT3	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C004h	PORT4	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.

**Table 5.1 List of I/O Registers (Address Order) (16/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 C005h	PORT5	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C009h	PORT9	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C00Ah	PORTA	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C00Bh	PORTB	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C00Ch	PORTC	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C00Dh	PORTD	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C00Eh	PORTE	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C00Fh	PORTF	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C012h	PORTJ	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 18.
0008 C020h	PORT0	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C021h	PORT1	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C022h	PORT2	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C023h	PORT3	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C024h	PORT4	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C025h	PORT5	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C029h	PORT9	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C02Ah	PORTA	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C02Bh	PORTB	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C02Ch	PORTC	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C02Dh	PORTD	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C02Eh	PORTE	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C02Fh	PORTF	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C032h	PORTJ	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 18.
0008 C040h	PORT0	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C041h	PORT1	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C042h	PORT2	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C043h	PORT3	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C044h	PORT4	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C045h	PORT5	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C049h	PORT9	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C04Ah	PORTA	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C04Bh	PORTB	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C04Ch	PORTC	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C04Dh	PORTD	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.

**Table 5.1 List of I/O Registers (Address Order) (17/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 C04Eh	PORTE	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C04Fh	PORTF	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C051h	PORTH	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C052h	PORTJ	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 18.
0008 C060h	PORT0	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C061h	PORT1	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C062h	PORT2	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C063h	PORT3	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C064h	PORT4	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C065h	PORT5	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C069h	PORT9	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C06Ah	PORTA	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C06Bh	PORTB	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C06Ch	PORTC	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C06Dh	PORTD	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C06Eh	PORTE	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C06Fh	PORTF	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C071h	PORTH	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C072h	PORTJ	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 18.
0008 C080h	PORT0	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C081h	PORT0	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C082h	PORT1	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C083h	PORT1	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C084h	PORT2	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C085h	PORT2	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C086h	PORT3	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C08Ah	PORT5	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C08Bh	PORT5	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C094h	PORTA	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C095h	PORTA	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C096h	PORTB	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C097h	PORTB	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C098h	PORTC	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C099h	PORTC	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C09Ch	PORTE	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C09Dh	PORTE	Open Drain Control Register 1	ODR1	8	8, 16	2 or 3 PCLKB	section 18.
0008 C0A4h	PORTJ	Open Drain Control Register 0	ODR0	8	8, 16	2 or 3 PCLKB	section 18.
0008 C0C0h	PORT0	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0C1h	PORT1	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0C2h	PORT2	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0C3h	PORT3	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0C5h	PORT5	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0CAh	PORTA	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0CBh	PORTB	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0CCh	PORTC	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0CDh	PORTD	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.

**Table 5.1 List of I/O Registers (Address Order) (18/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 C0CEh	PORTE	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0CFh	PORTF	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C0D2h	PORTJ	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 18.
0008 C11Fh	MPC	Write-Protect Register	PWPR	8	8	2 or 3 PCLKB	section 19.
0008 C121h	PORT	Port Switching Register A	PSRA	8	8	2 or 3 PCLKB	section 18.
0008 C142h	MPC	P02 Pin Function Control Register	P02PFS	8	8	2 or 3 PCLKB	section 18.
0008 C144h	MPC	P04 Pin Function Control Register	P04PFS	8	8	2 or 3 PCLKB	section 19.
0008 C147h	MPC	P07 Pin Function Control Register	P07PFS	8	8	2 or 3 PCLKB	section 19.
0008 C148h	MPC	P10 Pin Function Control Register	P10PFS	8	8	2 or 3 PCLKB	section 19.
0008 C149h	MPC	P11 Pin Function Control Register	P11PFS	8	8	2 or 3 PCLKB	section 19.
0008 C14Ah	MPC	P12 Pin Function Control Register	P12PFS	8	8	2 or 3 PCLKB	section 19.
0008 C14Bh	MPC	P13 Pin Function Control Register	P13PFS	8	8	2 or 3 PCLKB	section 19.
0008 C14Ch	MPC	P14 Pin Function Control Register	P14PFS	8	8	2 or 3 PCLKB	section 19.
0008 C14Dh	MPC	P15 Pin Function Control Register	P15PFS	8	8	2 or 3 PCLKB	section 19.
0008 C14Eh	MPC	P16 Pin Function Control Register	P16PFS	8	8	2 or 3 PCLKB	section 19.
0008 C14Fh	MPC	P17 Pin Function Control Register	P17PFS	8	8	2 or 3 PCLKB	section 19.
0008 C150h	MPC	P20 Pin Function Control Register	P20PFS	8	8	2 or 3 PCLKB	section 19.
0008 C151h	MPC	P21 Pin Function Control Register	P21PFS	8	8	2 or 3 PCLKB	section 19.
0008 C152h	MPC	P22 Pin Function Control Register	P22PFS	8	8	2 or 3 PCLKB	section 19.
0008 C153h	MPC	P23 Pin Function Control Register	P23PFS	8	8	2 or 3 PCLKB	section 19.
0008 C154h	MPC	P24 Pin Function Control Register	P24PFS	8	8	2 or 3 PCLKB	section 19.
0008 C155h	MPC	P25 Pin Function Control Register	P25PFS	8	8	2 or 3 PCLKB	section 19.
0008 C156h	MPC	P26 Pin Function Control Register	P26PFS	8	8	2 or 3 PCLKB	section 19.
0008 C157h	MPC	P27 Pin Function Control Register	P27PFS	8	8	2 or 3 PCLKB	section 19.
0008 C158h	MPC	P30 Pin Function Control Register	P30PFS	8	8	2 or 3 PCLKB	section 19.
0008 C159h	MPC	P31 Pin Function Control Register	P31PFS	8	8	2 or 3 PCLKB	section 19.
0008 C15Ah	MPC	P32 Pin Function Control Register	P32PFS	8	8	2 or 3 PCLKB	section 19.
0008 C160h	MPC	P40 Pin Function Control Register	P40PFS	8	8	2 or 3 PCLKB	section 19.
0008 C161h	MPC	P41 Pin Function Control Register	P41PFS	8	8	2 or 3 PCLKB	section 19.
0008 C162h	MPC	P42 Pin Function Control Register	P42PFS	8	8	2 or 3 PCLKB	section 19.
0008 C163h	MPC	P43 Pin Function Control Register	P43PFS	8	8	2 or 3 PCLKB	section 19.
0008 C164h	MPC	P44 Pin Function Control Register	P44PFS	8	8	2 or 3 PCLKB	section 19.
0008 C166h	MPC	P46 Pin Function Control Register	P46PFS	8	8	2 or 3 PCLKB	section 19.
0008 C168h	MPC	P50 Pin Function Control Register	P50PFS	8	8	2 or 3 PCLKB	section 19.
0008 C169h	MPC	P51 Pin Function Control Register	P51PFS	8	8	2 or 3 PCLKB	section 19.
0008 C16Ah	MPC	P52 Pin Function Control Register	P52PFS	8	8	2 or 3 PCLKB	section 19.
0008 C16Bh	MPC	P53 Pin Function Control Register	P53PFS	8	8	2 or 3 PCLKB	section 19.
0008 C16Ch	MPC	P54 Pin Function Control Register	P54PFS	8	8	2 or 3 PCLKB	section 19.
0008 C16Dh	MPC	P55 Pin Function Control Register	P55PFS	8	8	2 or 3 PCLKB	section 19.
0008 C16Eh	MPC	P56 Pin Function Control Register	P56PFS	8	8	2 or 3 PCLKB	section 19.
0008 C188h	MPC	P90 Pin Function Control Register	P90PFS	8	8	2 or 3 PCLKB	section 19.
0008 C189h	MPC	P91 Pin Function Control Register	P91PFS	8	8	2 or 3 PCLKB	section 19.
0008 C18Ah	MPC	P92 Pin Function Control Register	P92PFS	8	8	2 or 3 PCLKB	section 19.
0008 C190h	MPC	PA0 Pin Function Control Register	PA0PFS	8	8	2 or 3 PCLKB	section 19.
0008 C191h	MPC	PA1 Pin Function Control Register	PA1PFS	8	8	2 or 3 PCLKB	section 19.
0008 C192h	MPC	PA2 Pin Function Control Register	PA2PFS	8	8	2 or 3 PCLKB	section 19.
0008 C193h	MPC	PA3 Pin Function Control Register	PA3PFS	8	8	2 or 3 PCLKB	section 19.
0008 C194h	MPC	PA4 Pin Function Control Register	PA4PFS	8	8	2 or 3 PCLKB	section 19.
0008 C195h	MPC	PA5 Pin Function Control Register	PA5PFS	8	8	2 or 3 PCLKB	section 19.
0008 C196h	MPC	PA6 Pin Function Control Register	PA6PFS	8	8	2 or 3 PCLKB	section 19.
0008 C197h	MPC	PA7 Pin Function Control Register	PA7PFS	8	8	2 or 3 PCLKB	section 19.
0008 C198h	MPC	PB0 Pin Function Control Register	PB0PFS	8	8	2 or 3 PCLKB	section 19.

**Table 5.1 List of I/O Registers (Address Order) (19/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 C199h	MPC	PB1 Pin Function Control Register	PB1PFS	8	8	2 or 3 PCLKB	section 19.
0008 C19Ah	MPC	PB2 Pin Function Control Register	PB2PFS	8	8	2 or 3 PCLKB	section 19.
0008 C19Bh	MPC	PB3 Pin Function Control Register	PB3PFS	8	8	2 or 3 PCLKB	section 19.
0008 C19Ch	MPC	PB4 Pin Function Control Register	PB4PFS	8	8	2 or 3 PCLKB	section 19.
0008 C19Dh	MPC	PB5 Pin Function Control Register	PB5PFS	8	8	2 or 3 PCLKB	section 19.
0008 C19Eh	MPC	PB6 Pin Function Control Register	PB6PFS	8	8	2 or 3 PCLKB	section 19.
0008 C19Fh	MPC	PB7 Pin Function Control Register	PB7PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1A0h	MPC	PC0 Pin Function Control Register	PC0PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1A1h	MPC	PC1 Pin Function Control Register	PC1PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1A2h	MPC	PC2 Pin Function Control Register	PC2PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1A3h	MPC	PC3 Pin Function Control Register	PC3PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1A4h	MPC	PC4 Pin Function Control Register	PC4PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1A5h	MPC	PC5 Pin Function Control Register	PC5PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1A6h	MPC	PC6 Pin Function Control Register	PC6PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1A7h	MPC	PC7 Pin Function Control Register	PC7PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1A8h	MPC	PD0 Pin Function Control Register	PD0PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1A9h	MPC	PD1 Pin Function Control Register	PD1PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1AAh	MPC	PD2 Pin Function Control Register	PD2PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1ABh	MPC	PD3 Pin Function Control Register	PD3PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1ACh	MPC	PD4 Pin Function Control Register	PD4PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1B0h	MPC	PE0 Pin Function Control Register	PE0PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1B1h	MPC	PE1 Pin Function Control Register	PE1PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1B2h	MPC	PE2 Pin Function Control Register	PE2PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1B3h	MPC	PE3 Pin Function Control Register	PE3PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1B4h	MPC	PE4 Pin Function Control Register	PE4PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1B5h	MPC	PE5 Pin Function Control Register	PE5PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1B6h	MPC	PE6 Pin Function Control Register	PE6PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1B7h	MPC	PE7 Pin Function Control Register	PE7PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1BEh	MPC	PF6 Pin Function Control Register	PF6PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1BFh	MPC	PF7 Pin Function Control Register	PF7PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1D0h	MPC	PJ0 Pin Function Control Register	PJ0PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1D2h	MPC	PJ2 Pin Function Control Register	PJ2PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1D3h	MPC	PJ3 Pin Function Control Register	PJ3PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1D6h	MPC	PJ6 Pin Function Control Register	PJ6PFS	8	8	2 or 3 PCLKB	section 19.
0008 C1D7h	MPC	PJ7 Pin Function Control Register	PJ7PFS	8	8	2 or 3 PCLKB	section 19.
0008 C290h	SYSTEM	Reset Status Register 0	RSTSR0	8	8	4 or 5 PCLKB	section 6.
0008 C291h	SYSTEM	Reset Status Register 1	RSTSR1	8	8	4 or 5 PCLKB	section 6.
0008 C293h	SYSTEM	Main Clock Oscillator Forced Oscillation Control Register	MOFCR	8	8	4 or 5 PCLKB	section 9.
0008 C297h	SYSTEM	Voltage Monitoring Circuit Control Register	LVCMPCR	8	8	4 or 5 PCLKB	section 8.
0008 C298h	SYSTEM	Voltage Detection Level Select Register	LVDLVL	8	8	4 or 5 PCLKB	section 8.
0008 C29Ah	SYSTEM	Voltage Monitoring 1 Circuit Control Register 0	LVD1CR0	8	8	4 or 5 PCLKB	section 8.
0008 C29Bh	SYSTEM	Voltage Monitoring 2 Circuit Control Register 0	LVD2CR0	8	8	4 or 5 PCLKB	section 8.
0008 C400h	RTC	64-Hz Counter	R64CNT	8	8	2 or 3 PCLKB	section 24.
0008 C402h	RTC	Second Counter	RSECCNT	8	8	2 or 3 PCLKB	section 24.
0008 C402h	RTC	Binary Counter 0	BCNT0	8	8	2 or 3 PCLKB	section 24.
0008 C404h	RTC	Minute Counter	RMINCNT	8	8	2 or 3 PCLKB	section 24.
0008 C404h	RTC	Binary Counter 1	BCNT1	8	8	2 or 3 PCLKB	section 24.
0008 C406h	RTC	Hour Counter	RHRCNT	8	8	2 or 3 PCLKB	section 24.
0008 C406h	RTC	Binary Counter 2	BCNT2	8	8	2 or 3 PCLKB	section 24.
0008 C408h	RTC	Day-Of-Week Counter	RWKCNT	8	8	2 or 3 PCLKB	section 24.
0008 C408h	RTC	Binary Counter 3	BCNT3	8	8	2 or 3 PCLKB	section 24.
0008 C40Ah	RTC	Date Counter	RDAYCNT	8	8	2 or 3 PCLKB	section 24.

**Table 5.1 List of I/O Registers (Address Order) (20/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
0008 C40Ch	RTC	Month Counter	RMONCNT	8	8	2 or 3 PCLKB	section 24.
0008 C40Eh	RTC	Year Counter	RYRCNT	16	16	2 or 3 PCLKB	section 24.
0008 C410h	RTC	Second Alarm Register	RSECAR	8	8	2 or 3 PCLKB	section 24.
0008 C410h	RTC	Binary Counter 0 Alarm Register	BCNT0AR	8	8	2 or 3 PCLKB	section 24.
0008 C412h	RTC	Minute Alarm Register	RMINAR	8	8	2 or 3 PCLKB	section 24.
0008 C412h	RTC	Binary Counter 1 Alarm Register	BCNT1AR	8	8	2 or 3 PCLKB	section 24.
0008 C414h	RTC	Hour Alarm Register	RHRAR	8	8	2 or 3 PCLKB	section 24.
0008 C414h	RTC	Binary Counter 2 Alarm Register	BCNT2AR	8	8	2 or 3 PCLKB	section 24.
0008 C416h	RTC	Day-of-Week Alarm Register	RWKAR	8	8	2 or 3 PCLKB	section 24.
0008 C416h	RTC	Binary Counter 3 Alarm Register	BCNT3AR	8	8	2 or 3 PCLKB	section 24.
0008 C418h	RTC	Date Alarm Register	RDAYAR	8	8	2 or 3 PCLKB	section 24.
0008 C418h	RTC	Binary Counter 0 Alarm Enable Register	BCNT0AER	8	8	2 or 3 PCLKB	section 24.
0008 C41Ah	RTC	Month Alarm Register	RMONAR	8	8	2 or 3 PCLKB	section 24.
0008 C41Ah	RTC	Binary Counter 1 Alarm Enable Register	BCNT1AER	8	8	2 or 3 PCLKB	section 24.
0008 C41Ch	RTC	Year Alarm Register	RYRAR	16	16	2 or 3 PCLKB	section 24.
0008 C41Ch	RTC	Binary Counter 2 Alarm Enable Register	BCNT2AER	16	16	2 or 3 PCLKB	section 24.
0008 C41Eh	RTC	Year Alarm Enable Register	RYRAREN	8	8	2 or 3 PCLKB	section 24.
0008 C41Eh	RTC	Binary Counter 3 Alarm Enable Register	BCNT3AER	8	8	2 or 3 PCLKB	section 24.
0008 C422h	RTC	RTC Control Register 1	RCR1	8	8	2 or 3 PCLKB	section 24.
0008 C424h	RTC	RTC Control Register 2	RCR2	8	8	2 or 3 PCLKB	section 24.
0008 C426h	RTC	RTC Control Register 3	RCR3	8	8	2 or 3 PCLKB	section 24.
0008 C42Eh	RTC	Time Error Adjustment Register	RADJ	8	8	2 or 3 PCLKB	section 24.
0008 C580h	CMPB	Comparator B Control Register 1	CPBCNT1	8	8	2 or 3 PCLKB	section 38.
0008 C581h	CMPB	Comparator B Control Register 2	CPBCNT2	8	8	2 or 3 PCLKB	section 38.
0008 C582h	CMPB	Comparator B Flag Register	CPBFLG	8	8	2 or 3 PCLKB	section 38.
0008 C583h	CMPB	Comparator B Interrupt Control Register	CPBINT	8	8	2 or 3 PCLKB	section 38.
0008 C584h	CMPB	Comparator B Filter Select Register	CPBPF	8	8	2 or 3 PCLKB	section 38.
0008 C585h	CMPB	Comparator B Mode Select Register	CPBMD	8	8	2 or 3 PCLKB	section 38.
0008 C586h	CMPB	Comparator B Reference Input Voltage Select Register	CPBREF	8	8	2 or 3 PCLKB	section 38.
0008 C587h	CMPB	Comparator B Output Control Register	CPBOCR	8	8	2 or 3 PCLKB	section 38.
000A 0000h	USB0	System Configuration Control Register	SYSCFG	16	16	3 or 4 PCLKB	section 26.
000A 0004h	USB0	System Configuration Status Register 0	SYSSTS0	16	16	9 PCLKB or more	section 26.
000A 0008h	USB0	Device State Control Register 0	DVSTCTR0	16	16	9 PCLKB or more	section 26.
000A 0014h	USB0	CFIFO Port Register	CFIFO	16	16	3 or 4 PCLKB	section 26.
000A 0018h	USB0	D0FIFO Port Register	D0FIFO	16	16	3 or 4 PCLKB	section 26.
000A 001Ch	USB0	D1FIFO Port Register	D1FIFO	16	16	3 or 4 PCLKB	section 26.
000A 0020h	USB0	CFIFO Port Select Register	CFIFOSEL	16	16	3 or 4 PCLKB	section 26.
000A 0028h	USB0	D0FIFO Port Select Register	D0FIFOSEL	16	16	3 or 4 PCLKB	section 26.
000A 002Ch	USB0	D1FIFO Port Select Register	D1FIFOSEL	16	16	3 or 4 PCLKB	section 26.
000A 0022h	USB0	CFIFO Port Control Register	CFIFOCTR	16	16	3 or 4 PCLKB	section 26.
000A 002Ah	USB0	D0FIFO Port Control Register	D0FIFOCTR	16	16	3 or 4 PCLKB	section 26.
000A 002Eh	USB0	D1FIFO Port Control Register	D1FIFOCTR	16	16	3 or 4 PCLKB	section 26.
000A 0030h	USB0	Interrupt Enable Register 0	INTENB0	16	16	9 PCLKB or more	section 26.
000A 0032h	USB0	Interrupt Enable Register 1	INTENB1	16	16	9 PCLKB or more	section 26.
000A 0036h	USB0	BRDY Interrupt Enable Register	BRDYENB	16	16	9 PCLKB or more	section 26.
000A 0038h	USB0	NRDY Interrupt Enable Register	NRDYENB	16	16	9 PCLKB or more	section 26.
000A 003Ah	USB0	BEMP Interrupt Enable Register	BEMPENB	16	16	9 PCLKB or more	section 26.
000A 003Ch	USB0	SOF Output Configuration Register	SOFCFG	16	16	9 PCLKB or more	section 26.
000A 0040h	USB0	Interrupt Status Register 0	INTSTS0	16	16	9 PCLKB or more	section 26.
000A 0042h	USB0	Interrupt Status Register 1	INTSTS1	16	16	9 PCLKB or more	section 26.
000A 0046h	USB0	BRDY Interrupt Status Register	BRDYSTS	16	16	9 PCLKB or more	section 26.
000A 0048h	USB0	NRDY Interrupt Status Register	NRDYSTS	16	16	9 PCLKB or more	section 26.

**Table 5.1 List of I/O Registers (Address Order) (21/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
000A 004Ah	USB0	BEMP Interrupt Status Register	BEMPSTS	16	16	9 PCLKB or more	section 26.
000A 004Ch	USB0	Frame Number Register	FRMNUM	16	16	9 PCLKB or more	section 26.
000A 0054h	USB0	USB Request Type Register	USBREQ	16	16	9 PCLKB or more	section 26.
000A 0056h	USB0	USB Request Value Register	USBVAL	16	16	9 PCLKB or more	section 26.
000A 0058h	USB0	USB Request Index Register	USBINDX	16	16	9 PCLKB or more	section 26.
000A 005Ah	USB0	USB Request Length Register	USBLENG	16	16	9 PCLKB or more	section 26.
000A 005Ch	USB0	DCP Configuration Register	DCPCFG	16	16	9 PCLKB or more	section 26.
000A 005Eh	USB0	DCP Maximum Packet Size Register	DCPMAXP	16	16	9 PCLKB or more	section 26.
000A 0060h	USB0	DCP Control Register	DCPCTR	16	16	9 PCLKB or more	section 26.
000A 0064h	USB0	Pipe Window Select Register	PIPESEL	16	16	9 PCLKB or more	section 26.
000A 0068h	USB0	Pipe Configuration Register	PIPECFG	16	16	9 PCLKB or more	section 26.
000A 006Ch	USB0	Pipe Maximum Packet Size Register	PIPEMAXP	16	16	9 PCLKB or more	section 26.
000A 006Eh	USB0	Pipe Cycle Control Register	PIPEPERI	16	16	9 PCLKB or more	section 26.
000A 0070h	USB0	PIPE1 Control Register	PIPE1CTR	16	16	9 PCLKB or more	section 26.
000A 0072h	USB0	PIPE2 Control Register	PIPE2CTR	16	16	9 PCLKB or more	section 26.
000A 0074h	USB0	PIPE3 Control Register	PIPE3CTR	16	16	9 PCLKB or more	section 26.
000A 0076h	USB0	PIPE4 Control Register	PIPE4CTR	16	16	9 PCLKB or more	section 26.
000A 0078h	USB0	PIPE5 Control Register	PIPE5CTR	16	16	9 PCLKB or more	section 26.
000A 007Ah	USB0	PIPE6 Control Register	PIPE6CTR	16	16	9 PCLKB or more	section 26.
000A 007Ch	USB0	PIPE7 Control Register	PIPE7CTR	16	16	9 PCLKB or more	section 26.
000A 007Eh	USB0	PIPE8 Control Register	PIPE8CTR	16	16	9 PCLKB or more	section 26.
000A 0080h	USB0	PIPE9 Control Register	PIPE9CTR	16	16	9 PCLKB or more	section 26.
000A 0090h	USB0	PIPE1 Transaction Counter Enable Register	PIPE1TRE	16	16	9 PCLKB or more	section 26.
000A 0092h	USB0	PIPE1 Transaction Counter Register	PIPE1TRN	16	16	9 PCLKB or more	section 26.
000A 0094h	USB0	PIPE2 Transaction Counter Enable Register	PIPE2TRE	16	16	9 PCLKB or more	section 26.
000A 0096h	USB0	PIPE2 Transaction Counter Register	PIPE2TRN	16	16	9 PCLKB or more	section 26.
000A 0098h	USB0	PIPE3 Transaction Counter Enable Register	PIPE3TRE	16	16	9 PCLKB or more	section 26.
000A 009Ah	USB0	PIPE3 Transaction Counter Register	PIPE3TRN	16	16	9 PCLKB or more	section 26.
000A 009Ch	USB0	PIPE4 Transaction Counter Enable Register	PIPE4TRE	16	16	9 PCLKB or more	section 26.
000A 009Eh	USB0	PIPE4 Transaction Counter Register	PIPE4TRN	16	16	9 PCLKB or more	section 26.
000A 00A0h	USB0	PIPE5 Transaction Counter Enable Register	PIPE5TRE	16	16	9 PCLKB or more	section 26.
000A 00A2h	USB0	PIPE5 Transaction Counter Register	PIPE5TRN	16	16	9 PCLKB or more	section 26.
000A 00B0h	USB0	BC Control Register 0	USBBCCTRL0	16	16	9 PCLKB or more	section 26.
000A 00CCh	USB0	USB Module Control Register	USBMC	16	16	9 PCLKB or more	section 26.
000A 00D0h	USB0	Device Address 0 Configuration Register	DEVADD0	16	16	9 PCLKB or more	section 26.
000A 00D2h	USB0	Device Address 1 Configuration Register	DEVADD1	16	16	9 PCLKB or more	section 26.
000A 00D4h	USB0	Device Address 2 Configuration Register	DEVADD2	16	16	9 PCLKB or more	section 26.
000A 00D6h	USB0	Device Address 3 Configuration Register	DEVADD3	16	16	9 PCLKB or more	section 26.
000A 00D8h	USB0	Device Address 4 Configuration Register	DEVADD4	16	16	9 PCLKB or more	section 26.
000A 00DAh	USB0	Device Address 5 Configuration Register	DEVADD5	16	16	9 PCLKB or more	section 26.
000A 0800h	LCDC	LCD mode register 0	LCDM0	8	8	1 or 2 PCLKB	section 33.
000A 0801h	LCDC	LCD mode register 1	LCDM1	8	8	1 or 2 PCLKB	section 33.
000A 0802h	LCDC	LCD Clock Control Register 0	LCDC0	8	8	1 or 2 PCLKB	section 33.
000A 0803h	LCDC	LCD Boost Level Control Register	VLCD	8	8	1 or 2 PCLKB	section 33.
000A 0840h	LCDC	LCD Display Data Register 00	SEG00	8	8	1 or 2 PCLKB	section 33.
000A 0841h	LCDC	LCD Display Data Register 01	SEG01	8	8	1 or 2 PCLKB	section 33.
000A 0842h	LCDC	LCD Display Data Register 02	SEG02	8	8	1 or 2 PCLKB	section 33.
000A 0843h	LCDC	LCD Display Data Register 03	SEG03	8	8	1 or 2 PCLKB	section 33.
000A 0844h	LCDC	LCD Display Data Register 04	SEG04	8	8	1 or 2 PCLKB	section 33.
000A 0845h	LCDC	LCD Display Data Register 05	SEG05	8	8	1 or 2 PCLKB	section 33.
000A 0846h	LCDC	LCD Display Data Register 06	SEG06	8	8	1 or 2 PCLKB	section 33.
000A 0847h	LCDC	LCD Display Data Register 07	SEG07	8	8	1 or 2 PCLKB	section 33.



**Table 5.1 List of I/O Registers (Address Order) (22/23)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
000A 0848h	LCDC	LCD Display Data Register 08	SEG08	8	8	1 or 2 PCLKB	section 33.
000A 0849h	LCDC	LCD Display Data Register 09	SEG09	8	8	1 or 2 PCLKB	section 33.
000A 084Ah	LCDC	LCD Display Data Register 10	SEG10	8	8	1 or 2 PCLKB	section 33.
000A 084Bh	LCDC	LCD Display Data Register 11	SEG11	8	8	1 or 2 PCLKB	section 33.
000A 084Ch	LCDC	LCD Display Data Register 12	SEG12	8	8	1 or 2 PCLKB	section 33.
000A 084Dh	LCDC	LCD Display Data Register 13	SEG13	8	8	1 or 2 PCLKB	section 33.
000A 084Eh	LCDC	LCD Display Data Register 14	SEG14	8	8	1 or 2 PCLKB	section 33.
000A 084Fh	LCDC	LCD Display Data Register 15	SEG15	8	8	1 or 2 PCLKB	section 33.
000A 0850h	LCDC	LCD Display Data Register 16	SEG16	8	8	1 or 2 PCLKB	section 33.
000A 0851h	LCDC	LCD Display Data Register 17	SEG17	8	8	1 or 2 PCLKB	section 33.
000A 0852h	LCDC	LCD Display Data Register 18	SEG18	8	8	1 or 2 PCLKB	section 33.
000A 0853h	LCDC	LCD Display Data Register 19	SEG19	8	8	1 or 2 PCLKB	section 33.
000A 0854h	LCDC	LCD Display Data Register 20	SEG20	8	8	1 or 2 PCLKB	section 33.
000A 0855h	LCDC	LCD Display Data Register 21	SEG21	8	8	1 or 2 PCLKB	section 33.
000A 0856h	LCDC	LCD Display Data Register 22	SEG22	8	8	1 or 2 PCLKB	section 33.
000A 0857h	LCDC	LCD Display Data Register 23	SEG23	8	8	1 or 2 PCLKB	section 33.
000A 0858h	LCDC	LCD Display Data Register 24	SEG24	8	8	1 or 2 PCLKB	section 33.
000A 0859h	LCDC	LCD Display Data Register 25	SEG25	8	8	1 or 2 PCLKB	section 33.
000A 085Ah	LCDC	LCD Display Data Register 26	SEG26	8	8	1 or 2 PCLKB	section 33.
000A 085Bh	LCDC	LCD Display Data Register 27	SEG27	8	8	1 or 2 PCLKB	section 33.
000A 085Ch	LCDC	LCD Display Data Register 28	SEG28	8	8	1 or 2 PCLKB	section 33.
000A 085Dh	LCDC	LCD Display Data Register 29	SEG29	8	8	1 or 2 PCLKB	section 33.
000A 085Eh	LCDC	LCD Display Data Register 30	SEG30	8	8	1 or 2 PCLKB	section 33.
000A 085Fh	LCDC	LCD Display Data Register 31	SEG31	8	8	1 or 2 PCLKB	section 33.
000A 0860h	LCDC	LCD Display Data Register 32	SEG32	8	8	1 or 2 PCLKB	section 33.
000A 0861h	LCDC	LCD Display Data Register 33	SEG33	8	8	1 or 2 PCLKB	section 33.
000A 0862h	LCDC	LCD Display Data Register 34	SEG34	8	8	1 or 2 PCLKB	section 33.
000A 0863h	LCDC	LCD Display Data Register 35	SEG35	8	8	1 or 2 PCLKB	section 33.
000A 0864h	LCDC	LCD Display Data Register 36	SEG36	8	8	1 or 2 PCLKB	section 33.
000A 0865h	LCDC	LCD Display Data Register 37	SEG37	8	8	1 or 2 PCLKB	section 33.
000A 0866h	LCDC	LCD Display Data Register 38	SEG38	8	8	1 or 2 PCLKB	section 33.
000A 0867h	LCDC	LCD Display Data Register 39	SEG39	8	8	1 or 2 PCLKB	section 33.
000A 0900h	CTSU	CTSU Control Register 0	CTSUCR0	8	8	1 or 2 PCLKB	section 34.
000A 0901h	CTSU	CTSU Control Register 1	CTSUCR1	8	8	1 or 2 PCLKB	section 34.
000A 0902h	CTSU	CTSU Synchronous Noise Reduction Setting Register	CTSUSDPRS	8	8	1 or 2 PCLKB	section 34.
000A 0903h	CTSU	CTSU Sensor Stabilization Wait Time Register	CTSUSST	8	8	1 or 2 PCLKB	section 34.
000A 0904h	CTSU	CTSU Measurement Channel Register 0	CTSUSMCH0	8	8	1 or 2 PCLKB	section 34.
000A 0905h	CTSU	CTSU Measurement Channel Register 1	CTSUSMCH1	8	8	1 or 2 PCLKB	section 34.
000A 0906h	CTSU	CTSU Channel Enable Control Register 0	CTSUCHAC0	8	8	1 or 2 PCLKB	section 34.
000A 0907h	CTSU	CTSU Channel Enable Control Register 1	CTSUCHAC1	8	8	1 or 2 PCLKB	section 34.
000A 090Bh	CTSU	CTSU Channel Transmit/Receive Control Register 0	CTSUCHTRC0	8	8	1 or 2 PCLKB	section 34.
000A 090Ch	CTSU	CTSU Channel Transmit/Receive Control Register 1	CTSUCHTRC1	8	8	1 or 2 PCLKB	section 34.
000A 0910h	CTSU	CTSU High-Pass Noise Reduction Control Register	CTSUDCLKC	8	8	1 or 2 PCLKB	section 34.
000A 0911h	CTSU	CTSU Status Register	CTSUST	8	8	1 or 2 PCLKB	section 34.
000A 0912h	CTSU	CTSU High-Pass Noise Spectrum Diffusion Control Register	CTSUSSC	16	16	1 or 2 PCLKB	section 34.
000A 0914h	CTSU	CTSU Sensor Offset Register 0	CTSUSO0	16	16	1 or 2 PCLKB	section 34.
000A 0916h	CTSU	CTSU Sensor Offset Register 1	CTSUSO1	16	16	1 or 2 PCLKB	section 34.
000A 0918h	CTSU	CTSU Sensor Counter	CTSUSC	16	16	1 or 2 PCLKB	section 34.
000A 091Ah	CTSU	CTSU Reference Counter	CTSURC	16	16	1 or 2 PCLKB	section 34.
000A 091Ch	CTSU	CTSU Error Status Register	CTSUERRS	16	16	1 or 2 PCLKB	section 34.
007F C090h	FLASH	E2 DataFlash Control Register	DFLCTL	8	8	2 or 3 FCLK	section 41.
007F C0ACh	TEMPS	Temperature Sensor Calibration Data Register	TSCDRL	8	8	1 or 2 PCLKB	section 37.

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

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States	Reference Section
007F C0ADh	TEMPS	Temperature Sensor Calibration Data Register	TSCDRH	8	8	1 or 2 PCLKB	section 37.
007F C0B0h	FLASH	Flash Start-Up Setting Monitor Register	FSCMR	16	16	2 or 3 FCLK	section 41.
007F C0B2h	FLASH	Flash Access Window Start Address Monitor	FAWSMR	16	16	2 or 3 FCLK	section 41.
007F C0B4h	FLASH	Flash Access Window End Address Monitor Register	FAWEMR	16	16	2 or 3 FCLK	section 41.
007F C0B6h	FLASH	Flash Initial Setting Register	FISR	8	8	2 or 3 FCLK	section 41.
007F C0B7h	FLASH	Flash Extra Area Control Register	FEXCR	8	8	2 or 3 FCLK	section 41.
007F C0B8h	FLASH	Flash Error Address Monitor Register L	FEAML	16	16	2 or 3 FCLK	section 41.
007F C0BAh	FLASH	Flash Error Address Monitor Register H	FEAMH	8	8	2 or 3 FCLK	section 41.
007F C0C0h	FLASH	Protection Unlock Register	FPR	8	8	2 or 3 FCLK	section 41.
007F C0C1h	FLASH	Protection Unlock Status Register	FPSR	8	8	2 or 3 FCLK	section 41.
007F C0C2h	FLASH	Flash Read Buffer Register L	FRBL	16	16	2 or 3 FCLK	section 41.
007F C0C4h	FLASH	Flash Read Buffer Register H	FRBH	16	16	2 or 3 FCLK	section 41.
007F FF80h	FLASH	Flash P/E Mode Control Register	FPMCR	8	8	2 or 3 FCLK	section 41.
007F FF81h	FLASH	Flash Area Select Register	FASR	8	8	2 or 3 FCLK	section 41.
007F FF82h	FLASH	Flash Processing Start Address Register L	FSARL	16	16	2 or 3 FCLK	section 41.
007F FF84h	FLASH	Flash Processing Start Address Register H	FSARH	8	8	2 or 3 FCLK	section 41.
007F FF85h	FLASH	Flash Control Register	FCR	8	8	2 or 3 FCLK	section 41.
007F FF86h	FLASH	Flash Processing End Address Register L	FEARL	16	16	2 or 3 FCLK	section 41.
007F FF88h	FLASH	Flash Processing End Address Register H	FEARH	8	8	2 or 3 FCLK	section 41.
007F FF89h	FLASH	Flash Reset Register	FRESETR	8	8	2 or 3 FCLK	section 41.
007F FF8Ah	FLASH	Flash Status Register 0	FSTATR0	8	8	2 or 3 FCLK	section 41.
007F FF8Bh	FLASH	Flash Status Register 1	FSTATR1	8	8	2 or 3 FCLK	section 41.
007F FF8Ch	FLASH	Flash Write Buffer Register L	FWBL	16	16	2 or 3 FCLK	section 41.
007F FF8Eh	FLASH	Flash Write Buffer Register H	FWBH	16	16	2 or 3 FCLK	section 41.
007F FFB2h	FLASH	Flash P/E Mode Entry Register	FENTRYR	16	16	2 or 3 FCLK	section 41.

Note 1. Odd addresses cannot be accessed in 16-bit units. When accessing a register in 16-bit units, access the address of the TMR0 or TMR2 register. Table 22.4 lists register allocation for 16-bit access.

Note 2. Odd addresses cannot be accessed in 16-bit units. When accessing a register in 16-bit units, access the address of the TMOCTL register. Table 29.6 lists register allocation for 16-bit access.

## 6. Resets

### 6.1 Overview

There are six types of resets: RES# pin reset, power-on reset, voltage monitoring 1 reset, voltage monitoring 2 reset, independent watchdog timer reset, and software reset.

Table 6.1 lists the reset names and sources.

**Table 6.1 Reset Names and Sources**

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

Note 1. For details on the voltages to be monitored (VPOR, Vdet1, and Vdet2), refer to section 8, Voltage Detection Circuit (LVDAa) and section 42, Electrical Characteristics

The internal state and pins are initialized by a reset.

Table 6.2 lists the reset targets to be initialized.

**Table 6.2 Targets Initialized According to Reset Source**

Target to be Initialized	Reset Source					
	RES# Pin Reset	Power-On Reset	Independent Watchdog Timer Reset	Voltage Monitoring 1 Reset	Voltage Monitoring 2 Reset	Software Reset
Power-on reset detect flag (RSTSR0.PORF)	○	—	—	—	—	—
Register related to the cold start/warm start determination flag (RSTSR1.CWSF)	— *1	○	—	—	—	—
Independent watchdog timer reset detect flag (RSTSR2.IWDTRF)	○	○	—	—	—	—
Registers related to the independent watchdog timer (IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDCSTPR, ILOCOCR)	○	○	—	—	—	—
Voltage monitoring 1 reset detect flag (RSTSR0.LVD1RF)	○	○	○	—	—	—
Registers related to the voltage monitor function 1 (LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL[3:0])	○	○	○	—	—	—
(LVD1CR1, LVD1SR)	○	○	○	—	—	—
Voltage monitoring 2 reset detect flag (RSTSR0.LVD2RF)	○	○	○	○	—	—
Registers related to the voltage monitor function 2 (LVD2CR0, EXVCCINP2, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL[1:0])	○	○	○	○	—	—
(LVD2CR1, LVD2SR)	○	○	○	○	—	—
Software reset detect flag (RSTSR2.SWRF)	○	○	○	○	○	—
Registers related to the realtime clock*2	—	—	—	—	—	—
Registers other than the above, the CPU, and internal state	○	○	○	○	○	○

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

Note 1. Initialized at a power-on.

Note 2. Some control bits (RCR1.CIE, RCR1.RTCOS, RCR2.RTCOE, ADJ30, and RESET) are initialized by all types of resets. For details on the target bits, refer to section 24, Realtime Clock (RTCA).

When a reset is canceled, the reset exception handling starts. For details on the reset exception handling, refer to section 13, Exception Handling.

Table 6.3 lists the pin related to the reset.

**Table 6.3 Pin Related to Reset**

Pin Name	I/O	Function
RES#	Input	Reset pin

## 6.2 Register Descriptions

### 6.2.1 Reset Status Register 0 (RSTSR0)

Address(es): 0008 C290h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LVD2R F	LVD1R F	—	PORF
Value after reset:	0	0	0	0	0*1	0*1	0	0*1

Note 1. The value after reset depends on the reset source.

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) *1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	LVD1RF	Voltage Monitoring 1 Reset Detect Flag	0: Voltage monitoring 1 reset not detected. 1: Voltage monitoring 1 reset detected.	R/(W) *1
b3	LVD2RF	Voltage Monitoring 2 Reset Detect Flag	0: Voltage monitoring 2 reset not detected. 1: Voltage monitoring 2 reset detected.	R/(W) *1
b7 to b4	—	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.

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

#### LVD1RF Flag (Voltage Monitoring 1 Reset Detect Flag)

The LVD1RF flag indicates that VCC voltage has fallen below Vdet1.

[Setting condition]

- When Vdet1-level VCC voltage is detected.

[Clearing conditions]

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

#### LVD2RF Flag (Voltage Monitoring 2 Reset Detect Flag)

The LVD2RF flag indicates that VCC voltage has fallen below Vdet2.

[Setting condition]

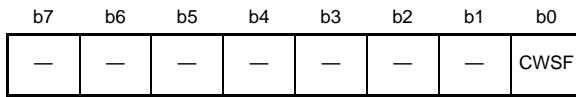
- When Vdet2-level VCC voltage is detected.

[Clearing conditions]

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

## 6.2.2 Reset Status Register 1 (RSTSR1)

Address(es): 0008 C291h



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

Note 1. The value after reset depends on the reset source.

Bit	Symbol	Bit Name	Description	R/W
b0	CWSF	Cold/Warm Start Determination Flag	0: Cold start 1: Warm start	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 to set the flag.

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

### CWSF Flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing: cold start or warm start.

The CWSF flag is initialized at a power-on.

[Setting condition]

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

[Clearing condition]

- When a reset listed in Table 6.2 occurs.

### 6.2.3 Reset Status Register 2 (RSTSR2)

Address(es): 0008 00C0h

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

Note 1. The value after reset depends on the reset source.

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) *1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	SWRF	Software Reset Detect Flag	0: Software reset not detected. 1: Software reset detected.	R/(W) *1
b7 to b3	—	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.

#### IWDTRF Flag (Independent Watchdog Timer Reset Detect Flag)

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

[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

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

#### SWRF Flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset has occurred.

[Setting condition]

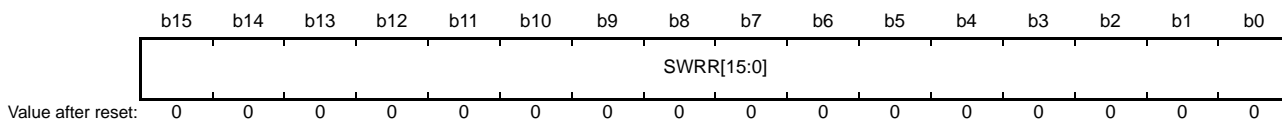
- When a software reset occurs.

[Clearing conditions]

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

### 6.2.4 Software Reset Register (SWRR)

Address(es): 0008 00C2h



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

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



## 6.3 Operation

### 6.3.1 RES# Pin Reset

This is a reset generated by the RES# pin.

When the RES# pin is driven low, all the processing in progress is aborted and the MCU enters a reset state.

To ensure the MCU is reset, 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 RES# post-cancellation wait time ( $t_{RESWT}$ ) has elapsed, and then the CPU starts the reset exception handling.

For details, refer to section 42, Electrical Characteristics.

### 6.3.2 Power-On Reset

The power-on reset is an internal reset generated by the power-on reset circuit.

A power-on reset is generated when power is supplied to the RES# pin while it is connected to VCC via a resistor. When connecting a capacitor to the RES# pin, also ensure that the voltage on the RES# pin is always at least  $V_{IH}$ . For details on  $V_{IH}$ , refer to section 42, Electrical Characteristics. After VCC has exceeded VPOR and the specified period (power-on reset time) has elapsed, the internal reset is canceled and the CPU starts the reset exception handling. The power-on reset time is a stabilization period for the external power supply and the MCU circuit. After a power-on reset has been generated, the RSTSR0.PORF flag is set to 1. The PORF flag is initialized by a RES# pin reset.

Figure 6.1 shows an example of the power-on reset circuit and its operation.

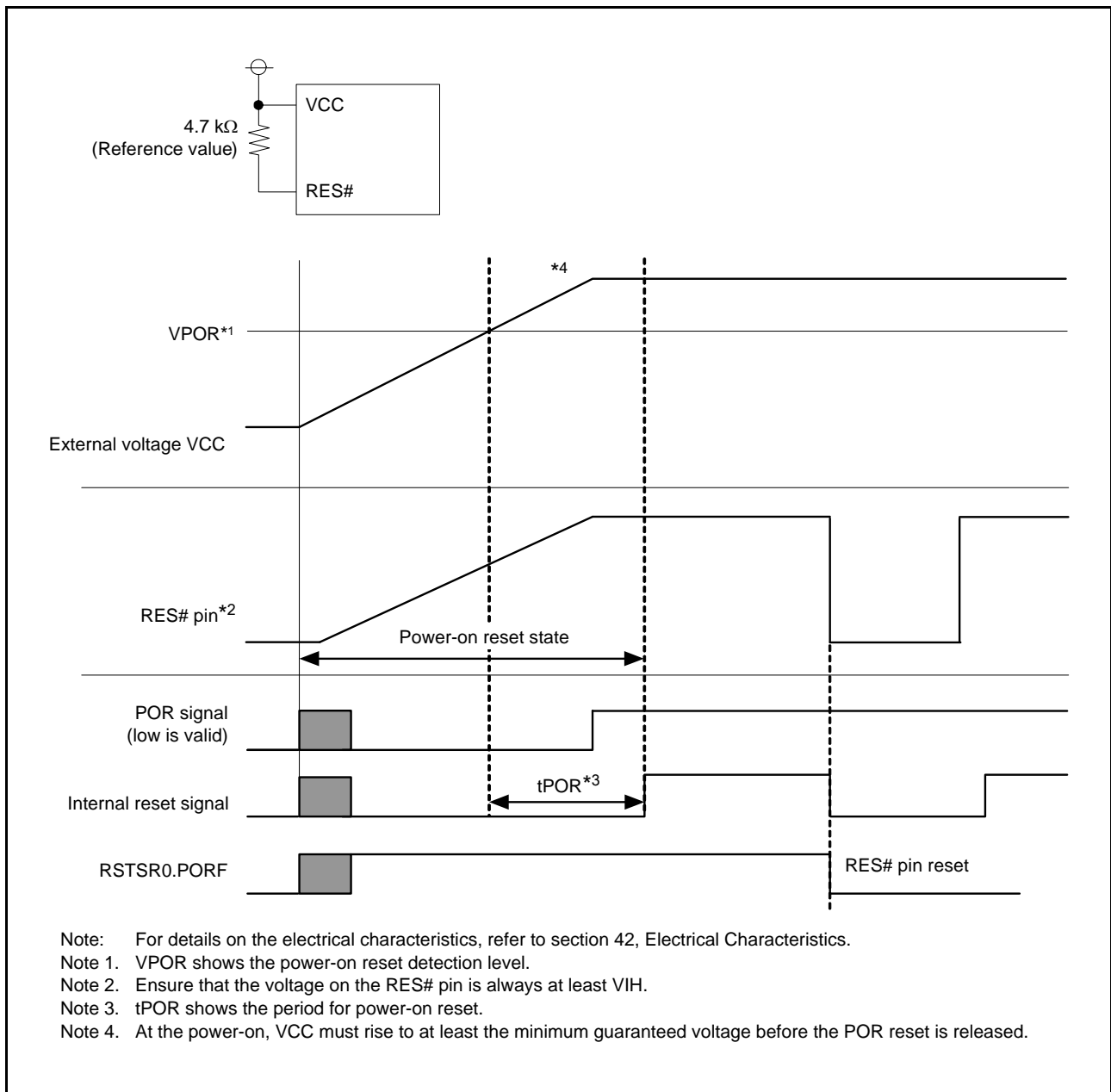


Figure 6.1 Operation Examples during Power-On Reset

### 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 (enabled) and the voltage monitoring 1 circuit mode select bit (LVD1RI) is set to 1 (voltage monitoring 1 reset enabled when the voltage falls to and below Vdet1) in 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.

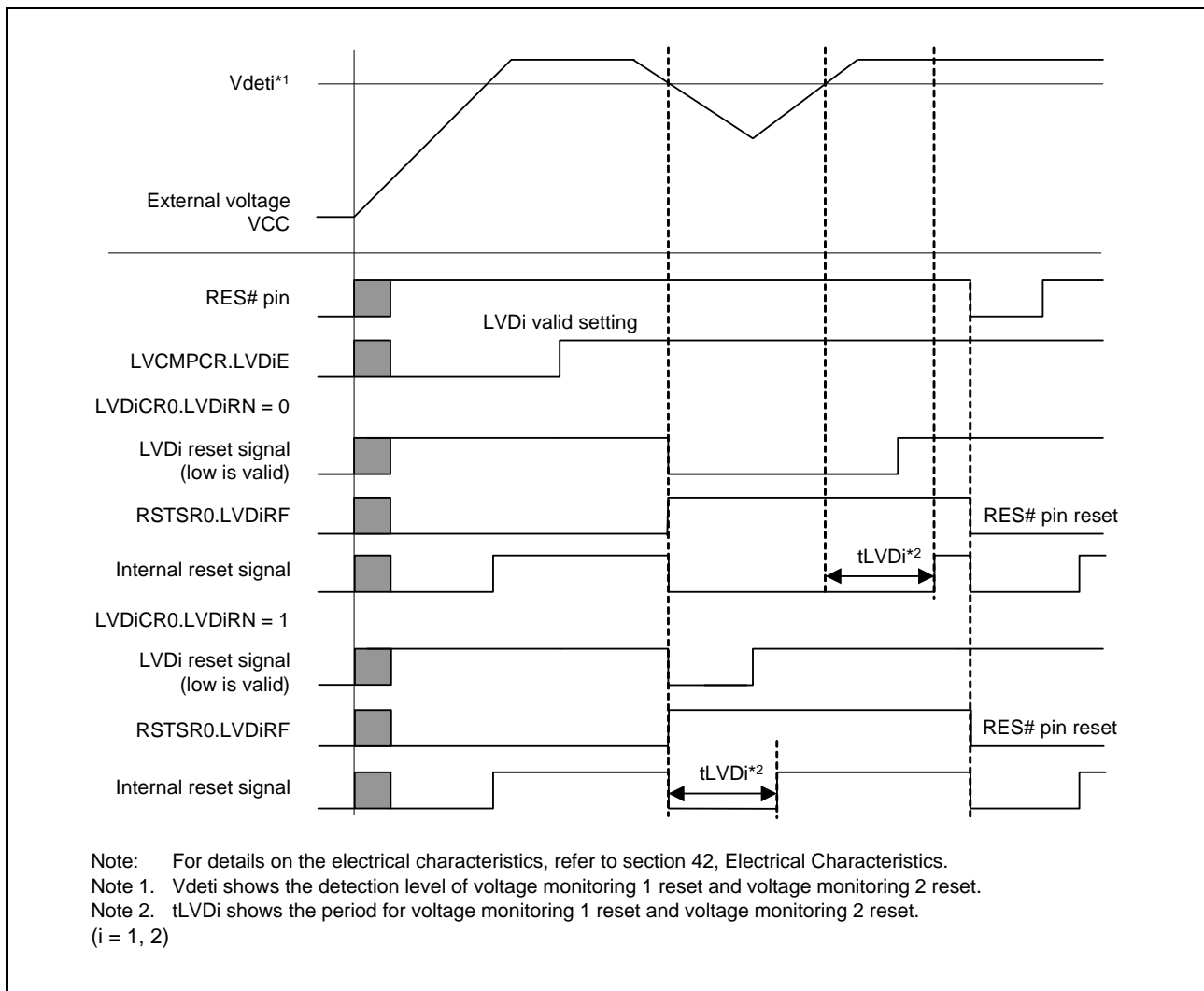
Likewise, when the voltage monitoring 2 interrupt/reset enable bit (LVD2RIE) is set to 1 (enabled) and the voltage monitoring 2 circuit mode select bit (LVD2RI) is set to 1 (voltage monitoring 2 reset enabled when the voltage falls to and below Vdet2) in voltage monitoring 2 circuit control register 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 register. When the LVD1CR0.LVD1RN bit is 0 and VCC has fallen to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling once the 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 the LVD2CR0 register. Detection levels Vdet1 and Vdet2 can be changed by settings in the voltage detection level select register (LVDLVLR).

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

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



**Figure 6.2 Operation Examples during Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset**

### 6.3.4 Independent Watchdog Timer Reset

The independent watchdog timer reset is an internal reset generated by the independent watchdog timer. Enabling and disabling output of the independent watchdog timer reset from the independent watchdog timer can be selected by setting the IWDTRCR and option function select register 0 (OFS0). When output of the independent watchdog timer reset is enabled, an independent watchdog timer reset is generated if the independent watchdog timer underflows, or if data is written outside the refresh-permitted period. After the independent watchdog timer reset has been generated and the internal reset time (tRESW2) has elapsed, the internal reset is canceled and the CPU starts the reset exception handling. For details on the independent watchdog timer reset, refer to section 25, Independent Watchdog Timer (IWDtA).

### 6.3.5 Software Reset

The software reset is an internal reset generated by the software reset circuit. A software reset is generated when A501h is written to the SWRR register. After the software reset has been generated and tRESW2 has elapsed, the internal reset is canceled and the CPU starts the reset exception handling.

### 6.3.6 Determination of Cold/Warm Start

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

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

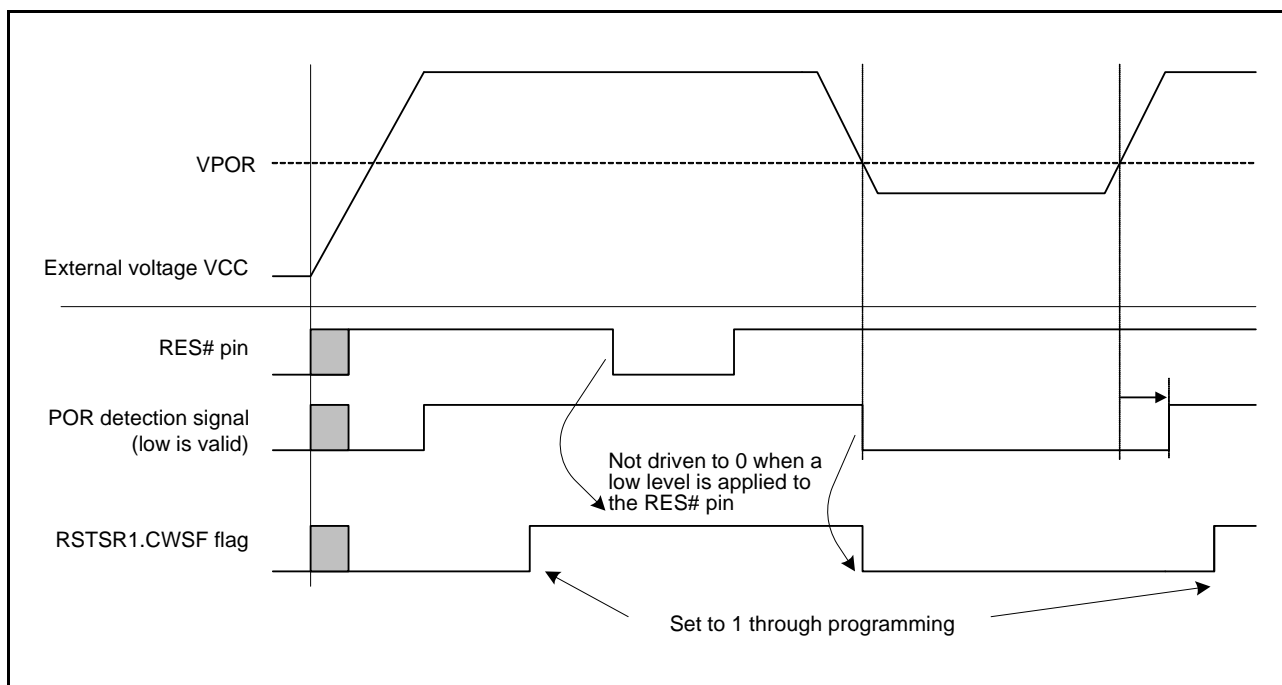


Figure 6.3 Example of Cold/Warm Start Determination Operation

### 6.3.7 Determination of Reset Generation Source

Reading the RSTSR0 and RSTSR2 registers 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.

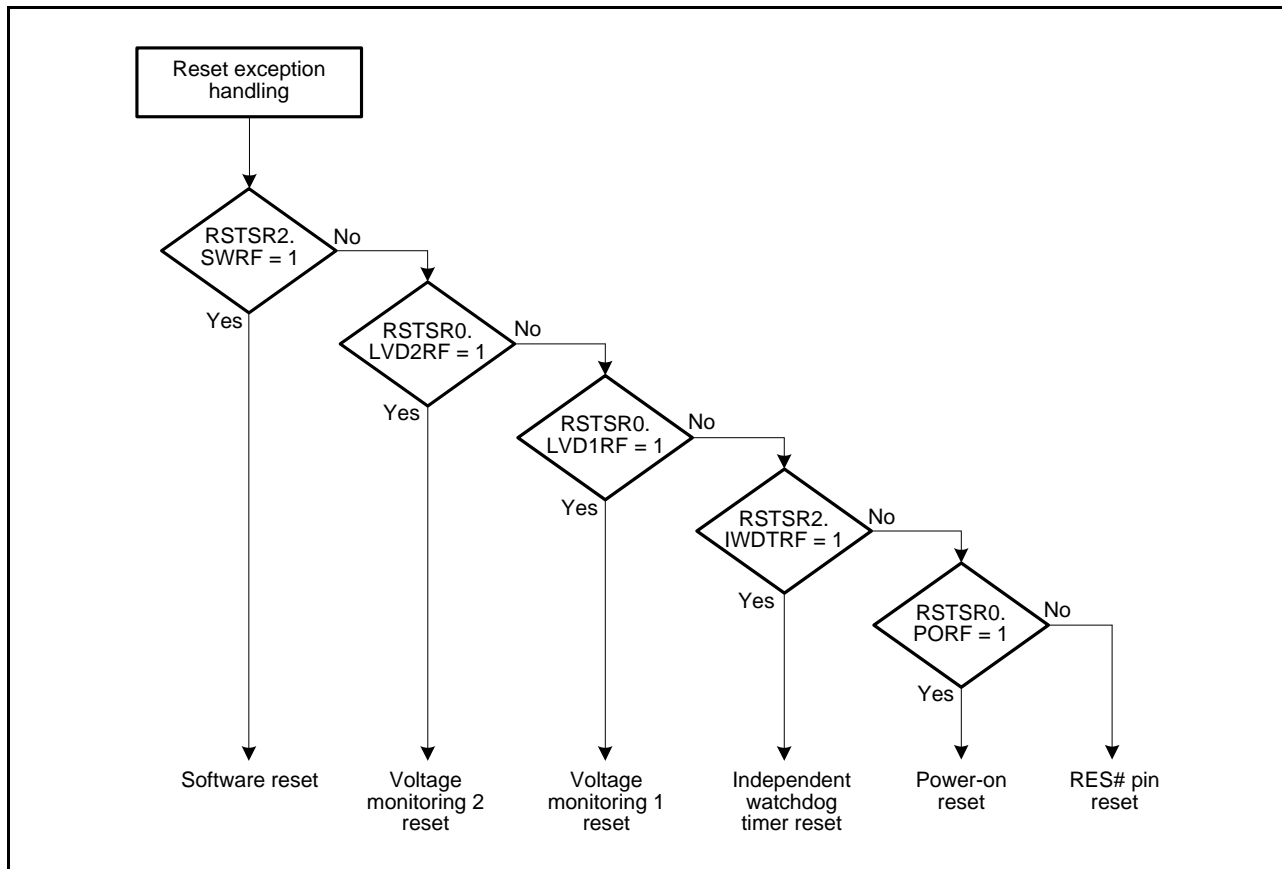


Figure 6.4 Example of Reset Generation Source Determination Flow

## 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.

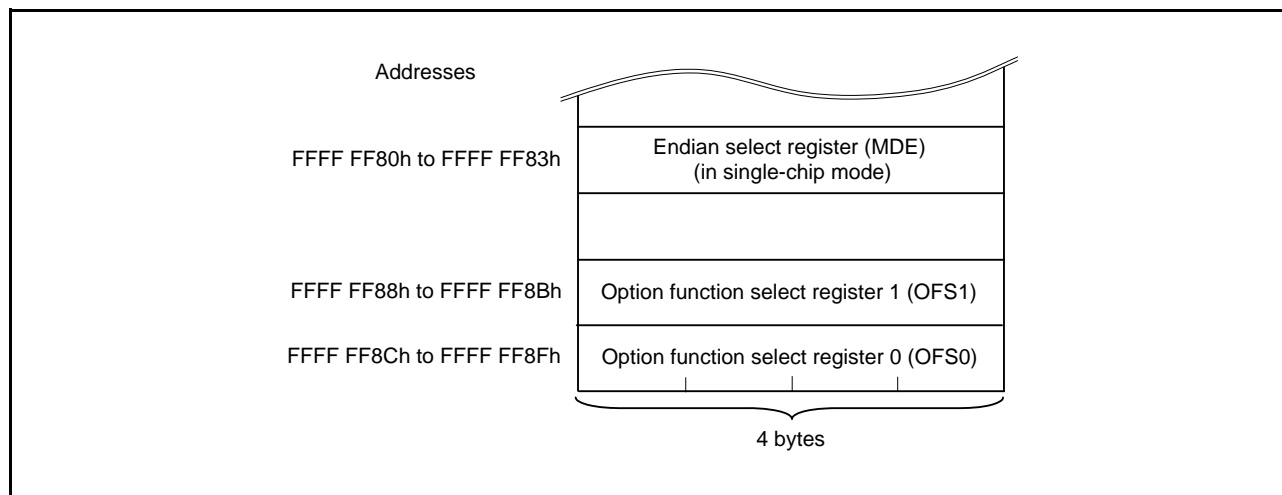


Figure 7.1 Option-Setting Memory Area

## 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
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

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	—	IWDTR STIRQS	IWDTRPSS[1:0]	IWDTRPES[1:0]	IWDTCKS[3:0]			IWDTTOPS[1:0]	IWDTS TRT	—				

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

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

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b1	IWDTSTRT	IWDT Start Mode Select	0: IWDT is automatically activated in auto-start mode after a reset 1: IWDT is halted after a reset	R
b3, b2	IWDTTOPS[1:0]	IWDT Timeout Period Select	b3 b2 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	R
b7 to b4	IWDTCKS[3:0]	IWDT Clock Frequency Division Ratio Select	b7 b4 0 0 0 0: x1 (Cycle period: 136 ms) 0 0 1 0: x1/16 (Cycle period: 2.18 s) 0 0 1 1: x1/32 (Cycle period: 4.36 s) 0 1 0 0: x1/64 (Cycle period: 8.73 s) 1 1 1 1: x1/128 (Cycle period: 17.5 s) 0 1 0 1: x1/256 (Cycle period: 34.9 s) Settings other than above are prohibited.	R
b9, b8	IWDTRPES[1:0]	IWDT Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R
b11, b10	IWDTRPSS[1:0]	IWDT Window Start Position Select	b11 b10 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R
b12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select	0: Non-maskable interrupt request is enabled 1: Reset is enabled	R
b13	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b14	IWDTS LCSTP	IWDT Sleep Mode Count Stop Control	0: Counting stop is disabled 1: Counting stop is enabled when entering sleep, software standby mode, and deep sleep mode	R
b31 to b15	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

The OFS0 register selects the operations of the independent watchdog timer (IWDT) after a reset.

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 boot mode.

**IWDTSTRT Bit (IWDT Start Mode Select)**

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

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

These bits select the timeout period, i.e. the time it takes for the down-counter to underflow, as 128, 512, 1024, or 2048 cycles of the frequency-divided clock set by the IWDTCKS[3:0] bits. The time (number of 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, refer to section 25, 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 128 to 524288 clock cycles for the IWDT.

For details, refer to section 25, 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 IWDRPSS[1:0] and IWDRPES[1:0] bits vary with the setting of the IWDTTOPS[1:0] bits.

For details, refer to section 25, Independent Watchdog Timer (IWDTa).

**IWDRPSS[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 25, Independent Watchdog Timer (IWDTa).

**IWDRSTIRQS 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 25, Independent Watchdog Timer (IWDTa).

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

This bit selects to stop counting when entering sleep, software standby mode, and deep sleep mode.

For details, refer to section 25, Independent Watchdog Timer (IWDTa).

## 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	STUPLVD1LVL[3:0]			—	—	STUPLV D1REN	FASTS TUP	

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

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

Bit	Symbol	Bit Name	Description	R/W
b0	FASTSTUP	Power-On Fast Startup Time	0: Fast startup time at power on 1: Normal startup	R
b1	STUPLVD1REN	Startup Voltage Monitoring 1 Reset Enable	0: Voltage monitoring 1 reset is enabled at startup 1: Voltage monitoring 1 reset is disabled at startup	R
b3, b2	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b7 to b4	STUPLVD1LVL[3:0]	Startup Voltage Monitoring 1 Reset Detection Level Select	b7 b4 0 1 0 0: 3.10 V 0 1 0 1: 3.00 V 0 1 1 0: 2.90 V 0 1 1 1: 2.79 V 1 0 0 0: 2.68 V 1 0 0 1: 2.58 V 1 0 1 0: 2.48 V 1 0 1 1: 2.06 V 1 1 0 0: 1.96 V 1 1 0 1: 1.86 V Settings other than above are prohibited when the STUPLVD1REN bit is 0.	R
b8	HOCOEN	HOCO Oscillation Enable	0: HOCO oscillation is enabled after a reset 1: HOCO oscillation is disabled after a reset	R
b31 to b9	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

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 boot mode.

**FASTSTUP Bit (Power-On Fast Startup Time)**

The startup time can be reduced by setting this bit to 0 (fast startup time at power on) when it is possible to meet the power-on VCC rising gradient (during fast startup time) shown in Electrical Characteristics.

Do not set this bit to 0 when it is not possible to meet the power-on VCC rising gradient (during fast startup time).

**STUPLVD1REN Bit (Startup Voltage Monitoring 1 Reset Enable)**

This bit selects whether the voltage monitoring 1 reset is enabled or disabled after a reset.

The Vdet1 voltage to be monitored by the voltage detection 1 circuit is selected by the STUPLVD1LVL[3:0] bits.

When this bit is set to 0 (voltage monitoring 1 reset is enabled at startup) and the power is turned on, the power-on VCC rising gradient specification is the power-on VCC rising gradient (when voltage monitoring 1 reset is enabled at startup) shown in section 42, Electrical Characteristics, and there is no specified maximum value.

When the STUPLVD1REN bit is set to 0, the MCU starts up with voltage monitoring 1 reset enabled, regardless of the FASTSTUP bit setting.

**STUPLVD1LVL[3:0] Bits (Startup Voltage Monitoring 1 Reset Detection Level Select)**

These bits select the voltage detection level to be monitored by the voltage detection 1 circuit when the STUPLVD1REN bit is set to 0.

**HOCOEN Bit (HOCO Oscillation Enable)**

This bit selects whether the HOCO oscillation enable bit is enabled or disabled after a reset.

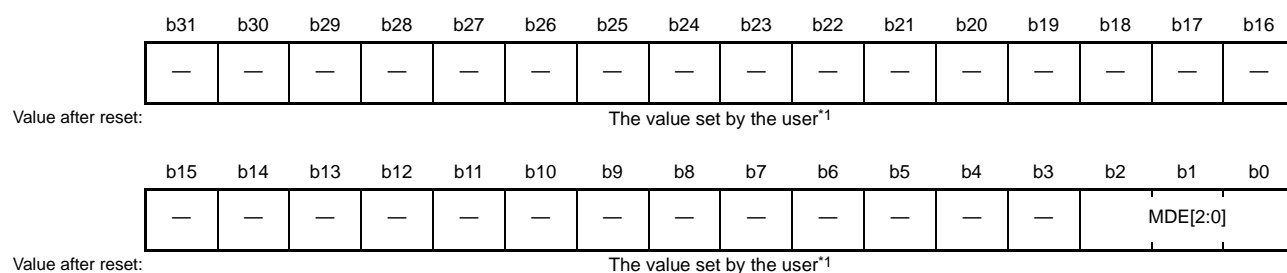
Setting the HOCOEN bit to 0 allows the HOCO oscillation to be started before the CPU starts operation, and therefore reduces the waiting time for oscillation stabilization.

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.

Also, when the OFS1.HOCOEN bit is set to 0, the HOCO oscillation stabilization time (tHOCO) is secured by hardware, so the clock with the accuracy of the HOCO oscillation frequency (fHOCO) shown in Electrical Characteristics is supplied after release from the CPU reset state.

### 7.2.3 Endian Select Register (MDE)

Address(es): FFFF FF80h: MDE (in single-chip mode)



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

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

The MDE register selects the endian for the CPU. The endian select register (MDE) at address FFFF FF80h is used to select the endian.

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

When erasing the block including the MDE register, the MDE register value becomes FFFF FFFFh.

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

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

## 7.3 Usage Note

### 7.3.1 Setting Example of Option-Setting Memory

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

- To set ffff ff8h in the OFS0 register

```
.org 0fff ff8ch
```

```
.lword 0ffffff8h
```

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

## 8. Voltage Detection Circuit (LVDAa)

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

### 8.1 Overview

In voltage detection 1, the detection voltage can be selected from 10 levels using the voltage detection level select register (LVDLVLR).

In voltage detection 2, the detection voltage can be selected from four levels by switching between input voltages to VCC and the CMPA2 pin.

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 LVD Specifications**

Item		Voltage Monitoring 1	Voltage Monitoring 2
VCC monitoring	Monitored voltage	Vdet1	Vdet2
	Detection target	When voltage rises above or drops below Vdet1	When voltage rises above or drops below Vdet2
	Detection voltage	Voltage selectable from 10 levels using the LVDLVLR.LVD1LVL[3:0] bits	Voltage selectable from four levels using the LVDLVLR.LVD2LVL[1:0] bits
	Monitoring flag	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 1 reset Reset when Vdet1 > VCC CPU restart timing selectable: after specified time with VCC > Vdet1 or Vdet1 > VCC	Voltage monitoring 2 reset Reset when Vdet2 > VCC or the CMPA2 pin CPU restart timing selectable: after specified time with VCC or the CMPA2 pin > Vdet2 or after specified time with Vdet2 > VCC or the CMPA2 pin
		Interrupt	Voltage monitoring 1 interrupt Non-maskable or maskable interrupt is selectable Interrupt request issued when Vdet1 > VCC and VCC > Vdet1 or either
	Event link function	Available Vdet1 passage detection event output	Not available

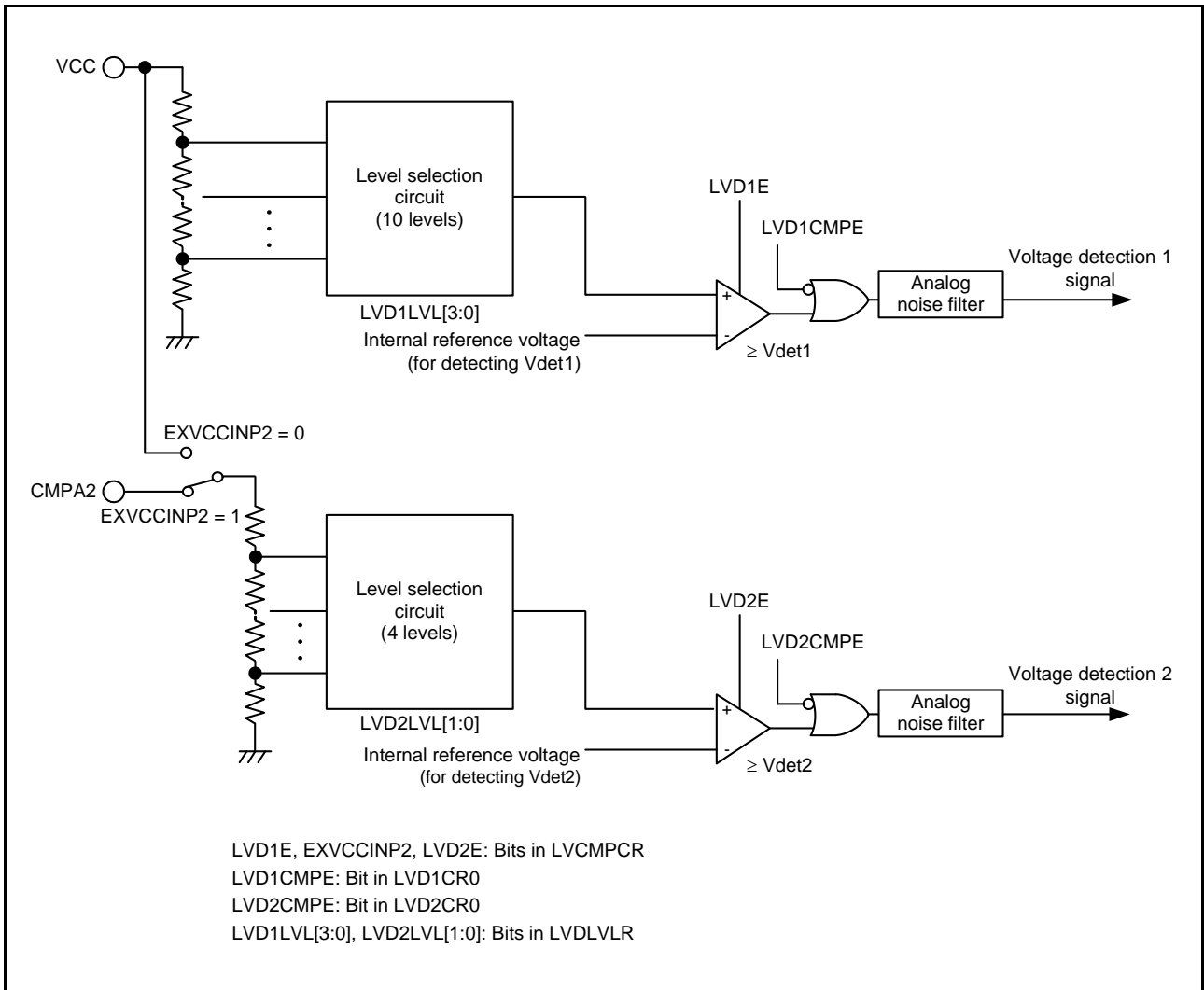


Figure 8.1 Block Diagram of the LVD

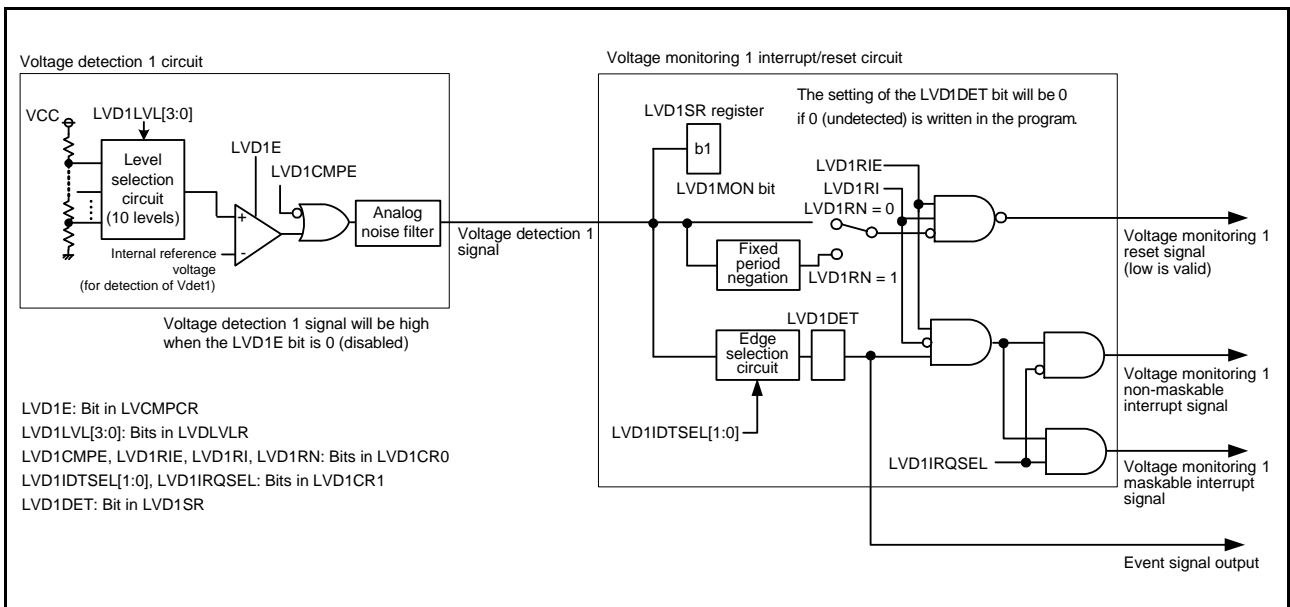


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

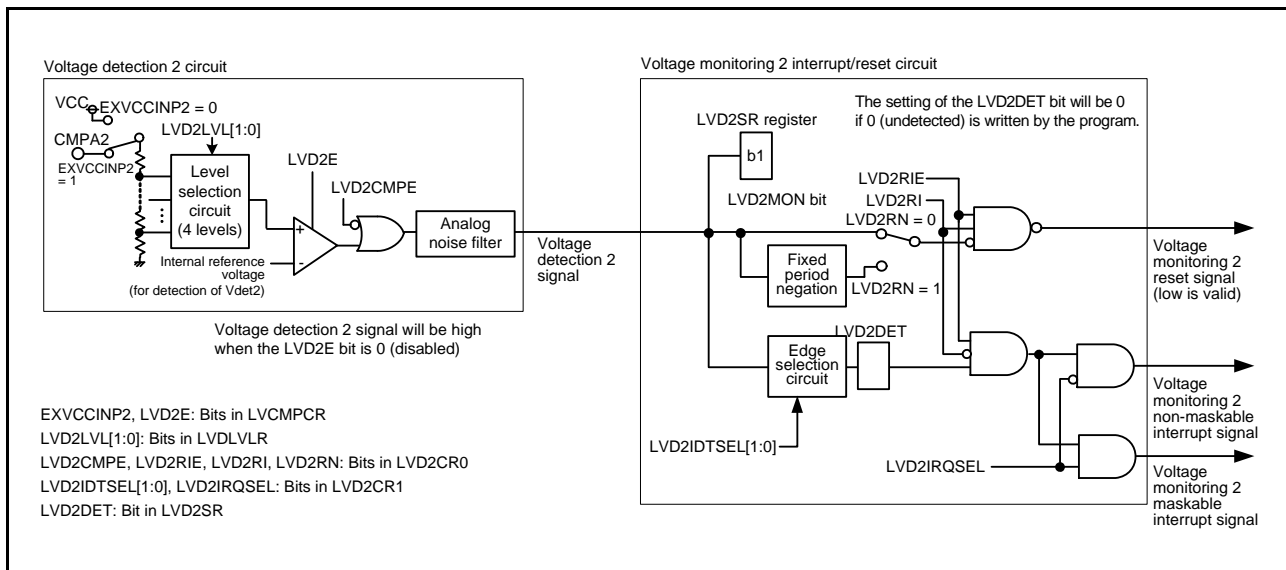


Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit

Table 8.2 lists the I/O pins relevant to the voltage detection circuit.

Table 8.2 I/O Pins of the Voltage Detection Circuit

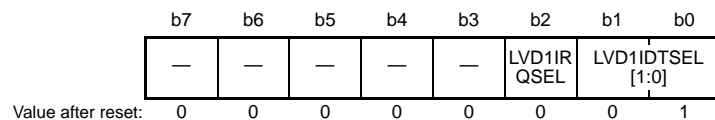
Pin Name	I/O	Function
CMPA2	Input	Detection target voltage pin for voltage detection 2



## 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 ELC Event Generation Condition Select	b1 b0 0 0: When VCC $\geq$ Vdet1 (rise) is detected 0 1: When VCC < Vdet1 (drop) is detected 1 0: When drop and rise are detected 1 1: Setting prohibited	R/W
b2	LVD1IRQSEL	Voltage Monitoring 1 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

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

Address(es): 0008 00E1h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	LVD1M ON	LVD1D ET

Value after reset: 0 0 0 0 0 0 1 0

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

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

Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes two system clock cycles for the bit to be read as 0.

### LVD1DET Flag (Voltage Monitoring 1 Voltage Change Detection Flag)

The LVD1DET flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).

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

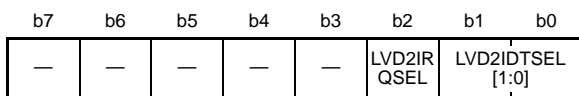
Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles of PCLKB may have to be secured as waiting time.

### 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



Value after reset: 0 0 0 0 0 0 0 1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD2IDTSEL [1:0]	Voltage Monitoring 2 Interrupt Generation Condition Select	b1 b0 0 0: When VCC or the CMPA2 pin $\geq$ Vdet2 (rise) is detected 0 1: When VCC or the CMPA2 pin $<$ Vdet2 (drop) is detected 1 0: When drop and rise are detected 1 1: Setting prohibited	R/W
b2	LVD2IRQSEL	Voltage Monitoring 2 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

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

Address(es): 0008 00E3h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	LVD2MON	LVD2DET

Value after reset: 0 0 0 0 0 0 1 0

Bit	Symbol	Bit Name	Description	R/W
b0	LVD2DET	Voltage Monitoring 2 Voltage Change Detection Flag	0: Not detected 1: Vdet2 passage detection	R/(W) *1
b1	LVD2MON	Voltage Monitoring 2 Signal Monitor Flag	0: VCC or the CMPA2 pin < Vdet2 1: VCC or the CMPA2 pin ≥ Vdet2 or LVD2MON is disabled	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes two system clock cycles for the bit to be read as 0.

### LVD2DET Flag (Voltage Monitoring 2 Voltage Change Detection Flag)

The LVD2DET flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.LVD2CMPE bit is 1 (voltage monitoring 2 circuit comparison result output enabled).

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

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

### 8.2.5 Voltage Monitoring Circuit Control Register (LVCMPCR)

Address(es): 0008 C297h

b7	b6	b5	b4	b3	b2	b1	b0
—	LVD2E	LVD1E	—	EXVCC INP2	—	—	—

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

Note 1. The value after a reset is 1 when the OSF1.STUPLVD1REN bit is 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	EXVCCINP2	Voltage Detection 2 Comparison Voltage External Input Select *1	0: Power supply voltage (VCC) 1: CMPA2 pin input voltage	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	LVD1E	Voltage Detection 1 Enable	0: Voltage detection 1 circuit disabled 1: Voltage detection 1 circuit enabled	R/W
b6	LVD2E	Voltage Detection 2 Enable	0: Voltage detection 2 circuit disabled 1: Voltage detection 2 circuit enabled	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

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

Note 1. The EXVCCINP2 bit can be changed only when the LVD1E and LVD2E bits are both 0 (voltage detection 1 circuit and voltage detection 2 circuit disabled).

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

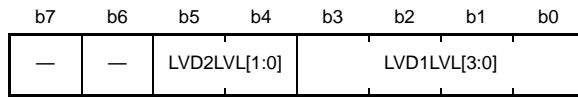
When using voltage detection 1 interrupt/reset or the LVD1SR.LVD1MON flag, set the LVD1E bit to 1. The voltage detection 1 circuit starts once  $t_d(E-A)$  passes after the LVD1E bit value is changed from 0 to 1.

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

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

## 8.2.6 Voltage Detection Level Select Register (LVDLVLR)

Address(es): 0008 C298h



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

Note 1. The value after a reset is the same as the value of the OFS1.STUPLVD1LVL[3:0] bits when the OSF1.STUPLVD1REN bit is 0.

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	LVD1LVL[3:0]	Voltage Detection 1 Level Select (Standard voltage during drop in voltage)	b3 b0 0 1 0 0: 3.10 V 0 1 0 1: 3.00 V 0 1 1 0: 2.90 V 0 1 1 1: 2.79 V 1 0 0 0: 2.68 V 1 0 0 1: 2.58 V 1 0 1 0: 2.48 V 1 0 1 1: 2.06 V 1 1 0 0: 1.96 V 1 1 0 1: 1.86 V Settings other than those listed above are prohibited.	R/W
b5, b4	LVD2LVL[1:0]	Voltage Detection 2 Level Select (Standard voltage during drop in voltage)	b5 b4 0 0: 2.90 V 0 1: 2.60 V 1 0: 2.00 V 1 1: 1.80 V *1	R/W
b7, b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

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

Note 1. Do not set these bits to 11b when the LVCMPER.EXVCCINP2 bit is 0 (power supply voltage (VCC)).

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

When a setting is made so that the voltage detection level range set by the LVD1LVL register overlaps with the range set by the LVD2LVL register, it cannot be specified which of LVD1 and LVD2 is used for voltage detection. For details on the voltage detection level range, refer to section 42, Electrical Characteristics.

## 8.2.7 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0)

Address(es): 0008 C29Ah

b7	b6	b5	b4	b3	b2	b1	b0
LVD1RN	LVD1RI	—	—	—	LVD1CMPE	—	LVD1RIE

Value after reset: 1 \*1 0 \*2 0 0 X 0 \*2 0 0 \*2

x: Undefined

Note 1. The value after a reset is 0 when the OSF1.STUPLVD1REN bit is 0.

Note 2. The value after a reset is 1 when the OSF1.STUPLVD1REN bit is 0.

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

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

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

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

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

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

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

## 8.2.8 Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0)

Address(es): 0008 C29Bh

b7	b6	b5	b4	b3	b2	b1	b0
LVD2RN	LVD2RI	—	—	—	LVD2CMPE	—	LVD2RIE

Value after reset: 1 0 0 0 X 0 0 0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	LVD2RIE	Voltage Monitoring 2 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	LVD2CMPE	Voltage Monitoring 2 Circuit Comparison Result Output Enable	0: Voltage monitoring 2 circuit comparison results output disabled 1: Voltage monitoring 2 circuit comparison results output enabled	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	LVD2RI	Voltage Monitoring 2 Circuit Mode Select	0: Voltage monitoring 2 interrupt during Vdet2 passage 1: Voltage monitoring 2 reset enabled when the voltage falls to and below Vdet2	R/W
b7	LVD2RN	Voltage Monitoring 2 Circuit Reset Negation Select	0: Negation follows a stabilization time (tLVD2) after VCC or the CMPA2 pin > Vdet2 is detected. 1: Negation follows a stabilization time (tLVD2) after assertion of the voltage monitoring 2 reset.	R/W

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

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

The LVD2RIE bit is enabled when the LVCMPER.LVD2E bit is set to 1 (voltage detection 2 circuit enabled) and the LVD2CMPE bit is set to 1 (voltage monitoring 2 circuit comparison results output enabled).

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

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

If the LVD2RN bit is to be set to 1 (negation follows a stabilization time after assertion of the voltage monitoring 2 reset), set the LOCOCR.LCSTP bit to 0 (LOCO is operating). Furthermore, if a transition to software standby mode, the only possible value for the LVD2RN bit is 0 (negation follows a stabilization time after VCC or the CMPA2 pin > Vdet2 is detected). Do not set the LVD2RN bit to 1 (negation follows a stabilization time after assertion of the voltage monitoring 2 reset).



### 8.3 VCC Input Voltage Monitor

#### 8.3.1 Monitoring Vdet1

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

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

#### 8.3.2 Monitoring Vdet2

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

- (1) Specify the detection voltage by setting the LVDLVLR.LVD2LVL[1:0] bits (voltage detection 2 level).
- (2) Set the LVCMPCR.EXVCCINP2 bit to 0 (VCC voltage) or 1 (CMPA2 pin input voltage).
- (3) Set the LVCMPCR.LVD2E bit to 1 (voltage detection 2 circuit enabled).
- (4) After waiting for  $t_d(E-A)$ , set the LVD2CR0.LVD2CMPE bit to 1 (voltage monitoring 2 circuit comparison results output enabled).

## 8.4 Interrupt and Reset from Voltage Monitoring 1

Table 8.3 shows the procedures for setting bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. Table 8.4 shows the procedures for stopping bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. Figure 8.4 shows an example of operations for a voltage monitoring 1 interrupt. For the operation of the voltage monitoring 1 reset, see Figure 6.2 in section 6, Resets.

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

Step	Voltage Monitoring 1 Interrupt, Voltage Monitoring 1 ELC Event Output	Voltage Monitoring 1 Reset
1 <sup>*1</sup>	Select the detection voltage by setting the LVDLVL.R.LVD1LVL[3:0] bits.	
2 <sup>*1</sup>	Set the LVD1CR0.LVD1RI bit to 0 (voltage monitoring 1 interrupt).	Set the LVD1CR0.LVD1RI bit to 1 (voltage monitoring 1 reset). Select the type of reset negation by setting the LVD1CR0.LVD1RN bit.
3	Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits. Select the type of interrupt by setting the LVD1CR1.LVD1IRQSEL bit.	—
4	—	Set the LVD1CR0.LVD1RIE bit to 1 (voltage monitoring 1 interrupt/reset enabled).
5 <sup>*1</sup>	Set the LVCMP.R.LVD1E bit to 1 (voltage detection 1 circuit enabled).	
6 <sup>*1</sup>	Wait for at least $t_d(E-A)$ .	
7	Set the LVD1CR0.LVD1CMPE bit to 1 (voltage monitoring 1 circuit comparison results output enabled).	
8	Set the LVD1SR.LVD1DET bit to 0.	—
9	Set the LVD1CR0.LVD1RIE bit to 1 (voltage monitoring 1 interrupt/reset enabled).	—

Note 1. Steps 1, 2, 5, and 6 are not required if operation is with the setting to select the voltage monitoring 1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR1.LVD1IRQSEL and LVD1IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 9.

**Table 8.4 Procedures for Stopping Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset**

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

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

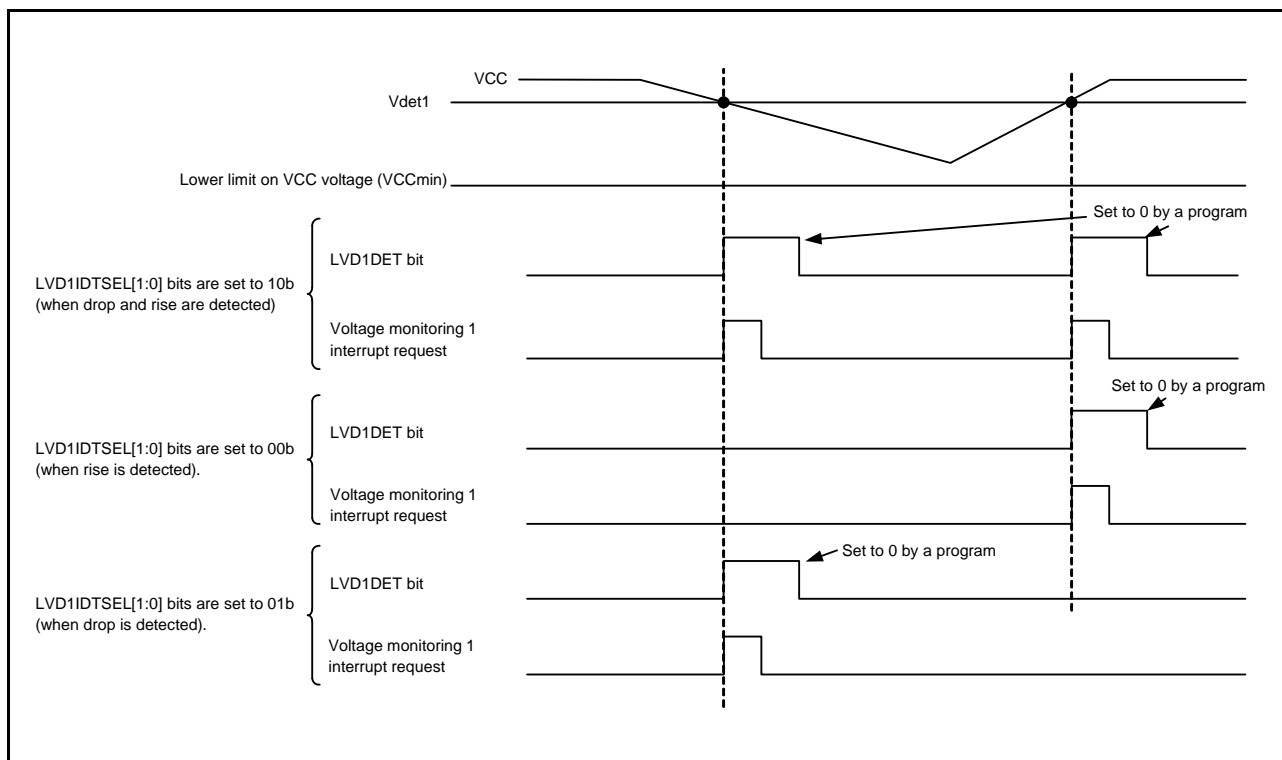


Figure 8.4 Example of Voltage Monitoring 1 Interrupt Operation

## 8.5 Interrupt and Reset from Voltage Monitoring 2

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

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

Step	Voltage Monitoring 2 Interrupt	Voltage Monitoring 2 Reset
1 <sup>*1</sup>	Select the detection voltage by setting the LVDLVL.R.LVD2LVL[1:0] bits.	
2 <sup>*1</sup>	Set the LVCMPCR.EXVCCINP2 bit to 0 (VCC voltage) or set it to 1 (CMPA2 pin input voltage).	
3 <sup>*1</sup>	Set the LVD2CR0.LVD2RI bit to 0 (voltage monitoring 2 interrupt).	Set the LVD2CR0.LVD2RI bit to 1 (voltage monitoring 2 reset). Select the type of reset negation by setting the LVD2CR0.LVD2RN bit.
4	Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits. Select the type of interrupt by setting the LVD2CR1.LVD2IRQSEL bit.	—
5	—	Set the LVD2CR0.LVD2RIE bit to 1 (voltage monitoring 2 interrupt/reset enabled).
6 <sup>*1</sup>	Set the LVCMPCR.LVD2E bit to 1 (voltage detection 2 circuit enabled).	
7 <sup>*1</sup>	Wait for at least td(E-A).	
8	Set the LVD2CR0.LVD2CMPE bit to 1 (voltage monitoring 2 circuit comparison results output enabled).	
9	Set the LVD2SR.LVD2DET bit to 0.	—
10	Set the LVD2CR0.LVD2RIE bit to 1 (voltage monitoring 2 interrupt/reset enabled)	—

Note 1. Steps 1, 2, 3, 6, and 7 are not required if operation is with the setting to select the voltage monitoring 2 interrupt (LVD2CR0.LVD2RI = 0) and operation can be restarted by simply changing the settings of the LVD2CR1.LVD2IRQSEL and LVD2IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 2 reset (LVD2CR0.LVD2RI = 1), proceed through all steps from 1 to 10.

**Table 8.6 Procedures for Stopping Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset**

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

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

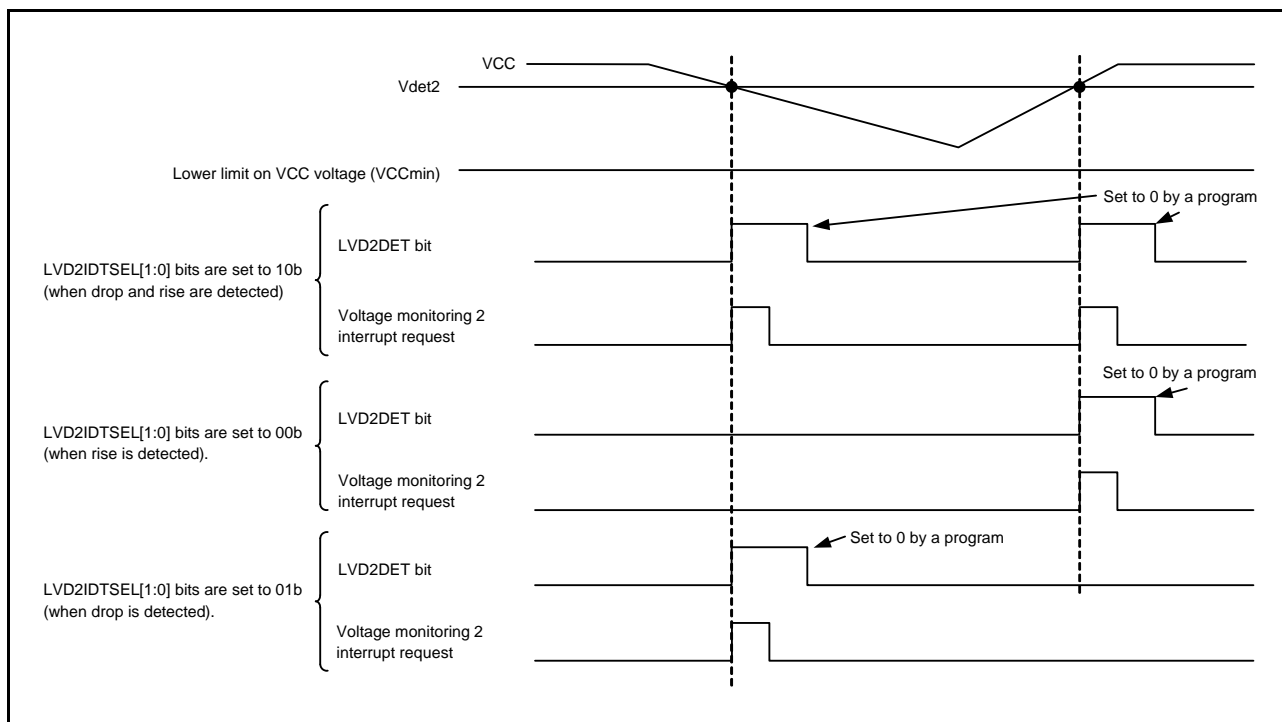


Figure 8.5 Example of Voltage Monitoring 2 Interrupt Operation

## 8.6 Event Link Output

The LVD can output the event signals to the event link controller (ELC).

### (1) Vdet1 passage detection event output

The LVD outputs the event signal when it is detected that the voltage has passed the Vdet1 voltage while both the voltage detection 1 circuit and the voltage monitoring 1 circuit comparison result output are enabled.

When enabling the LVD's event link output function, be sure to make settings for enabling the LVD before enabling the LVD event link function of the ELC. To stop the LVD's event link output function, be sure to make settings for stopping the LVD before disabling the LVD event link function of the ELC.

### 8.6.1 Interrupt Handling and Event Linking

The LVD has the bits to separately enable or disable the voltage monitoring 1 and 2 interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt request signal is output to the CPU.

On the contrary, as soon as an interrupt source is generated, the event link signal is output as the event signal to the other module via the ELC regardless of the state of the interrupt enable bit.

It is possible to output voltage monitoring 1 and 2 interrupts in software standby. The event signals for the ELC, however, are output as follows:

- When the events of passing Vdet1 are detected in software standby mode, no event signals are generated for the ELC because no clock is presented in software standby mode. Since the Vdet1 passage detection flags are preserved, however, when the supply of the clock is resumed after restoring from software standby mode, the event signals for the ELC are output according to the state of the Vdet1 passage detection flags.

## 9. Clock Generation Circuit

### 9.1 Overview

This MCU incorporates a clock generation circuit.

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

**Table 9.1 Specifications of Clock Generation Circuit**

Item	Specification
Uses	<ul style="list-style-type: none"> <li>Generates the system clock (ICLK) to be supplied to the CPU, DTC, ROM, and RAM.</li> <li>Generates the peripheral module clocks (PCLKB and PCLKD) to be supplied to peripheral modules. The peripheral module clock used as the operating clock is PCLKD for S12AD and PCLKB for other modules.</li> <li>Generates the FlashIF clock (FCLK) to be supplied to the FlashIF.</li> <li>Generates the USB clock (UCLK) to be supplied to the USB.</li> <li>Generates the CAC clock (CACCLK) to be supplied to the CAC.</li> <li>Generates the RTC-dedicated sub-clock (RTCSCCLK) to be supplied to the RTC.</li> <li>Generates the IWDT-dedicated low-speed clock (IWDTCLK) to be supplied to the IWDT.</li> <li>Generates the LCD source clock (LCDSRCCLK) to be supplied to the LCD.</li> <li>Generates the SSI clock (SSISCK) to be supplied to the SSI.</li> </ul>
Operating frequencies*1	<ul style="list-style-type: none"> <li>ICLK: 32 MHz (max)*2</li> <li>PCLKB: 32 MHz (max)*2</li> <li>PCLKD: 32 MHz (max)*2</li> <li>FCLK: 1 to 32 MHz (for programming and erasing the ROM and E2 DataFlash) 32 MHz (max) (for reading from the E2 DataFlash)</li> <li>UCLK: 48 MHz</li> <li>CACCLK: Same frequency as each oscillator</li> <li>RTCSCCLK: 32.768 kHz</li> <li>IWDTCLK: 15 kHz</li> <li>LCDSRCCLK: Same frequency as each oscillator</li> </ul>
Main clock oscillator*3	<ul style="list-style-type: none"> <li>Resonator frequency: 1 to 20 MHz (VCC ≥ 2.4 V), 1 to 8 MHz (VCC &lt; 2.4 V)</li> <li>External clock input frequency: 20 MHz (max)</li> <li>Connectable resonator or additional circuit: ceramic resonator, crystal</li> <li>Connection pins: EXTAL, XTAL</li> <li>Oscillation stop detection function: When a main clock oscillation stop is detected, the system clock source is switched to LOCO and MTU output can be forcedly driven to 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</li> <li>Connection pin: XCIN, XCOUT</li> </ul>
PLL circuit*4	<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 to 8 MHz</li> <li>Frequency multiplication ratio: Selectable from 6 and 8</li> <li>VCO oscillation frequency: 32 to 48 MHz (VCC ≥ 2.4 V)</li> </ul>
USB-dedicated PLL circuit*4	<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: 6 and 8 MHz</li> <li>Frequency multiplication ratio: Selectable from 6 and 8</li> <li>VCO oscillation frequency: 48 MHz (VCC ≥ 2.4 V)</li> </ul>
High-speed on-chip oscillator (HOCO)	Oscillation frequency: 32 MHz
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 4 MHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 15 kHz

Note 1. The maximum operating frequency in high-speed operating mode. For the maximum operating frequency in the other operating modes, refer to section 11.2.6, Operating Power Control Register (OPCCR).

Note 2. The relationship of frequencies must be set as follows. ICLK: FCLK, PCLKB, and PCLKD = N: 1 (N is an integer)

Note 3. When oscillating the USB-dedicated PLL at 48 MHz and the PLL at 32 MHz, the frequency of the main clock oscillator is limited to 8 or 16 MHz.

Note 4. The PLL and USB-dedicated PLL can be used when the external voltage (VCC) is 2.4 V or above.

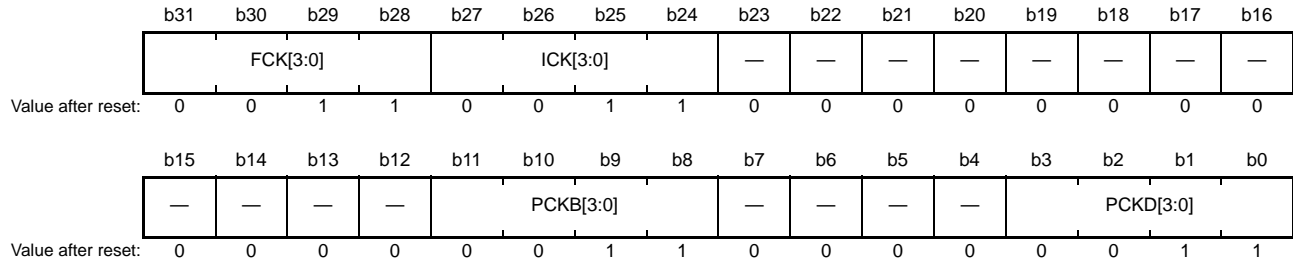




## 9.2 Register Descriptions

### 9.2.1 System Clock Control Register (SCKCR)

Address(es): 0008 0020h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PCKD[3:0]	Peripheral Module Clock D (PCLKD) Select*1, *2	b3 b0 0 0 0 0: x1 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 those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	PCKB[3:0]	Peripheral Module Clock B (PCLKB) Select*1, *2	b11 b8 0 0 0 0: x1 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
b23 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	ICK[3:0]	System Clock (ICLK) Select*1, *2	b27 b24 0 0 0 0: x1 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, *2	b31 b28 0 0 0 0: x1 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: • Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note 1. The setting for division by one is prohibited if the PLL is selected and the PLL oscillation frequency is 48 MHz.

Note 2. The relationship of frequencies must be set as follows. ICLK: FCLK, PCLKB, and PCLKD = 1: N (N is an integer).

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

When an instruction for writing to SCKCR or SCKCR3 is to follow writing to the SCKCR register, do so in accord with the procedure below.

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

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

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

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

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

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

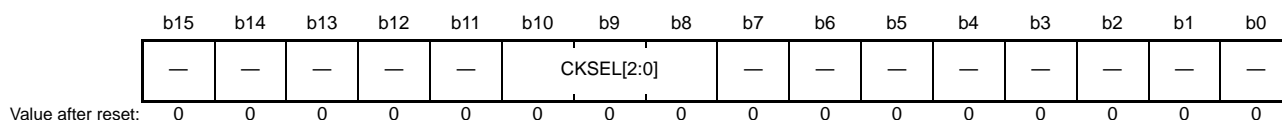
These bits select the frequency of the system clock (ICLK).

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

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

## 9.2.2 System Clock Control Register 3 (SCKCR3)

Address(es): 0008 0026h



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

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

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

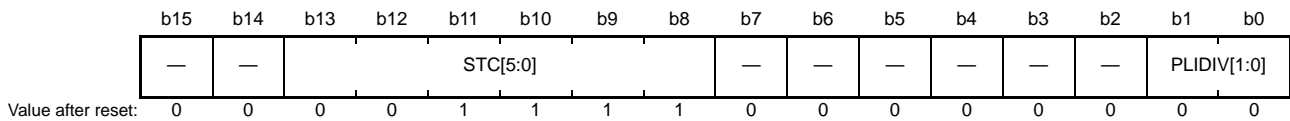
### CKSEL[2:0] Bits (Clock Source Select)

These bits select the source of the system clock (ICLK), peripheral module clock (PCLKB and PCLKD), FlashIF clock (FCLK), 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.

### 9.2.3 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 1 0 1 1: x6 0 0 1 1 1 1: x8 Settings other than those listed above are prohibited.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

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

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

These bits select the frequency division ratio of the PLL clock source.

Set these bits so that the frequency of PLL input signal is within the range of 4 MHz to 8 MHz.

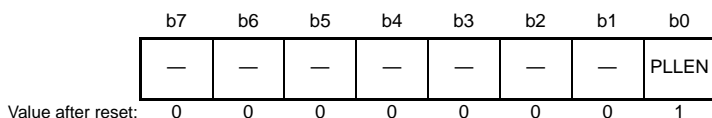
#### STC[5:0] Bits (Frequency Multiplication Factor Select)

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

Set these bits so that the PLL oscillation frequency is within the range of 32 MHz to 48 MHz.

### 9.2.4 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

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

#### PLLEN Bit (PLL Stop Control)

This bit runs or stops the PLL circuit.

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

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

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

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

When the external voltage (VCC) is below 2.4 V or low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the PLLEN bit to 0 (PLL is operating).

### 9.2.5 USB-dedicated PLL Control Register (UPLLCR)

Address(es): 0008 002Ch

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	USTC[5:0]					—	—	—	UCKUPLLSEL	—	—	UPLIDIV[1:0]		
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b1, b0	UPLIDIV[1:0]	USB-dedicated PLL Input Frequency Division Ratio Select	b1 b0 0 0: x1 0 1: x1/2 Settings other than above are prohibited.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	UCKUPLLSEL	UCLK Source USB-Dedicated PLL Select	0: System clock is selected as UCLK 1: USB-dedicated PLL is selected as UCLK	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	USTC[5:0]	Frequency Multiplication Factor Select	b13 b8 0 0 1 0 1 1: x6 0 0 1 1 1 1: x8 Settings other than above are prohibited.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

Writing to the UPLLCR is prohibited when the UPLLCR2.UPLLEN bit is 0 (USB-dedicated PLL is operating).

#### UPLIDIV[1:0] Bits (USB-dedicated PLL Input Frequency Division Ratio Select)

These bits select the input frequency division ratio of the USB-dedicated PLL clock source. Set these bits so that the input frequency of the USB-dedicated PLL input is 6 or 8 MHz.

#### UCKUPLLSEL Bit (UCLK Source USB-Dedicated PLL Select)

This bit selects the USB-dedicated PLL clock. When this bit is set to 1, the USB-dedicated PLL (48 MHz) is supplied to the UCLK, and the clock of the oscillator (up to 32 MHz) selected by the SCKCR3.CKSEL[2:0] bits is supplied to the system clock.

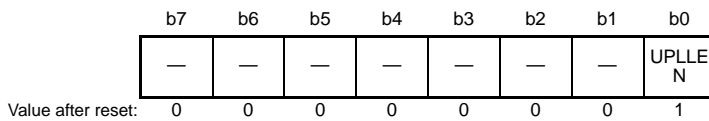
When setting this bit to change the UCLK source clock, set the MSTPCRB.MSTPB19 bit (this module clock is disabled) and wait for at least three cycles of the clock before the change.

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

These bits select the frequency multiplication factor of the USB-dedicated PLL circuit. Set these bits so that the USB-dedicated PLL oscillation frequency is 48 MHz.

## 9.2.6 USB-dedicated PLL Control Register 2 (UPLLCR2)

Address(es): 0008 002Eh



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

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

The UPLLCR2 register runs or stops the USB-dedicated PLL circuit.

### UPLLEN Bit (USB-dedicated PLL Stop Control)

This bit runs or stops the USB-dedicated PLL circuit.

After setting the UPLLEN bit to 0 (USB-dedicated PLL is operating), read the OSCOVFSR.UPLOVF flag to confirm that it has become 1, and then start using the USB-dedicated PLL clock.

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

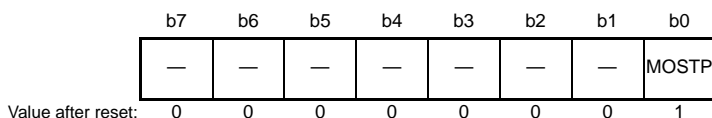
- After stopping the USB-dedicated PLL, confirm that the OSCOVFSR.UPLOVF bit is 0 before restarting the USB-dedicated PLL.
- Confirm that the USB-dedicated PLL is operating and that the OSCOVFSR.UPLOVF bit is 1 before stopping the USB-dedicated PLL.
- Regardless of whether or not it is selected as the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

When the external voltage (VCC) is below 2.4 V or low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the UPLLEN bit to 0 (USB-dedicated PLL is operating).

When the USB-dedicated PLL is selected by the UPLLCR.UCKUPLLSEL bit and the MSTPCRB.MSTPB19 bit is set to 0 (this module clock is enabled), do not set the UPLLEN bit to 1 (USB-dedicated PLL is stopped).

## 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

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

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

### MOSTP Bit (Main Clock Oscillator Stop)

This bit runs or stops the main clock oscillator.

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

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

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

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

- When the main clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 010b)
- When the PLL clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 100b)
- When the PLL is operating (PLLCR2.PPLEN = 0)
- When the USB-dedicated PLL is operating (UPLLCR2.PPLEN = 0)

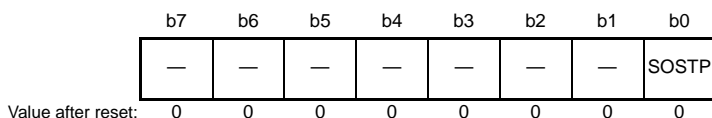
Do not set the MOSTP bit to 0 when the following condition is met.

- When low-speed operating mode is selected by the SOPCCR.SOPCM bit



## 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

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

The XCIN pin is also used as ports. In their initialized state, they function as pins for the sub-clock oscillator.

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

This bit runs or stops the sub-clock oscillator.

The SOSTP bit and the sub-clock oscillator control bit in 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).

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 time ( $t_{SUBOSC}$ ) has elapsed.

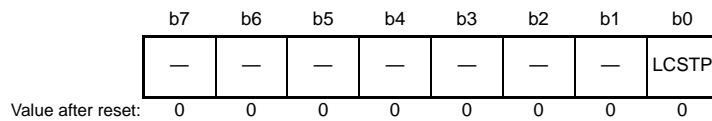
That is, a fixed time for stabilization is required after the setting for sub-clock oscillator operation. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- 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.
- When a transition to software standby mode 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.

While the sub-clock oscillator is selected by the SCKCR3.CKSEL[2:0] bits, do not set the SOSTP bit to 1 (sub-clock oscillator is stopped).

### 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

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

#### LCSTP Bit (LOCO Stop)

This bit runs or stops the LOCO.

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

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

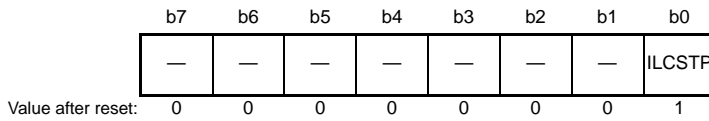
- When restarting the LOCO after it has been stopped, allow at least five cycles of the LOCO as an interval over which it is still stopped.
- Ensure that oscillation by the LOCO is stable when making the setting to stop the LOCO.
- Regardless of whether or not it is selected as the system clock, ensure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

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

While low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the LCSTP bit to 0 (LOCO is operating).

### 9.2.10 IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR)

Address(es): 0008 0035h



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

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

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

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

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

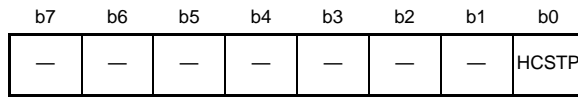
After the setting of the ILCSTP bit has been changed so that the IWDT-dedicated on-chip oscillator operates, supply of the clock is started the MCU internally after a fixed time corresponding to the IWDT-dedicated clock oscillation stabilization time ( $t_{ILOCO}$ ) has elapsed.

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

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

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

Address(es): 0008 0036h



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

Note 1. The HCSTP bit value after a reset is 0 when the HOCO oscillation enable bit in option function select register 1 (OFS1.HOCOEN) is 0. The HCSTP bit value after a reset is 1 when the OFS1.HOCOEN bit is 1.

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

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

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

#### HCSTP Bit (HOCO Stop)

This bit runs or stops the HOCO.

When changing the HCSTP bit from 1 to 0 (i.e. changing the HOCO clock from stopped to operating), confirm that the OSCOVFSR.HCOVF bit is 1 before switching the system clock to the HOCO clock.

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

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

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

While low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the HCSTP bit to 0 (HOCO is operating).

## 9.2.12 Oscillation Stabilization Flag Register (OSCOVFSR)

Address(es): 0008 003Ch

b7	b6	b5	b4	b3	b2	b1	b0
—	—	UPLOV F	—	HCOVF	PLOVF	—	MOOV F

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

Note 1. The HCOVF value after a reset is 1 when the HOCO oscillation enable bit in option function selection register 1 (OFS1.HOCOEN) is 0. The HCSTP value after a reset is 1 when the OFS1.HOCOEN bit is 0.

Bit	Symbol	Bit Name	Description	R/W
b0	MOOVF	Main Clock Oscillation Stabilization Flag	0: Main clock is stopped 1: Oscillation is stable and the clock can be used as the system clock*1	R
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	PLOVF	PLL Clock Oscillation Stabilization Flag	0: PLL is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the system clock	R
b3	HCOVF	HOCO Clock Oscillation Stabilization Flag	0: HOCO is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the system clock*1	R
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	UPLOVF	USB-Dedicated PLL Clock Oscillation Stabilization Flag	0: USB-dedicated PLL is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the UCLK	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When an appropriate value is set in the wait control register for each oscillator. If a set value (wait time) is not adequate, clock supply starts before oscillation becomes stable.

The OSCOVFSR register monitors whether oscillation of each oscillator has become stable.

If a wait control register is provided for each oscillator, specify a wait time that is longer than or equal to the stabilization time of the corresponding oscillation circuit.

### MOOVF Flag (Main Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the main clock is stable.

[Setting condition]

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

[Clearing condition]

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

### PLOVF Flag (PLL Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the PLL clock is stable.

[Setting condition]

After the PLLCR2.PLEN is set to 0 (PLL is operating) when the PLEN bit is 1 (PLL is stopped), the MOOVF flag becomes 1, the PLL clock oscillation stabilization time (tPLL) has elapsed, and supply of the PLL clock is started to the MCU internally.

[Clearing condition]

After the PLLCR2.PLEN bit is set to 1, the processing to stop the oscillation of the PLL is completed.

**HCOVF Flag (HOCO Clock Oscillation Stabilization Flag)**

This flag indicates whether oscillation of the HOCO clock is stable.

[Setting condition]

- After the HOCOCCR.HCSTP bit is set to 0 (HOCO is operating) when the HCSTP bit is 1 (HOCO is stopped), the corresponding time set in the register has elapsed and supply of the HOCO clock is started to the MCU internally.

[Clearing condition]

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

**UPLOVF Flag (USB-Dedicated PLL Clock Oscillation Stabilization Flag)**

This flag indicates whether oscillation of the USB-dedicated PLL clock is stable.

[Setting condition]

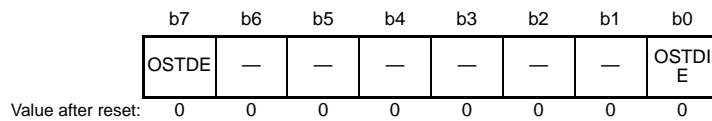
- After the UPLLCR2.UPLLEN bit is set to 0 (USB-dedicated PLL is operating) when the UPLLEN bit is 1 (USB-dedicated PLL is stopped), the MOOVF flag becomes 1 and the PLL clock stabilization wait time (tPLL) has elapsed, and then supply of the USB-dedicated PLL clock is started to the MCU internally.

[Clearing condition]

- After the UPLLCR2.UPLLEN bit is set to 1, the processing to stop the oscillation of the USB-dedicated PLL clock is completed.

### 9.2.13 Oscillation Stop Detection Control Register (OSTDCR)

Address(es): 0008 0040h



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

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

#### OSTDIE Bit (Oscillation Stop Detection Interrupt Enable)

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

#### OSTDE Bit (Oscillation Stop Detection Function Enable)

This bit enables or disables the oscillation stop detection function.

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

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

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

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

Note 1. This bit can only be set to 0.

#### OSTDF Flag (Oscillation Stop Detection Flag)

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

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

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

[Setting condition]

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

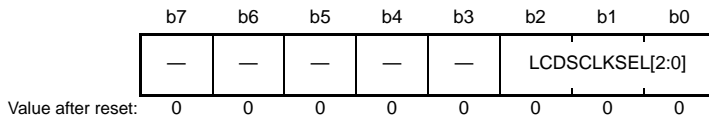
[Clearing condition]

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



## 9.2.15 LCD Source Clock Control Register (LCDSCLKCR)

Address(es): 0008 0050h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	LCDSCLKSEL[2:0]	LCD Source Clock Select	b2 b0 0 0 0: LOCO 0 0 1: HOCO 0 1 0: Main clock oscillator 0 1 1: Sub-clock oscillator 1 0 0: IWDT-dedicated on-chip oscillator Settings other than above are prohibited.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

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

These bits select the LCD source clock (LCDSRCCLK) from the LOCO, HOCO, main clock oscillator, sub-clock oscillator, and IWDT-dedicated on-chip oscillator.

When changing the selected clock, set the LCDSCLKCR2.LCDCLKSTP bit to 1 (LCD source clock output is stopped) and wait for three cycles of the clock before the change and two cycles of the system clock. After changing the clock, read the register to confirm that the selected clock has been changed.

If the MCU enters software standby mode when the LCDSCLKCR2.LCDCLKSTP bit is 0 (LCD source clock output is not stopped), operation of the LCD source clock (LCDSRCCLK) differs depending on the selected clock.

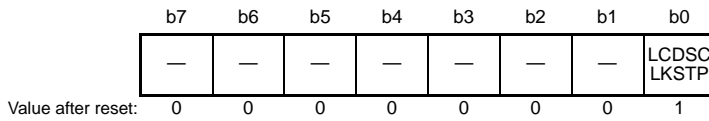
When the LOCO, HOCO, or main clock oscillator is selected, the LCD source clock stops.

When the sub-clock oscillator is selected, whether to operate or stop the oscillator can be selected by setting the RCR3.RTCEN and SOSCCR.SOSTP bits.

When the IWDT-dedicated on-chip oscillator is selected, whether to operate or stop the oscillator can be selected by setting the OFS0.SLCSTP bit while IWDT auto-start mode is selected. While IWDT register start mode is selected, whether to operate or stop the oscillator can be selected by setting the OFS0.SLCSTP bit.

## 9.2.16 LCD Source Clock Control Register 2 (LCDSCLKCR2)

Address(es): 0008 0051h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	LCDSCLKSTP	LCD Source Clock Output Stop Control	0: LCD source clock output is not stopped 1: LCD source clock output is stopped	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

The LCDSCLKCR2 register controls the LCD source clock (LCDSRCCLK).

### LCDSCLKSTP Bit (LCD Source Clock Output Stop Control)

This bit stops or outputs the LCDSRCCLK.

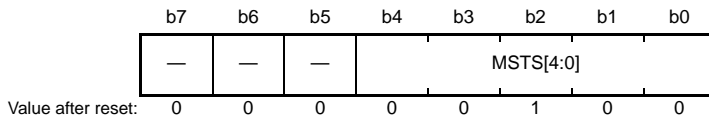
This bit should be rewritten when the oscillation of the clock selected by the LCDSCLKCR.LCDSCLKSEL[2:0] bits is stable.

When entering software standby mode after rewriting this bit, wait for two cycles of the selected clock after rewriting, and execute a WAIT instruction.

When stopping the selected clock after rewriting this bit, wait for two cycles of the selected clock after rewriting, and stop the clock.

### 9.2.17 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	b4      b0 0 0 0 0 0: Wait time = 2 cycles (0.5 $\mu$ s) 0 0 0 0 1: Wait time = 1024 cycles (256 $\mu$ s) 0 0 0 1 0: Wait time = 2048 cycles (512 $\mu$ s) 0 0 0 1 1: Wait time = 4096 cycles (1.024 ms) 0 0 1 0 0: Wait time = 8192 cycles (2.048 ms) 0 0 1 0 1: Wait time = 16384 cycles (4.096 ms) 0 0 1 1 0: Wait time = 32768 cycles (8.192 ms) 0 0 1 1 1: Wait time = 65536 cycles (16.384 ms) Settings other than above are prohibited. Wait time when LOCO = 4.0 MHz (0.25 $\mu$ s, TYP.)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

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

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

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

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

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

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

### 9.2.18 High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR)

Address(es): 0008 00A5h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	HSTS[4:0]	High-Speed On-Chip Oscillator Wait Time	b4      b0 0 0 1 0 1: Wait time = 138 cycles (34.5 $\mu$ s)*1, *2, *4 0 0 1 1 0: Wait time = 266 cycles (66.5 $\mu$ s)*3, *4 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: • Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.

Note 1. If this value is set, the HOCO oscillation stabilization time ( $t_{\text{HOCO}}$ ) is not secured, so the HOCO frequency accuracy shown in Electrical Characteristics is not guaranteed when supply of the clock starts. When  $t_{\text{HOCO}}$  has elapsed after oscillation starts, the HOCO frequency accuracy is as shown in Electrical Characteristics.

Note 2. When the OFS1.HOCOEN bit is set to 0, the HOCO oscillation stabilization time ( $t_{\text{HOCO}}$ ) is secured by hardware, so the clock with the accuracy of the HOCO frequency ( $f_{\text{HOCO}}$ ) shown in Electrical Characteristics is supplied after release from the CPU reset state.

Note 3. When this value is set, the HOCO oscillation stabilization time ( $t_{\text{HOCO}}$ ) is secured and the clock with the accuracy of the HOCO frequency ( $f_{\text{HOCO}}$ ) shown in Electrical Characteristics is supplied after release from the CPU reset state.

Note 4. Wait time when LOCO = 4.0 MHz (0.25  $\mu$ s, TYP.)

#### HSTS[4:0] Bits (High-Speed On-Chip Oscillator Wait Time)

These bits are used to select the oscillation stabilization wait time of the HOCO when setting HOCO operation (the HOCOCR.HCSTP bit to 0) and when canceling software standby mode.

Supply of the HOCO clock is started to the MCU internally after the number of LOCO cycles set by the HSTS[4:0] bits has been counted. Counting of LOCO cycles proceeds regardless of the setting of the LOCOCR.LOSTP bit and hardware automatically controls running and stopping the LOCO.

The clock is not supplied to the MCU internally until counting is completed.

After counting is completed, supply of the clock is started to the MCU internally and the OSCOVFSR.HCOVF flag is set to 1.

The HOCOWTCR register can be rewritten under the following cases. Otherwise, do not rewrite this register.

- When the HOCOCR.HCSTP bit is set to 0 (operating), and the OSCOVFSR.HCOVF flag is read and confirmed to be 1.
- When the HOCOCR.HCSTP bit is set to 1 (stopped), and the OSCOVFSR.HCOVF flag is read and confirmed to be 0.

## 9.2.19 CLKOUT Output Control Register (CKOCR)

Address(es): 0008 003Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CKOSTP	CKODIV[2:0]			—	CKOSEL[2:0]			—	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CKOSEL[2:0]	CLKOUT Output Source Select	b10 b8 0 0 0: LOCO clock 0 0 1: HOCO clock 0 1 0: Main clock oscillator 0 1 1: Sub-clock oscillator Settings other than above are prohibited.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14 to b12	CKODIV[2:0]	CLKOUT Output Division Ratio Select	b14 b2 0 0 0: No division 0 0 1: x1/2 0 1 0: x1/4 0 1 1: x1/8 1 0 0: x1/16 Settings other than above are prohibited.	R/W
b15	CKOSTP	CLKOUT Output Stop Control	0: CLKOUT pin output enabled*1 1: CLKOUT pin output disabled	R/W

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

Note 1. It is also necessary to set the pin function control register and port mode register for the corresponding pin.

### CKOSEL[2:0] Bits (CLKOUT Output Source Select)

Set these bits to select the LOCO clock, HOCO clock, main clock, or sub-clock as the source of the clock to be output from the CLKOUT pin.

### CKODIV[2:0] Bits (CLKOUT Output Division Ratio Select)

Set these bits to select the clock division ratio.

Set the CKOSTP bit to 1 when changing the division ratio.

The division ratio of the output clock frequency should be set to no higher than 8 MHz when VCC is 2.7 V or above, and no higher than 4 MHz when VCC is below 2.7 V.

For details on the characteristics of the clock output from the CLKOUT pin, see Table 36.32, Timing of On-Chip Peripheral Modules (1).

### CKOSTP Bit (CLKOUT Output Stop Control)

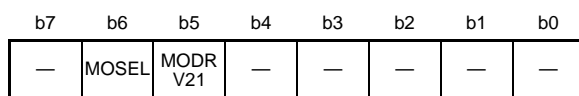
Set this bit to enable or disable output from the CLKOUT pin.

When this bit is set to 1, the selected clock is output. When this bit is set to 0, a low level is output.

If the CKOSTP bit is rewritten while the clock is still oscillating, a glitch may be generated in the output.

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

Address(es): 0008 C293h



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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	MODRV21	Main Clock Oscillator Drive Capability Switch	VCC ≥ 2.4 V 0: 1 MHz to 10 MHz 1: 10 MHz to 20 MHz VCC < 2.4 V 0: 1 MHz to 8 MHz 1: Setting prohibited	R/W
b6	MOSEL	Main Clock Oscillator Switch	0: Resonator 1: External oscillator input	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

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

### MODRV21 Bit (Main Clock Oscillator Drive Capability Switch)

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

### MOSEL Bit (Main Clock Oscillator Switch)

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

### 9.3 Main Clock Oscillator

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

#### 9.3.1 Connecting a Crystal

Figure 9.2 shows an example of connecting a crystal.

A damping resistor ( $R_d$ ) should be added, if necessary. Since the resistor values vary depending on the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If use of an external feedback resistor ( $R_f$ ) is directed by the resonator manufacturer, insert an  $R_f$  between EXTAL and XTAL by following the instruction.

When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency range of the resonator for the main clock oscillator described in Table 9.1.

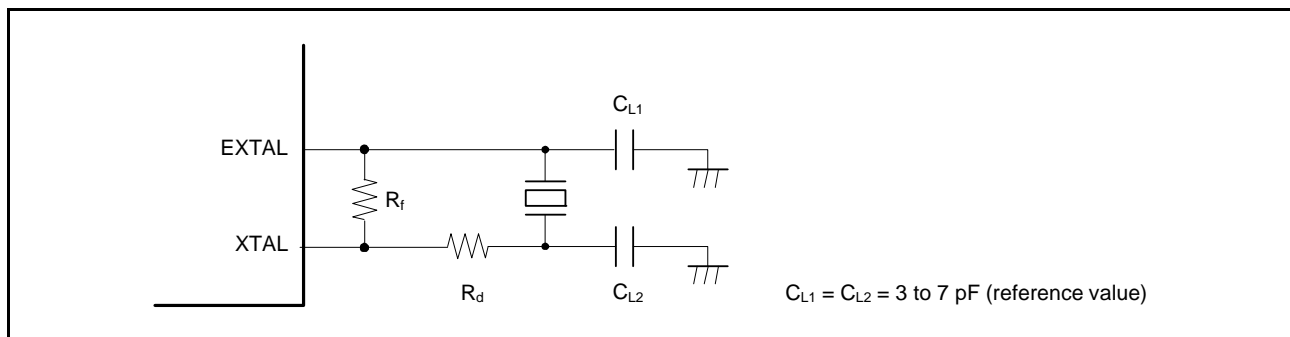


Figure 9.2 Example of Crystal Connection

Table 9.3 Damping Resistance (Reference Values)

Frequency (MHz)	2	8	16	20
$R_d$ ( $\Omega$ )	0	0	0	0

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

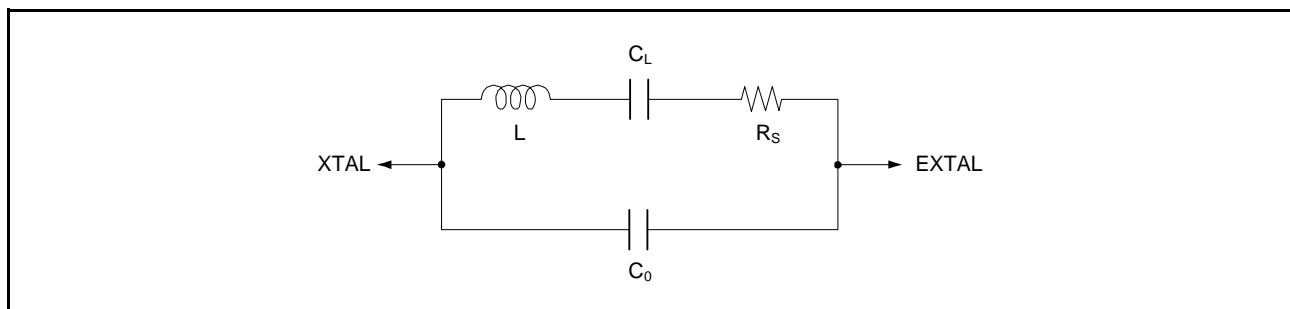


Figure 9.3 Equivalent Circuit of Crystal

Table 9.4 Crystal Characteristics (Reference Values)

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

### 9.3.2 External Clock Input

Figure 9.4 shows connection of an external clock. Set the MOFCR.MOSEL bit to 1 if operation is to be driven by an external clock. In this case, the EXTAL pin will be in the Hi-Z state.

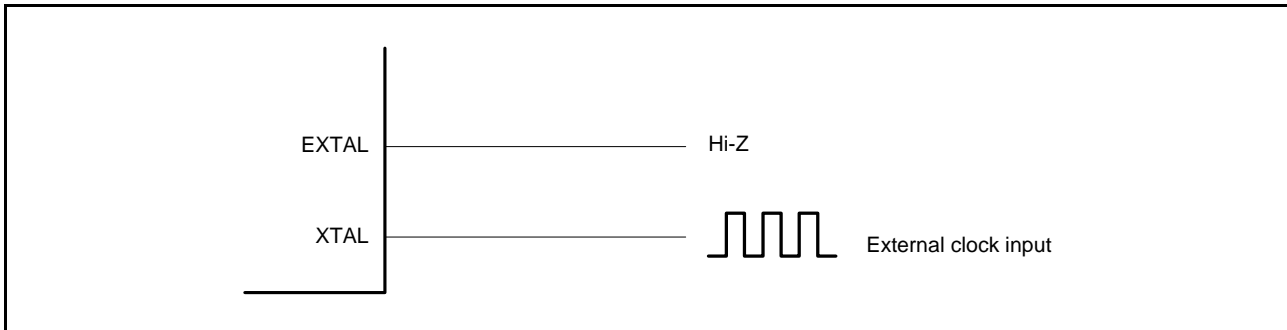


Figure 9.4 Connection Example of External Clock

### 9.3.3 Handling of Pins When the Main Clock is Not Used

For details on pin handling when the main clock is not used, refer to section 18.5, Handling of Unused Pins.

### 9.3.4 Notes on the External Clock Input

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



## 9.4 Sub-Clock Oscillator

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

### 9.4.1 Connecting 32.768-kHz Crystal

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal, 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 XCOUT 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.

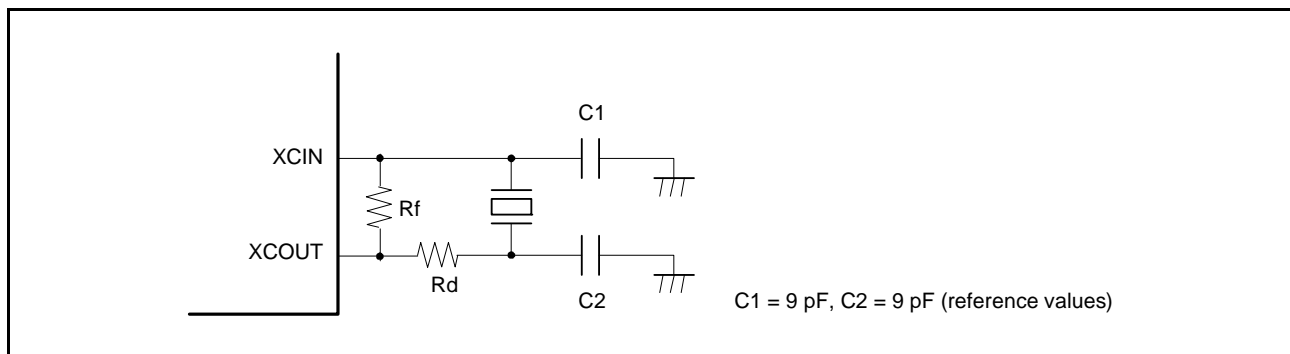


Figure 9.5 Connection Example of 32.768-kHz Crystal

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

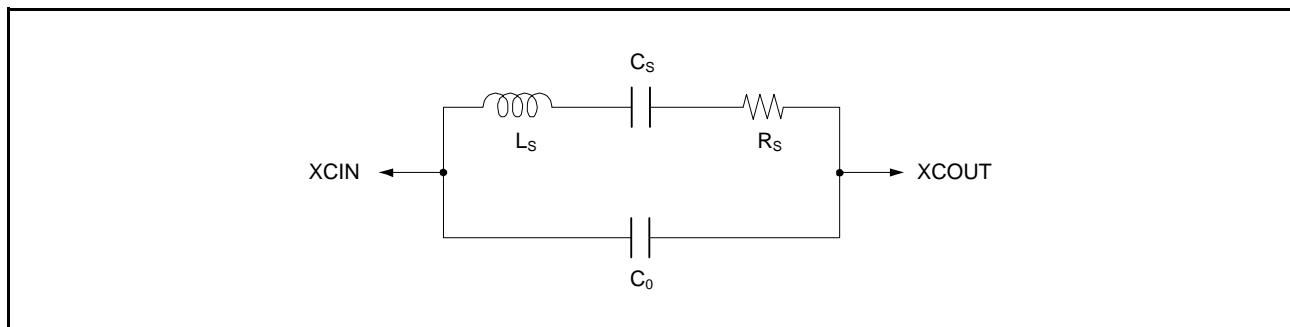


Figure 9.6 Equivalent Circuit for Crystal

Table 9.5 Crystal Characteristics (Reference Values)

Frequency (kHz)	32.768 (Low CL)
$R_s$ max (k $\Omega$ )	37
$C_0$ max (pF)	0.9

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

When the sub-clock is not used, set the SOSCCR.SOSTP bit to 1 (stopped) and set the RCR3.RTCEN bit to 0 (sub-clock oscillator is stopped) (set general port PH7). When this pin is not also used as port PH7, handle it as an unused pin. For handling of unused pins, refer to section 18.5, Handling of Unused Pins.

## 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.

An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the MTU output can be forcedly driven to the high-impedance on the detection. For details, refer to section 20, Multi-Function Timer Pulse Unit 2 (MTU2a) and section 21, Port Output Enable 2 (POE2a).

In the MCU, the main clock oscillation stop is detected when the input clock remains to be 0 or 1 for a certain period, for example, due to a malfunction of the main clock oscillator (refer to Figure 36.47, Oscillation Stop Detection Circuit Characteristics, in section 42, Electrical Characteristics).

When an oscillation stop is detected, the main clock selected by the clock source select bits (SCKCR3.CKSEL[2:0]) is switched to the LOCO clock by the corresponding selectors in the former stage. Therefore, if an oscillation stop is detected with the main clock selected as the system clock source, the system clock source is switched to the LOCO clock without a change of CKSEL[2:0].

If an oscillation stop is detected while the PLL clock is selected by the clock source select bits (SCKCR3.CKSEL[2:0]) in system clock control register 3, the SCKCR3.CKSEL[2:0] bit value does not change and the PLL clock remains the system clock source. However, the frequency becomes a free-running oscillation frequency.

Also, if an oscillation stop is detected while the USB-dedicated PLL clock is selected by the UPLLCR.UCKPLLSEL bit, the set value of these bits does not change and the USB-dedicated PLL clock remains the UCLK clock source. However, the frequency becomes a free-running oscillation frequency.

Switching between the main clock and LOCO clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF). The clock source is switched to the LOCO clock when the OSTDF flag is 1, and is switched to the main clock again when the OSTDF flag is set to 0. At this time, if the main clock or PLL clock is selected with the CKSEL[2:0] bits, the OSTDF flag cannot be set to 0. To switch the clock source to the main clock or PLL clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL clock and set the OSTDF flag to 0. After that, check that the OSTDF flag is not 1, and then set the CKSEL[2:0] bits to the main clock or PLL clock after the specified oscillation stabilization time has elapsed.

After a reset is released, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after a specified oscillation stabilization time has elapsed.

The oscillation stop detection function is provided against the main clock stop by an external cause. Therefore, the oscillation stop detection function should be disabled before the main clock oscillator is stopped by the software or a transition is made to software standby mode.

When the main clock or CAC main clock (CACMCLK) is selected as the system clock source, the main clock is selected as the LCD source clock, and the SSI clock is selected, these clocks are switched to the LOCO clock by the oscillation stop detection. The system clock (ICLK) frequency during the LOCO clock operation is specified by the LOCO oscillation frequency and the division ratio set by the system clock (ICLK) select bits (SCKCR.ICK[3:0]).

When the PLL clock is selected as the system clock source and the USB-dedicated PLL clock is selected as the UCLK source, these clocks operate at the PLL free-running frequency by the oscillation stop detection.

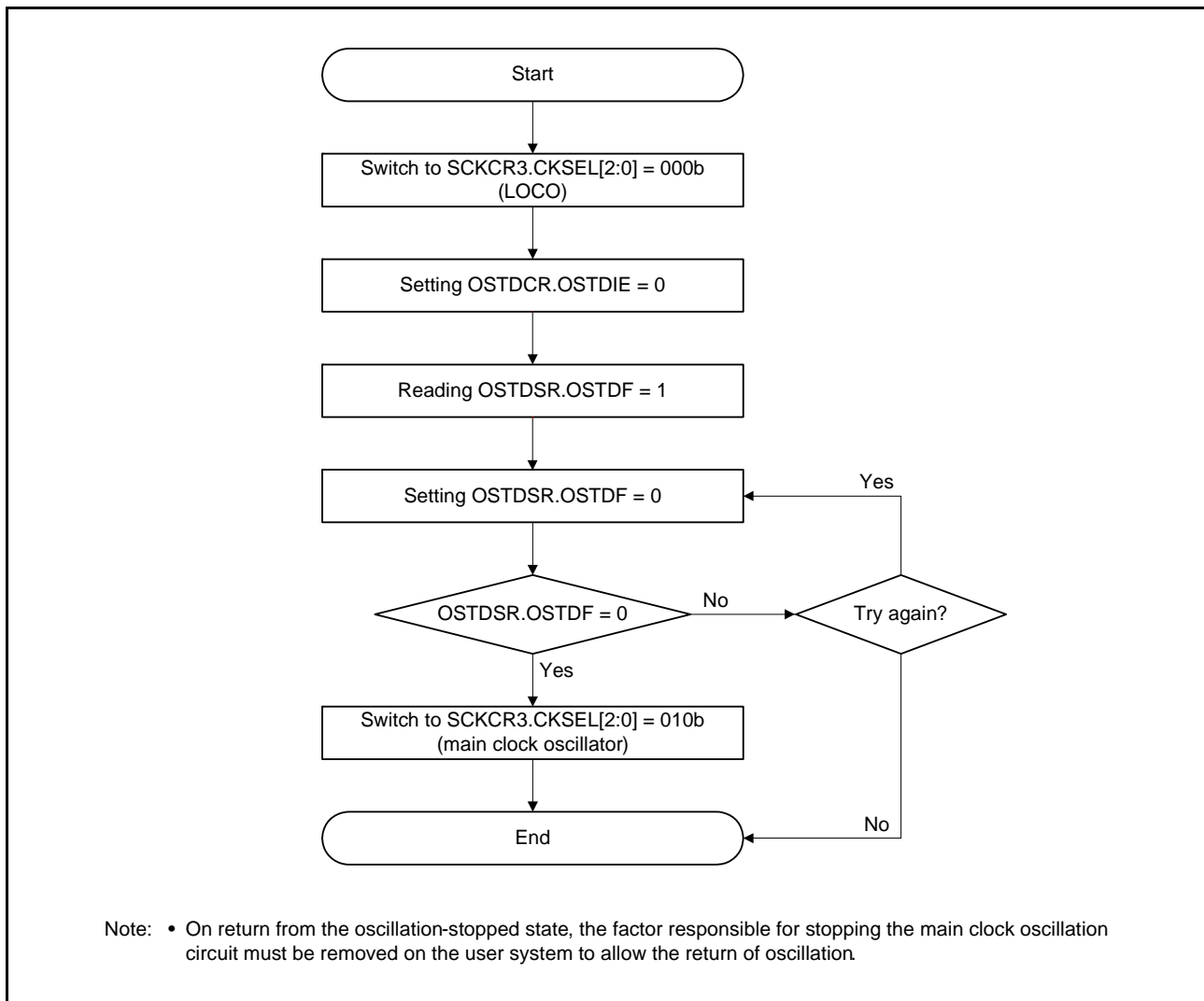


Figure 9.7 Flow of Recovery from Detection of Oscillator Stop

### 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 (OSTDCR.OSTDIE) is 1 (oscillation stop detection interrupt enabled). At this time, the main clock oscillator stop is notified to 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 setting the oscillation stop detection interrupt enable bit (OSTDCR.OSTDIE) to 0. Wait for at least two cycles of PCLKB clock before again setting the OSTDCR.OSTDIE bit to 1. According to the number of cycles for access to read a given I/O register, wait time longer than two cycles of PCLKB may have to be secured.

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

## 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, USB-dedicated PLL clock, and dedicated low-speed clock for the IWDT. The internal clocks listed below are produced from these sources.

- (1) Operating clock of the CPU, DTC, ROM, and RAM: System clock (ICLK)
- (2) Operating clock of peripheral modules: Peripheral module clock (PCLKB and PCLKD)
- (3) Operating clock of the FlashIF: FlashIF clock (FCLK)
- (4) Operating clock of USB modules: USB clock (UCLK)
- (5) Operating clock for the CAC: CAC clock (CACCLK)
- (6) Operating clock for the RTC: RTC-dedicated sub-clock (RTCSCCLK)
- (7) Operating clock for the IWDT: IWDT-dedicated low-speed clock (IWDTCLK)
- (8) Operating clock for the LCD: LCD source clock (LCDSRCCLK)
- (9) Operating clock for the SSI: SSI clock (SSISCK)

Frequencies of the internal clocks are set by the combination of the divisors selected by the SCKCR.FCK[3:0], ICK[3:0], PCKB[3:0], and PCKD[3:0] bits, the clock source selected by the SCKCR3.CKSEL[2:0] bits, and the bits that select the frequency of the PLL circuit (PLLCR.STC[5:0], PLIDIV[1:0] bits, and the UPLLCR.UPLIDIV[1:0] and USTC[5:0] bits). 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, DTC, ROM, and RAM.

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

### 9.7.2 Peripheral Module Clock

The peripheral module clocks (PCLKB and PCLKD) are the operating clocks for use by peripheral modules.

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

The peripheral module clock used as the operating clock is PCLKD for S12AD and PCLKB for other modules.

### 9.7.3 FlashIF Clock

The FlashIF clock (FCLK) is used as the operating clock of the FlashIF.

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

#### 9.7.4 USB Clock

The USB clock (UCLK) is an operating clock for the USB module. The UCLK frequency is specified by the SCKCR3.CKSEL[2:0] bits, and the PLLCR.STC[5:0], PLIDIV[1:0] bits, and the UPLLCR.UPLIDIV[1:0] and USTC[5:0] bits.

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.

#### 9.7.5 LCD Source Clock

The LCD source clock is an operating clock source for the LCD module.

The LCDSRCCLK clock is selected from among the main clock oscillator, sub-clock oscillator, high-speed or low-speed on-chip oscillator, and IWDT-dedicated oscillator.

#### 9.7.6 CAC Clock

The CAC clock (CACCLK) is an operating clock for the CAC module.

The CACCLK clocks include CACMCLK which is generated by the main clock oscillator, CACSCLK which is generated by the sub-clock oscillator, CACHCLK which is generated by the high-speed on-chip oscillator, CACLCLK which is generated by the low-speed on-chip oscillator, and CACILCLK which is generated by the IWDT-dedicated on-chip oscillator.

#### 9.7.7 RTC-Dedicated Clock

The RTC-dedicated clock (RTCSCCLK) is the operating clock for the RTC.

RTCSCCLK is generated by the sub-clock oscillator.

#### 9.7.8 IWDT-Dedicated Clock

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

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

#### 9.7.9 SSI Clock

The SSI clock (SSISCK) is the operating clock for the SSI.

SSISCK is generated by the main clock oscillator.

## 9.8 Usage Notes

### 9.8.1 Notes on Clock Generation Circuit

- (1) The frequencies of the system clock (ICLK), peripheral module clocks (PCLKB and PCLKD), and FlashIF clock (FCLK) supplied to each module change according to the settings of the SCKCR register. 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 and PCLKD. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.

- (2) The relationship of frequencies of the system clock (ICLK), peripheral module clocks B and D (PCLKB and PCLKD), and FlashIF clock (FCLK) must be set as follows.

ICLK: FCLK, PCLKB, and PCLKD = N: 1 (N is an integer)

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

### 9.8.2 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.

### 9.8.3 Notes on Board Design

When using a crystal, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible.

Other signal lines should be routed away from the oscillation circuit as shown in Figure 9.8 to prevent electromagnetic induction from interfering with correct oscillation.

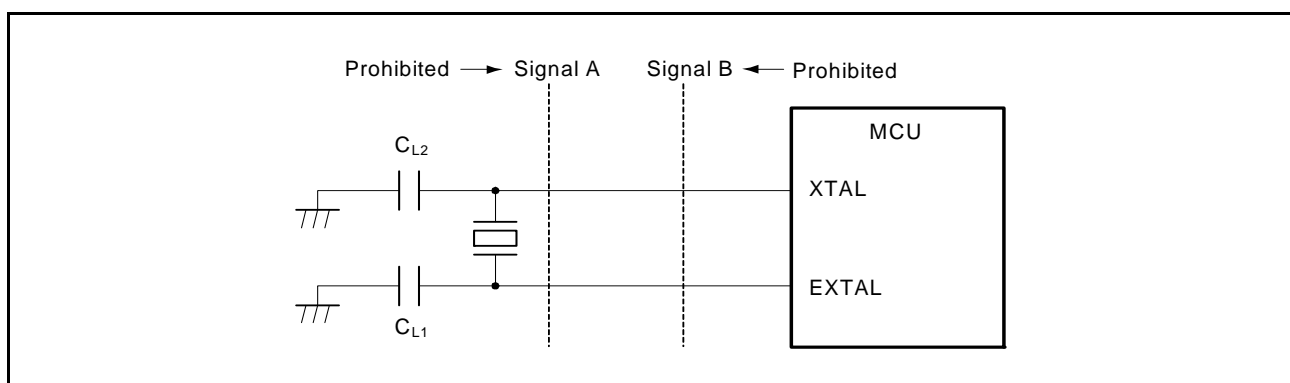


Figure 9.8 Notes on Board Design for Oscillation Circuit (Applies to the Sub-Clock Oscillator, in Case of the Main Clock Oscillator)

### 9.8.4 Notes on Sub-Clock

The sub-clock can be used as the system clock, as the count source for the realtime clock, or as both. Take note of the following limitations and points for caution regarding the settings, including when the sub-clock is not in use.

- With regard to making the sub-clock oscillator run or stop, setting either the sub-clock oscillator stop bit in the sub-clock oscillator control register (SOSCCR.SOSTP) or the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) will make the oscillator run.
- To use the sub-clock as the system clock and as the count source of the realtime clock simultaneously, perform initial settings according to the flowchart example shown in Figure 9.9. After that, perform the clock setting procedure shown in section 24.3.2, Clock and Count Mode Setting Procedure.

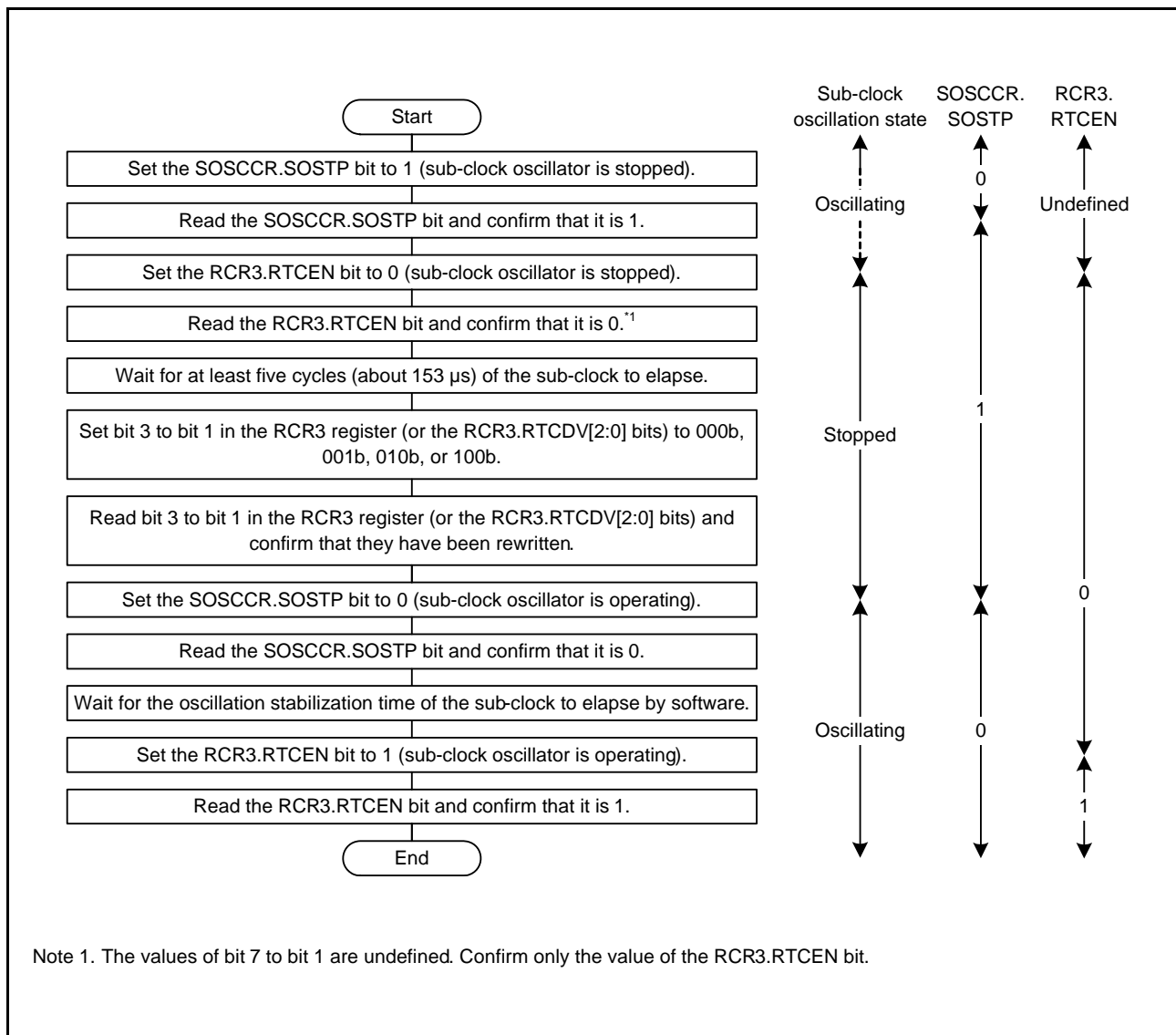
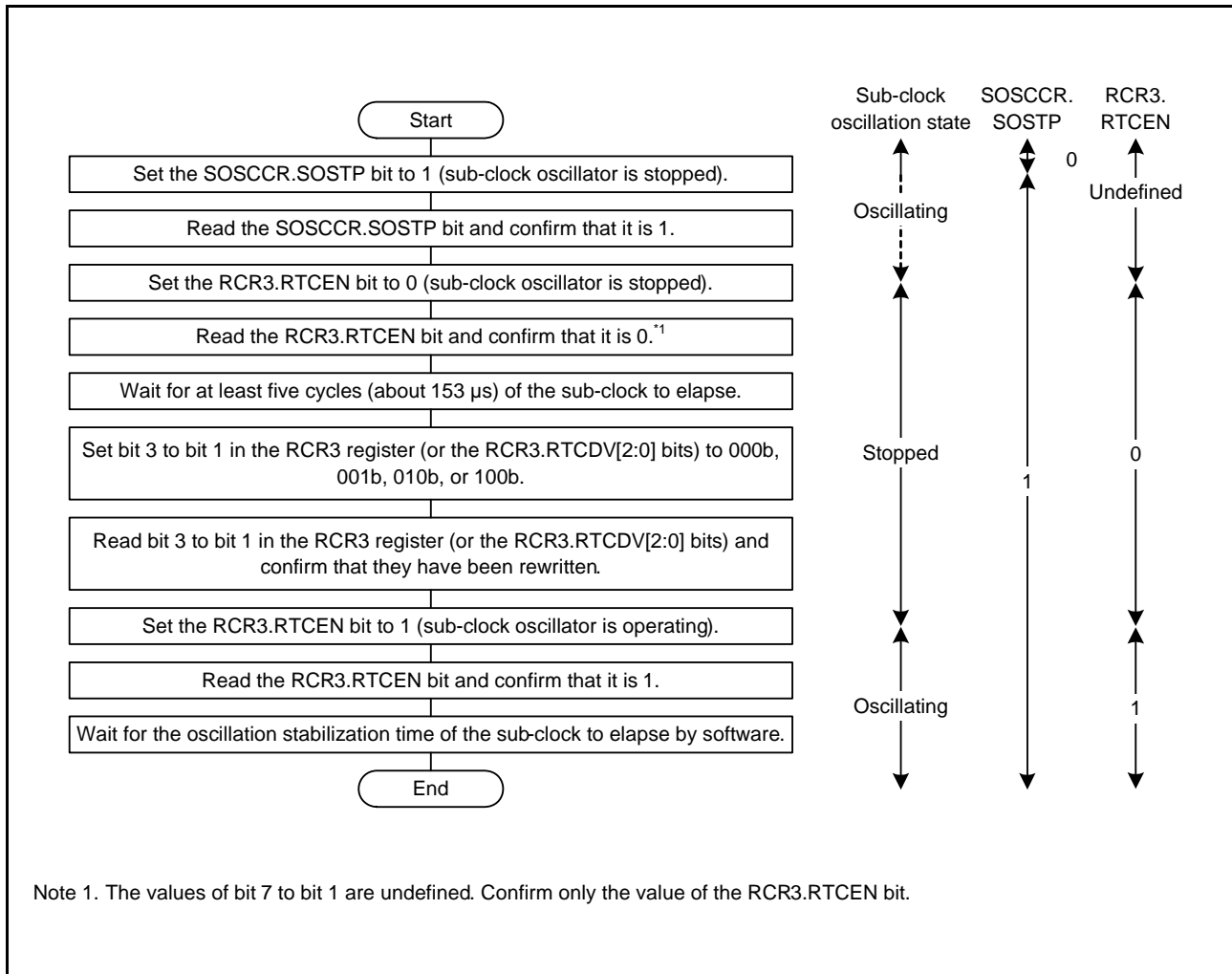


Figure 9.9 Example of Initialization Flowchart When Sub-Clock is Used as Count Source of Realtime Clock

- When using the sub-clock only as the count source of the realtime clock, perform initial settings according to the flowchart example shown in Figure 9.10. After that, perform the clock setting procedure shown in section 24.3.2, Clock and Count Mode Setting Procedure.



**Figure 9.10** Example of Initialization Flowchart When Sub-Clock is Used Only as Count Source of Realtime Clock



- When using the sub-clock only as the system clock, perform initial settings according to the flowchart example shown in Figure 9.11.

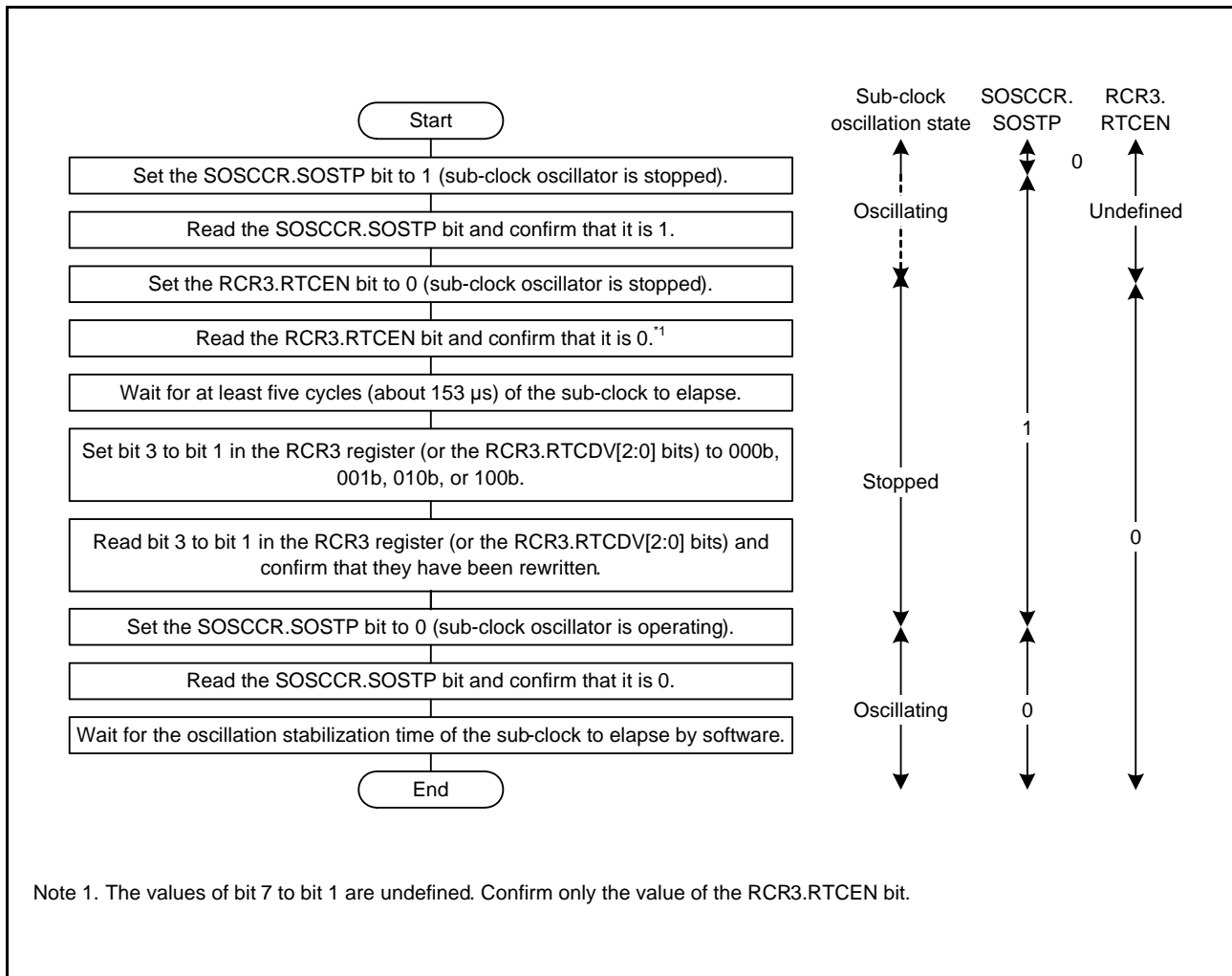
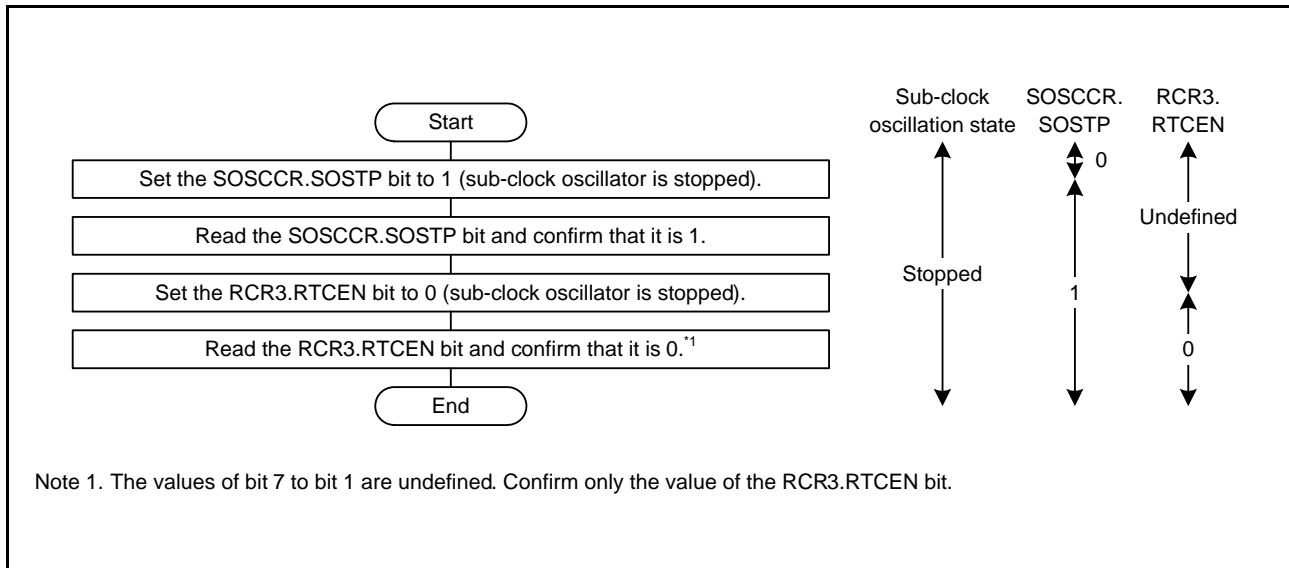


Figure 9.11 Example of Initialization Flowchart When Sub-Clock is Used Only as System Clock

- When not using the sub-clock, perform initial settings according to the flowchart example in Figure 9.12.



**Figure 9.12 Example of Initialization Flowchart When Sub-Clock is Not Used**

- Regardless of the RCR3.RTCEN bit setting, wait until the oscillator stabilization wait time elapses before rewriting the SOSCCR.SOSTP bit to 0 (sub-clock oscillator is operating).
- Since the sub-clock control circuit is in an unstable state after a cold start, it must be initialized regardless of whether or not the sub-clock is in use. The sub-clock is initialized by setting the SOSCCR.SOSTP bit to 1 and the RCR3.RTCEN bit to 0 (sub-clock oscillator is stopped). See section 24.2.19, RTC Control Register 3 (RCR3), for instructions to initialize the RCR3.RTCEN bit.  
Although the sub-clock oscillator pins are not available in 40 or fewer pin package products, initialize the sub-clock control circuit in the same way.
- The RCR3.RTCDV[2:0] bits must also be set when operating the sub-clock oscillator. Set these bits while the sub-clock oscillator is stopped. Do not rewrite these bits while the sub-clock oscillator is operating.
- When successively rewriting the SOSCCR.SOSTP bit followed by the RCR3.RTCEN bit or vice versa, confirm that the first bit rewrite was completed successfully before rewriting the second bit.

## 10. Clock Frequency Accuracy Measurement Circuit (CAC)

### 10.1 Overview

The clock frequency accuracy measurement circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock to be used as a measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range.

When measurement is completed or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.

Table 10.1 lists the specifications of the CAC and Figure 10.1 shows a block diagram of the CAC.

**Table 10.1 CAC Specifications**

Item	Description
Measurement target clocks	The frequency of the following clocks can be measured. <ul style="list-style-type: none"> <li>• Main clock</li> <li>• Sub-clock</li> <li>• HOCO clock</li> <li>• LOCO clock</li> <li>• IWDTCCLK clock</li> <li>• Peripheral module clock B (PCLKB)</li> </ul>
Measurement reference clocks	<ul style="list-style-type: none"> <li>• External clock input to the CACREF pin</li> <li>• Main clock</li> <li>• Sub-clock</li> <li>• HOCO clock</li> <li>• LOCO clock</li> <li>• IWDTCCLK clock</li> <li>• Peripheral module clock B (PCLKB)</li> </ul>
Selectable function	Digital filter function
Interrupt sources	<ul style="list-style-type: none"> <li>• Measurement end interrupt</li> <li>• Frequency error interrupt</li> <li>• Overflow interrupt</li> </ul>
Low power consumption function	Module stop state can be set.

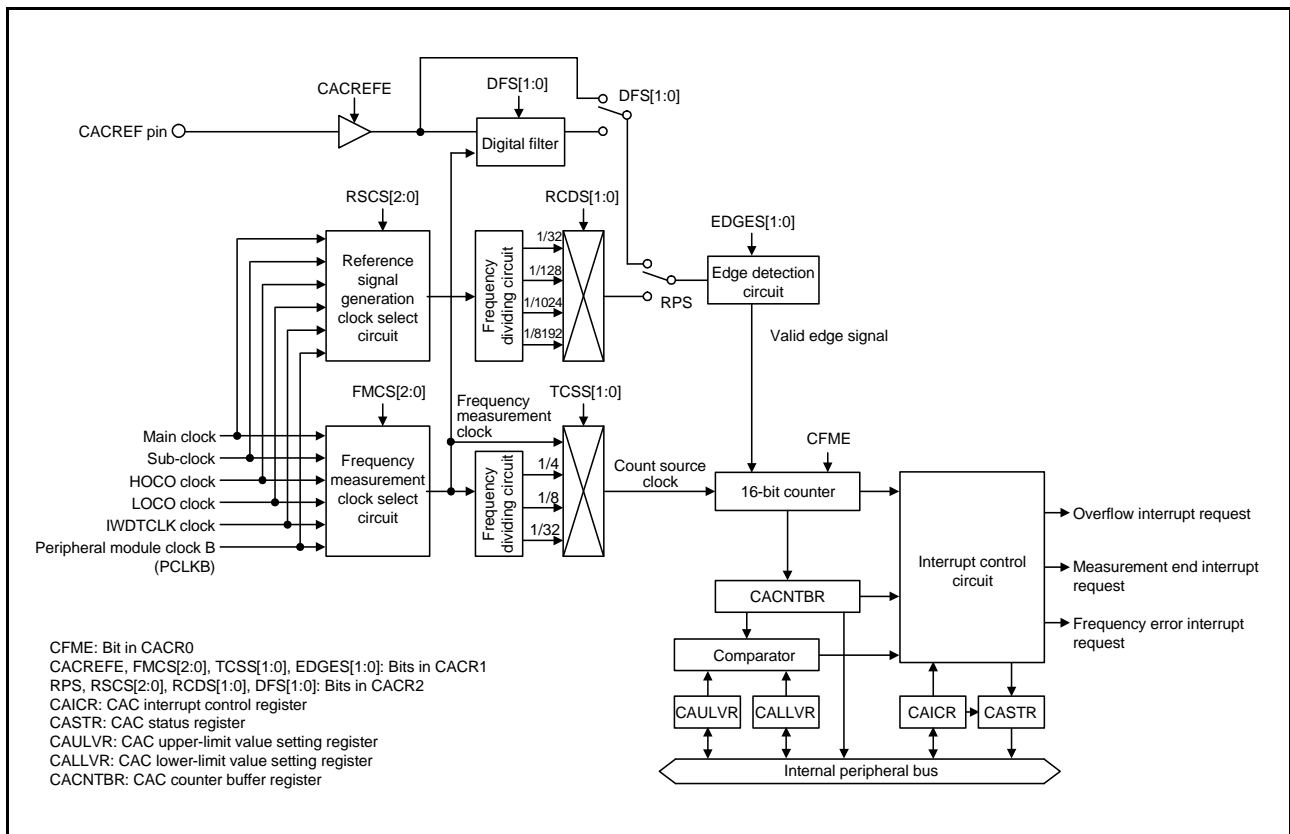


Figure 10.1 CAC Block Diagram

Table 10.2 shows the pin configuration of the CAC.

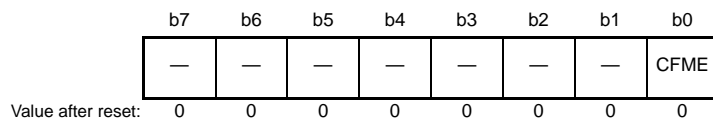
Table 10.2 Pin Configuration of CAC

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

## 10.2 Register Descriptions

### 10.2.1 CAC Control Register 0 (CACR0)

Address(es): 0008 B000h



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

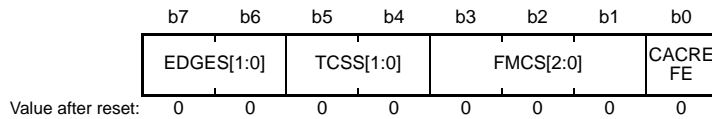
#### CFME Bit (Clock Frequency Measurement Enable)

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

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

## 10.2.2 CAC Control Register 1 (CACR1)

Address(es): 0008 B001h



Bit	Symbol	Bit Name	Description	R/W
b0	CACREFE	CACREF Pin Input Enable	0: CACREF pin input is disabled. 1: CACREF pin input is enabled.	R/W
b3 to b1	FMCS[2:0]	Measurement Target Clock Select	b3 b1 0 0 0: Main clock 0 0 1: Sub-clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDTCCLK clock 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	TCSS[1:0]	Timer Count Clock Source Select	b5 b4 0 0: No division 0 1: x1/4 clock 1 0: x1/8 clock 1 1: x1/32 clock	R/W
b7, b6	EDGES[1:0]	Valid Edge Select	b7 b6 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note 1. Set the CACR1 register when the CACR0.CFME bit is 0.

### CACREFE Bit (CACREF Pin Input Enable)

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

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

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

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

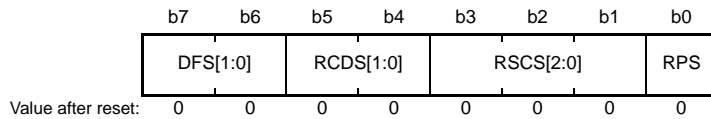
These bits select the count clock source for the clock frequency accuracy measurement circuit.

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

These bits select the valid edge for the reference signal.

### 10.2.3 CAC Control Register 2 (CACR2)

Address(es): 0008 B002h



Bit	Symbol	Bit Name	Description	R/W
b0	RPS	Reference Signal Select	0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
b3 to b1	RSCS[2:0]	Measurement Reference Clock Select	b3 b1 0 0 0: Main clock 0 0 1: Sub-clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDTCCLK clock 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ration Select	b5 b4 0 0: x1/32 clock 0 1: x1/128 clock 1 0: x1/1024 clock 1 1: x1/8192 clock	R/W
b7, b6	DFS[1:0]	Digital Filter Select	b7 b6 0 0: Digital filtering is disabled. 0 1: The sampling clock for the digital filter is the frequency measuring clock. 1 0: The sampling clock for the digital filter is the frequency measuring clock divided by 4. 1 1: The sampling clock for the digital filter is the frequency measuring clock divided by 16.	R/W

Note 1. Set the CACR2 register when the CACR0.CFME bit is 0.

#### RPS Bit (Reference Signal Select)

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

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

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

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

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

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

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

## 10.2.4 CAC Interrupt Request Enable Register (CAICR)

Address(es): 0008 B003h

b7	b6	b5	b4	b3	b2	b1	b0
—	OVFFC L	MENDF CL	FERRF CL	—	OVFIE	MENDI E	FERRI E

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

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

### FERRIE Bit (Frequency Error Interrupt Request Enable)

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

### MENDIE Bit (Measurement End Interrupt Request Enable)

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

### OVFIE Bit (Overflow Interrupt Request Enable)

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

### FERRFCL Bit (FERRF Clear)

Setting this bit to 1 clears the FERRF flag.

### MENDFCL Bit (MENDF Clear)

Setting this bit to 1 clears the MENDF flag.

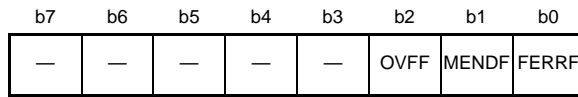
### OVFFCL Bit (OVFF Clear)

Setting this bit to 1 clears the OVFF flag.



## 10.2.5 CAC Status Register (CASTR)

Address(es): 0008 B004h



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

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

### FERRF Flag (Frequency Error Flag)

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

[Setting condition]

- The clock frequency is outside of the setting range.

[Clearing condition]

- 1 is written to the FERRFCL bit.

### MENDF Flag (Measurement End Flag)

This flag indicates the end of measurement.

[Setting condition]

- Measurement has finished.

[Clearing condition]

- 1 is written to the MENDFCL bit.

### OVFF Flag (Overflow Flag)

This flag indicates that the counter has overflowed.

[Setting condition]

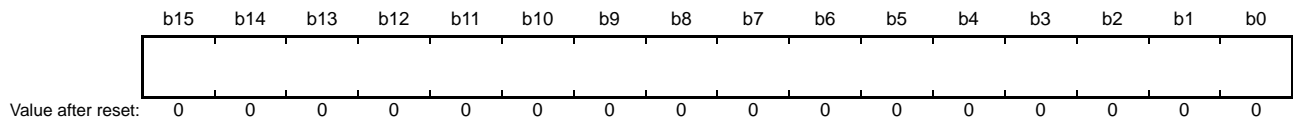
- The counter has overflowed.

[Clearing condition]

- 1 is written to the OVFFCL bit.

### 10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): 0008 B006h



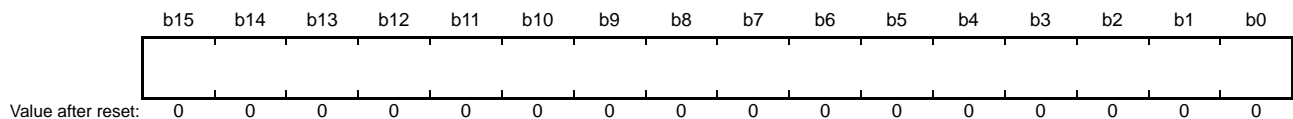
CAULVR is a 16-bit readable/writable register that specifies the upper-limit value of the counter used for measuring the frequency. When the frequency rises above the value specified in this register, a frequency error is detected.

Write to this register when the CACR0.CFME bit is 0.

The counter value held in CACNTBR can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

### 10.2.7 CAC Lower-Limit Value Setting Register (CALLVR)

Address(es): 0008 B008h



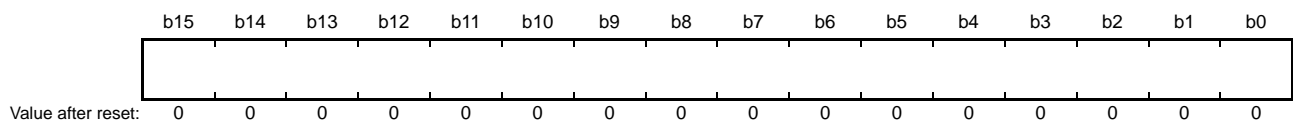
CALLVR is a 16-bit readable/writable register that specifies the lower-limit value of the counter used for measuring the frequency. When the frequency falls below the value specified in this register, a frequency error is detected.

Write to this register when the CACR0.CFME bit is 0.

The counter value held in CACNTBR can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

### 10.2.8 CAC Counter Buffer Register (CACNTBR)

Address(es): 0008 B00Ah



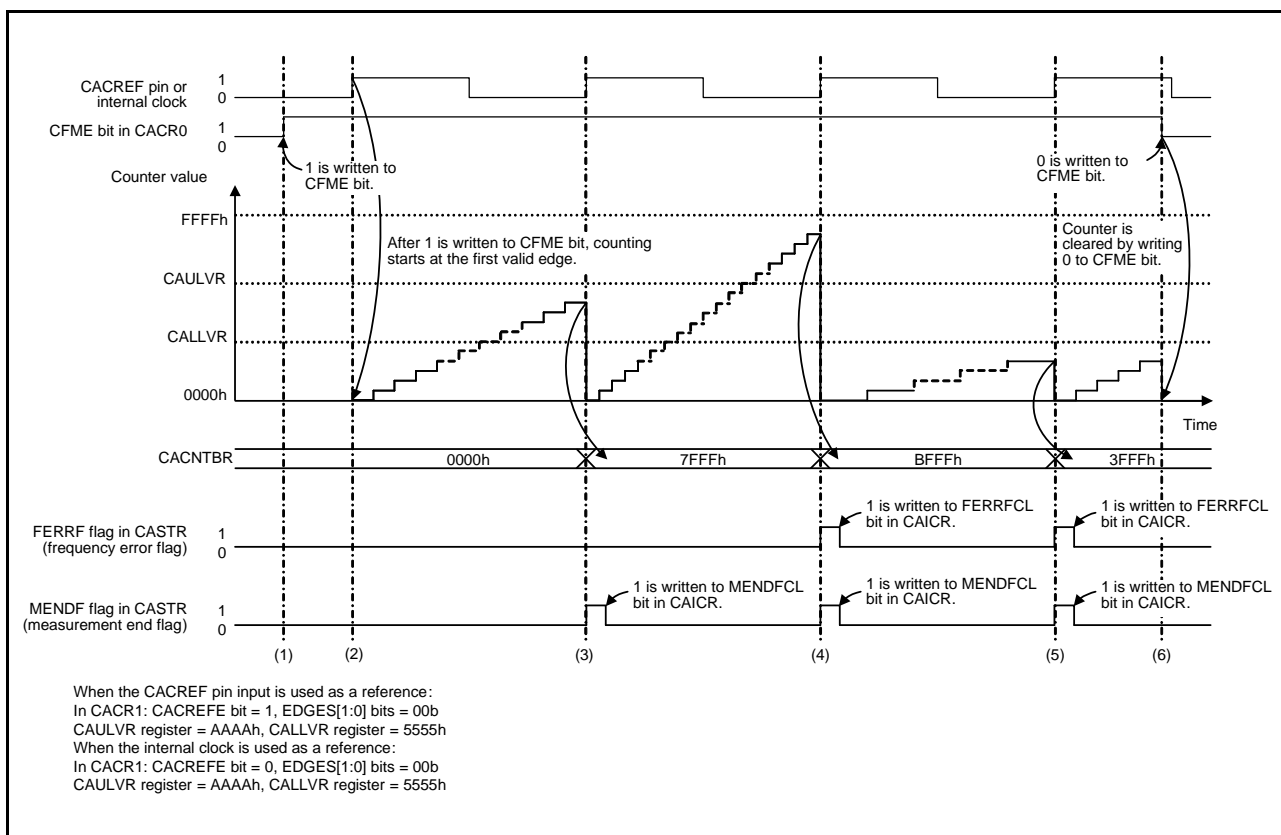
CACNTBR is a 16-bit read-only register that retains the counter value at the time a valid reference signal edge is input.

### 10.3 Operation

#### 10.3.1 Measuring Clock Frequency

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

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



**Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit**

- (1) When the CACREF pin input is used as a reference (CACR1.CACREFE bit = 1), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 0 and the CACR1.CACREFE bit is 1. On the other hand, when the internal clock is used as a reference (CACR1.CACREFE bit = 0), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 1.
- (2) When the CACREF pin input is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input from the CACREF pin after 1 is written to the CFME bit. The valid edge is a rising edge (CACR1.EDGES[1:0] = 00b) in Figure 10.2.  
 When the internal clock is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input based on the clock source selected by the CACR2.RSCS[2:0] bits after 1 is written to the CFME bit. The valid edge is a rising edge (CACR1.EDGES[1:0] = 00b) in Figure 10.2.
- (3) When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR. If both  $CACNTBR \leq CAULVR$  and  $CACNTBR \geq CALLVR$  are satisfied, only the MENDF flag in CASTR is set to 1 because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
- (4) When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR. In the case of  $CACNTBR > CAULVR$ , the FERRF flag in CASTR is set to 1 because the

- clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
- (5) When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR. In the case of CACNTBR < CALLVR, the FERRF flag in CASTR is set to 1 because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
- (6) While the CFME bit in CACR0 is 1, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

### 10.3.2 Digital Filtering of Signals on the CACREF Pin

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

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

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

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

$$\text{Counter value error} = (\text{One cycle of the count source clock}) / (\text{One cycle of the sampling clock})$$

## 10.4 Interrupt Requests

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

**Table 10.3 Interrupt Requests of Clock Frequency Accuracy Measurement Circuit**

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

## 10.5 Usage Notes

### 10.5.1 Module Stop Function Setting

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

## 11. Low Power Consumption

### 11.1 Overview

This MCU has several functions for reducing power consumption, by setting clock dividers, stopping modules, changing to low power consumption mode in normal operation, and changing to operating power control mode.

Table 11.1 lists the specifications of low power consumption functions, and Table 11.2 lists the conditions to change to low power consumption modes, states of the CPU and peripheral modules, and the method for exiting each mode.

After a reset, this MCU returns to normal mode, but modules except the DTC and RAM are stopped.

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

Item	Specification
Clock divider functions	The frequency division ratio can be set independently for the system clock (ICLK), peripheral module clock (PCLKB), S12AD clock (PCLKD), and FlashIF clock (FCLK).*1
Module stop function	Each peripheral module can be stopped independently by the module stop control register.
Low power consumption modes	Power consumption can be reduced by selecting an appropriate power consumption mode according to which module is need to operate.*2 Three low power consumption modes are available.*3 <ul style="list-style-type: none"> <li>• Sleep mode</li> <li>• Deep sleep mode</li> <li>• Software standby mode</li> </ul>
Operating power control modes	Power consumption can be reduced in normal operation, sleep mode, and deep sleep mode by selecting an appropriate operating power control mode according to the operating frequency and operating voltage. Three operating power control modes are available <ul style="list-style-type: none"> <li>• High-speed operating mode</li> <li>• Middle-speed operating mode</li> <li>• Low-speed operating mode</li> </ul>

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

Note 2. For details, refer to section 11.6, Low Power Consumption Modes.

Note 3. For details, refer to Figure 11.2, Operating Conditions of Each Power Consumption Mode.

**Table 11.2 Operating Conditions of Each Power Consumption Mode**

	<b>Sleep Mode</b>	<b>Deep Sleep Mode</b>	<b>Software Standby Mode</b>
Entry trigger	Control register + instruction	Control register + instruction	Control register + instruction
Exit trigger	Interrupt	Interrupt	Interrupt* <sup>1</sup>
After exiting from each mode, CPU begins from* <sup>2</sup>	Interrupt handling	Interrupt handling	Interrupt handling
Main clock oscillator	Operating possible	Operating possible	Stopped
Sub-clock oscillator	Operating possible	Operating possible	Operating possible
High-speed on-chip oscillator	Operating possible	Operating possible	Stopped
Low-speed on-chip oscillator	Operating possible	Operating possible	Stopped
IWDT-dedicated on-chip oscillator	Operating possible* <sup>3</sup>	Operating possible* <sup>3</sup>	Operating possible* <sup>3</sup>
PLL	Operating possible	Operating possible	Stopped
USB-dedicated PLL	Operating possible	Operating possible	Stopped
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
RAM0 (0000 0000h to 0000 3FFFh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)
DTC	Operating possible* <sup>5</sup>	Stopped (Retained)	Stopped (Retained)
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)
Independent watchdog timer (IWDT)	Operating possible* <sup>3</sup>	Operating possible* <sup>3</sup>	Operating possible* <sup>3</sup>
Realtime clock (RTC)	Operating possible	Operating possible	Operating possible
Voltage detection circuit (LVD)	Operating possible	Operating possible	Operating possible
Power-on reset circuit	Operating	Operating	Operating
Peripheral modules	Operating possible	Operating possible	Stopped (Retained)* <sup>4</sup>
I/O ports	Operating	Operating	Retained
RTCOU	Operating possible	Operating possible	Operating possible
CLKOUT	Operating possible	Operating possible	Operating possible* <sup>6</sup>
Comparator B	Operating possible	Operating possible	Operating possible* <sup>7</sup>
LCD controller/driver	Operating possible* <sup>8</sup>	Operating possible* <sup>8</sup>	Operating possible* <sup>9</sup>

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

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

- Note 1. "Interrupts" here indicates an external pin interrupt (the NMI or IRQ0 to IRQ7) or any of peripheral interrupts (the RTC alarm, RTC interval, IWDT, voltage monitoring, and USB interrupts).
- Note 2. This does not include a RES# pin reset, power-on reset, voltage monitoring reset, or independent watchdog-timer reset. One of these reset sources initiate transition to reset state.
- Note 3. Operating or stopping is selected by setting the IWDT sleep mode count stop control bit (IWDTSLCSTP) in option function select register 0 (OFS0) in IWDT auto-start mode. In any mode other than IWDT auto-start mode, operating or stopping is selected by the setting of the sleep mode count stop control bit (SLCSTP) in the IWDT count stop control register (IWDTCSSTPR).
- Note 4. The peripheral logic states are retained.
- Note 5. During sleep mode, do not write to the system control related registers (indicated by 'SYSTEM' in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order)).
- Note 6. Stopped when the clock output source select bits (CKOCR.CKOSSEL[2:0]) are set to a value other than 011b (sub-clock oscillator).
- Note 7. Using the digital filter function is prohibited. Operation for outputting the comparison result to the CMPOBn pin is possible.
- Note 8. Operating or stopping is selected according to Note 3 when 100b (IWDT-dedicated on-chip oscillator) is selected by the LCDSCLKCR.LCDSCLKSEL[2:0] bits.
- Note 9. Operating or stopping is selected according to Note 3 when the LCDSCLKCR.LCDSCLKSEL[2:0] bits = 100b (IWDT-dedicated on-chip oscillator) is selected. Operation is possible when 011b (sub-clock oscillator) is selected by the LCDSCLKCR.LCDSCLKSEL[2:0] bits. Stopping is selected when the LCDSCLKCR.LCDSCLKSEL[2:0] bits are set to a value other than 100b or 011b.

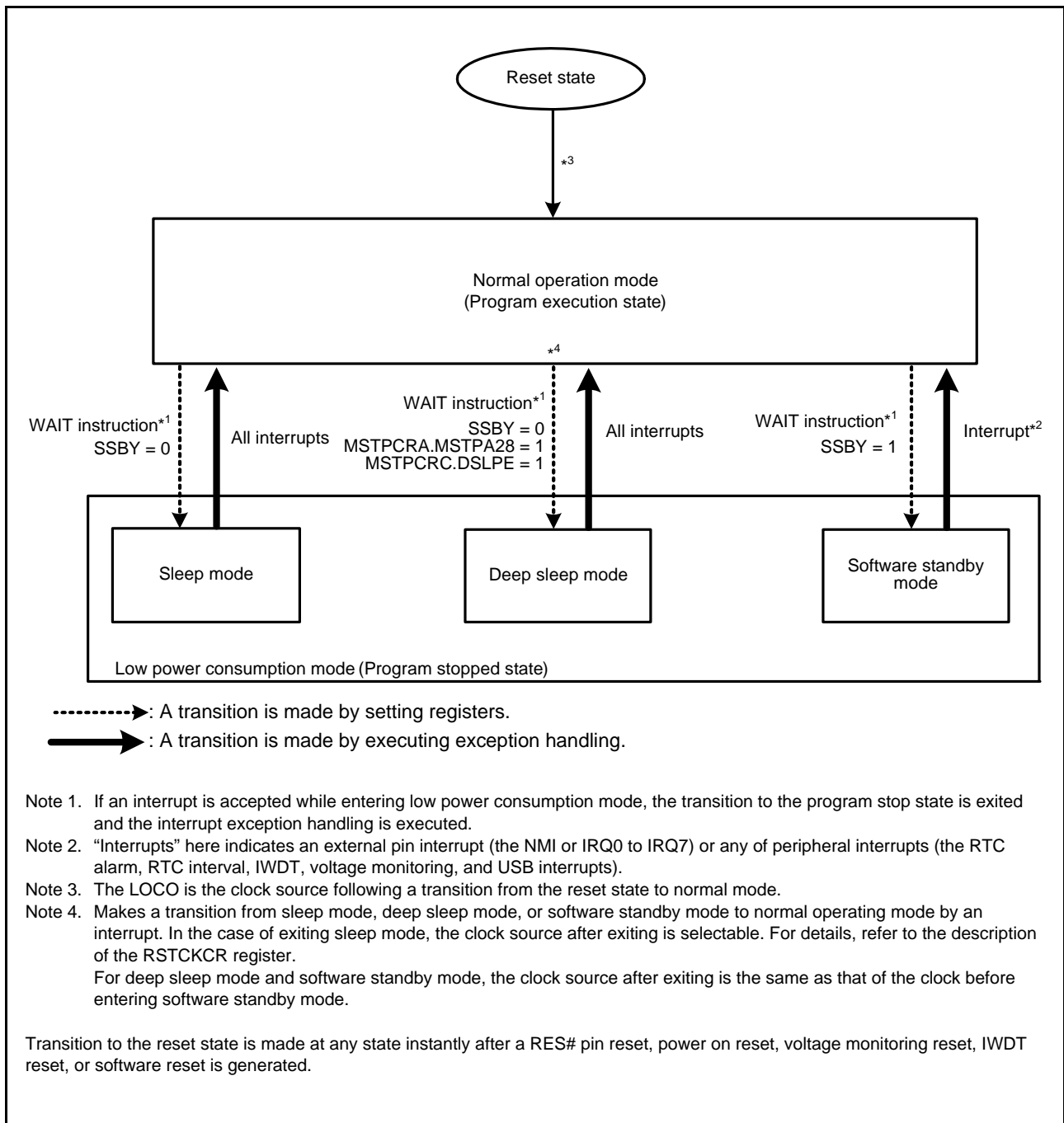


Figure 11.1 Mode Transitions



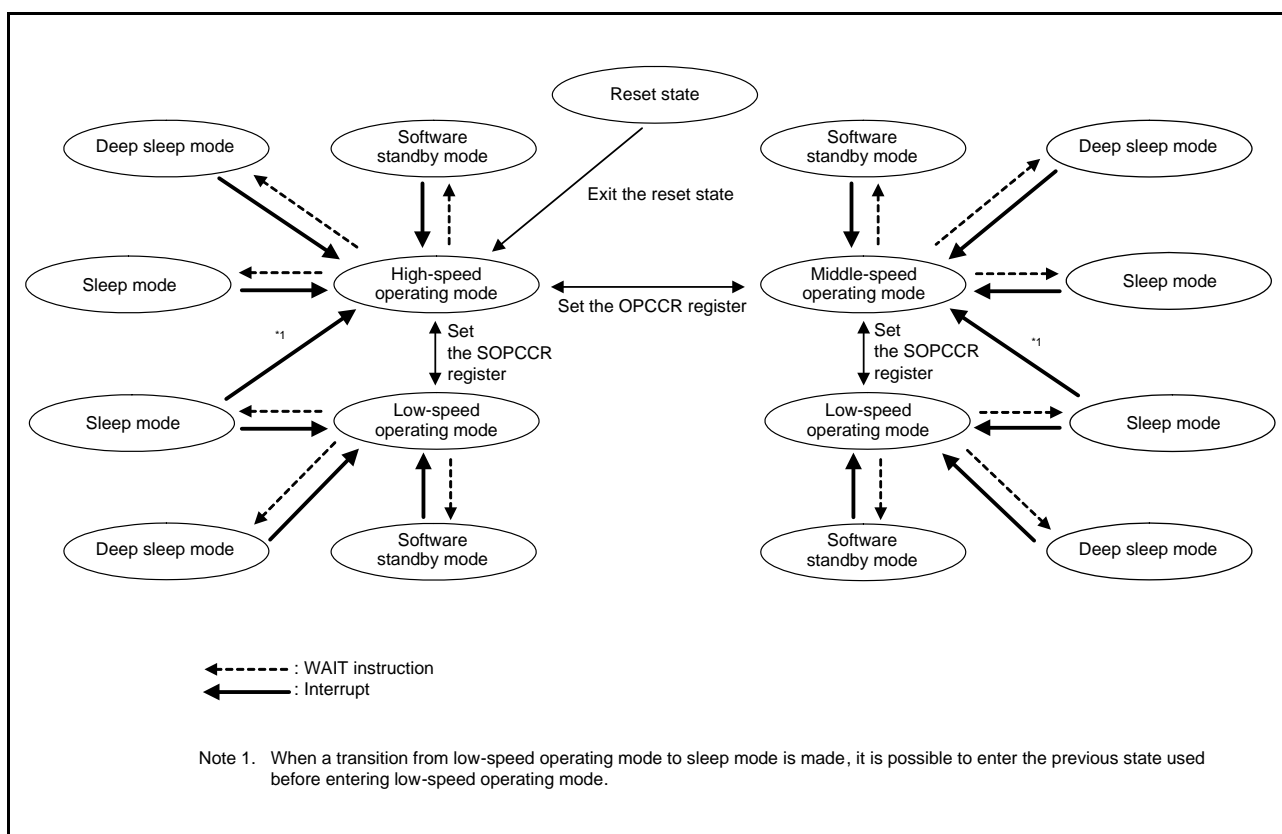


Figure 11.2 Operating Modes

- The sub-clock oscillator does not stop when entering software standby mode.
- It is possible to return from sleep mode to the previous operating state used before entering sleep mode. However, when a transition from low-speed operating mode to sleep mode is made, it is possible to enter the previous state used before entering low-speed operating mode.
- After exiting the reset state, operation starts in high-speed operating mode.

Table 11.3 Oscillator Usability in Each Mode

	PLL	USB-dedicated PLL	HOCO	LOCO	ILOCO	Main Clock Oscillator	Sub-Clock Oscillator
High-speed operating mode	Usable*1	Usable*1	Usable	Usable	Usable	Usable	Usable
Middle-speed operating mode	Usable*1	Usable*1	Usable	Usable	Usable	Usable	Usable
Low-speed operating mode	Not usable	Not usable	Not usable	Not usable	Usable	Not usable	Usable

Note 1. The PLL and USB-dedicated PLL are usable when the power supply voltage is 2.4 V or above.

## 11.2 Register Descriptions

### 11.2.1 Standby Control Register (SBYCR)

Address(es): 0008 000Ch

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

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

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

#### SSBY Bit (Software Standby)

The SSBY bit specifies the transition destination after the WAIT instruction is executed.

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

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

## 11.2.2 Module Stop Control Register A (MSTPCRA)

Address(es): 0008 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	MSTPA 28	—	—	—	—	—	—	—	—	MSTPA 19	MSTPA 18	MSTPA 17	—
Value after reset:	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPA 15	MSTPA 14	—	—	—	—	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 and 2 (Unit 1) Module Stop	Target module: TMR3, TMR2 0: This module clock is enabled 1: This module clock is disabled	R/W
b5	MSTPA5	8-bit Timer 1 and 0 (Unit 0) Module Stop	Target module: TMR1, TMR0 0: This module clock is enabled 1: This module clock is disabled	R/W
b8 to b6	—	Reserved	These bits are 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: This module clock is enabled 1: This module clock is disabled	R/W
b13 to b10	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b14	MSTPA14	Compare Match Timer 1 (Unit 1) Module Stop	Target module: CMT unit 1 (CMT2, CMT3) 0: This module clock is enabled 1: This module clock is disabled	R/W
b15	MSTPA15	Compare Match Timer (Unit 0) Module Stop	Target module: CMT unit 0 (CMT0, CMT1) 0: This module clock is enabled 1: This module clock is disabled	R/W
b16	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b17	MSTPA17	12-Bit A/D Converter Module Stop	Target module: S12AD 0: This module clock is enabled 1: This module clock is disabled	R/W
b18	MSTPA18	12-Bit D/A Converter Module Stop	Target module: DA 0: This module clock is enabled 1: This module clock is disabled	R/W
b27 to b19	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b28	MSTPA28	Data Transfer Controller Module Stop	Target module: DTC 0: This module clock is enabled 1: This module clock is disabled	R/W
b31 to b29	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

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

### 11.2.3 Module Stop Control Register B (MSTPCRB)

Address(es): 0008 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPB 31	MSTPB 30	MSTPB 29	—	—	MSTPB 26	MSTPB 25	—	MSTPB 23	—	MSTPB 21	—	MSTPB 19	—	MSTPB 17	—
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
	—	—	—	—	—	MSTPB 10	MSTPB 9	—	—	MSTPB 6	—	MSTPB 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	MSTPB4	Serial Communication Interface SCIf Module Stop	Target module: SCIf (SCI12) 0: This module clock is enabled 1: This module clock is disabled	R/W
b5	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6	MSTPB6	DOC Module Stop	Target module: DOC 0: This module clock is enabled 1: This module clock is disabled	R/W
b8, b7	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b9	MSTPB9	ELC Module Stop	Target module: ELC 0: This module clock is enabled 1: This module clock is disabled	R/W
b10	MSTPB10	Comparator B Module Stop	Target module: Comparator B 0: This module clock is enabled 1: This module clock is disabled	R/W
b16 to b11	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b17	MSTPB17	Serial Peripheral Interface 0 Module Stop	Target module: RSPi0 0: This module clock is enabled 1: This module clock is disabled	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPB19 *1	USB0 Module Stop	Target module: USB0 0: This module clock is enabled 1: This module clock is disabled	R/W
b20	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b21	MSTPB21	I <sup>2</sup> C Bus Interface 0 Module Stop	Target module: RIIC0 0: This module clock is enabled 1: This module clock is disabled	R/W
b22	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b23	MSTPB23	CRC Calculator Module Stop	Target module: CRC 0: This module clock is enabled 1: This module clock is disabled	R/W
b24	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b25	MSTPB25	Serial Communication Interface 6 Module Stop	Target module: SCI6 0: This module clock is enabled 1: This module clock is disabled	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: This module clock is enabled 1: This module clock is disabled	R/W
b28 to b27	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Bit	Symbol	Bit Name	Description	R/W
b29	MSTPB29	Serial Communication Interface 2 Module Stop	Target module: SCI2 0: This module clock is enabled 1: This module clock is disabled	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: This module clock is enabled 1: This module clock is disabled	R/W
b31	MSTPB31	Serial Communication Interface 0 Module Stop	Target module: SCI0 0: This module clock is enabled 1: This module clock is disabled	R/W

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

Note 1. This bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. The clock should be set for oscillation when this bit is 0 (this module clock is enabled). When entering software standby mode after rewriting this bit, wait for two cycles of the UCLK after rewriting, and execute a WAIT instruction. When stopping the clock after rewriting this bit to 1 (this module clock is disabled), wait for two cycles of the UCLK after rewriting, and stop the clock.

## 11.2.4 Module Stop Control Register C (MSTPCRC)

Address(es): 0008 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DSLPE	—	—	—	MSTPC 27	MSTPC 26	—	—	—	—	—	MSTPC 20	MSTPC 19	—	—	—
Value after reset:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPC 0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	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
b18 to b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b19	MSTPC19	Clock Frequency Accuracy Measurement Circuit Module Stop*2	Target module: CAC 0: This module clock is enabled 1: This module clock is disabled	R/W
b20	MSTPC20	IrDA Module Stop	Target module: IRDA 0: This module clock is enabled 1: This module clock is disabled	R/W
b25 to b21	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b26	MSTPC26	Serial Communication Interface 9 Module Stop	Target module: SCI9 0: This module clock is enabled 1: This module clock is disabled	R/W
b27	MSTPC27	Serial Communication Interface 8 Module Stop	Target module: SCI8 0: This module clock is enabled 1: This module clock is disabled	R/W
b30 to b28	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b31	DSLPE	Deep Sleep Mode Enable	0: Deep sleep mode is disabled 1: Deep sleep mode is enabled	R/W

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

Note 1. The MSTPC0 bit should not be set to 1 during access to the corresponding RAM. The corresponding RAM should not be accessed while the MSTPC0 bit is set to 1.

Note 2. The MSTPC19 bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. For entering software standby mode after rewriting this bit, wait for two cycles of the slowest clock among the clocks output by the oscillators actually oscillating and execute the WAIT instruction.

### DSLPE Bit (Deep Sleep Mode Enable)

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

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

### 11.2.5 Module Stop Control Register D (MSTPCRD)

Address(es): 0008 001Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPD 15	—	—	—	MSTPD 11	MSTPD 10	—	—	—	—	—	—	—	—	—	—
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
b9 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b10	MSTPD10	Touch Sensor Control Unit Module Stop	Target module: CTSU 0: This module clock is enabled 1: This module clock is disabled	R/W
b11	MSTPD11	LCD Controller Module Stop	Target module: LCDC 0: This module clock is enabled 1: This module clock is disabled	R/W
b14 to b12	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b15	MSTPD15	Serial Sound Interface Module Stop	Target module: SSI 0: This module clock is enabled 1: This module clock is disabled	R/W
b31 to b16	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

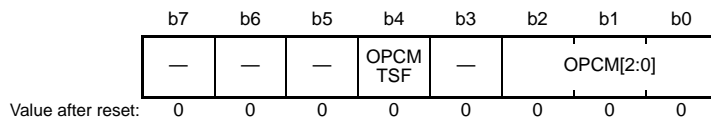
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. When the main clock oscillator is used by the serial sound interface

This bit should be rewritten when the oscillation of the main clock oscillator is stable. When entering software standby mode after rewriting this bit, wait for two cycles of the main clock after rewriting, and execute a WAIT instruction. When stopping the main clock oscillator after rewriting this bit, wait for two cycles of the main clock after rewriting, and stop the main clock oscillator.

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

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

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

The OPCCR register cannot be rewritten under the following conditions:

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

The OPCCR register cannot be rewritten while the flash memory is being programmed or erased (P/E).

For the procedures of changing operating power control modes, refer to Function in section 11.5, Function for Lower Operating Power Consumption.

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

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

The OPCM[2:0] bits select operating power control mode in normal operating mode, sleep mode, and deep sleep mode. Table 11.4 shows the relationship between operating power control modes, the OPCM[2:0] and SOPCM bit settings, and the operating frequency and voltage ranges.

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

This flag indicates the switching control state during and after operating power mode transition.

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



## 11.2.7 Sub Operating Power Control Register (SOPCCR)

Address(es): 0008 00AAh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	SOPC MTSF	—	—	—	SOPC M
0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b0	SOPCM	Sub Operating Power Control Mode Select	0: High-speed operating mode or middle-speed operating mode*1 1: Low-speed operating mode	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SOPCMTSF	Sub Operating Power Control Mode Transition Status Flag	0: Transition completed 1: During transition	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

Note 1. Depends on the setting of OPCCR.OPCM[2:0].

The SOPCCR register is used to reduce power consumption in normal operating mode, sleep mode, and deep sleep mode by controlling a transition to low-speed operating mode.

Setting this register initiates entry to/exit from low-speed operating mode.

Low-speed operating mode is used for the sub-clock oscillator only.

The OPCCR register cannot be rewritten when the SOPCM bit is 1 (low-speed operating mode).

The SOPCCR register cannot be rewritten under the following conditions:

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

This register cannot be rewritten while the flash memory is being programmed or erased (P/E).

For the procedures for changing operating power control modes, refer to Function in section 11.5, Function for Lower Operating Power Consumption.

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

### SOPCM Bit (Sub Operating Power Control Mode Select)

The SOPCM bit selects operating power control in normal operating mode and sleep mode.

Setting this bit to 1 allows a transition to low-speed operating mode. Setting this bit to 0 allows a return to the operating mode (operating mode set by OPCCR.OPCM[2:0]) before the transition to low-speed operating mode.

Table 11.4 shows the relationship between operating power control modes, the OPCM[2:0] and SOPCM bit settings, and the operating frequency and voltage ranges.

**SOPCMTSF Flag (Sub Operating Power Control Mode Transition Status Flag)**

The SOPCMTSF flag indicates the switching control state when the sub operating power control mode is switched. This flag becomes 1 when the value of the SOPCM bit is rewritten, and 0 when mode transition is completed. Read this flag and confirm that it is 0 before proceeding to the next processing. Only rewrite the SOPCM bit when this flag is 0.

**Table 11.4 Operating Frequency and Voltage Ranges in Operating Power Control Modes**

Operating Power Control Mode	OPCM [2:0] Bits	SOPCM Bit	Operating Voltage Range	Operating Frequency Range				
				Flash Memory Read Frequency				Flash Memory Programming/ Erasure Frequency
				ICLK	FCLK	PCLKD	PCLKB	FCLK
High-speed operating mode	000b	0	2.7 to 3.6 V	Up to 32 MHz	Up to 32 MHz	Up to 32 MHz	Up to 32 MHz	1 to 32 MHz
			2.4 to 2.7 V	Up to 16 MHz	Up to 16 MHz	Up to 16 MHz	Up to 16 MHz	—
			1.8 to 2.4 V	Up to 8 MHz	Up to 8 MHz	Up to 8 MHz	Up to 8 MHz	—
Middle-speed operating mode	010b	0	2.4 to 3.6 V	Up to 12 MHz	Up to 12 MHz	Up to 12 MHz	Up to 12 MHz	1 to 12 MHz
			1.8 to 2.4 V	Up to 8 MHz	Up to 8 MHz	Up to 8 MHz	Up to 8 MHz	1 to 8 MHz
Low-speed operating mode	000b	1	1.8 to 3.6 V	Up to 32.768 kHz	Up to 32.768 kHz	Up to 32.768 kHz	Up to 32.768 kHz	—
	010b	1	1.8 to 3.6 V					

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

Each operating power control mode is described below.

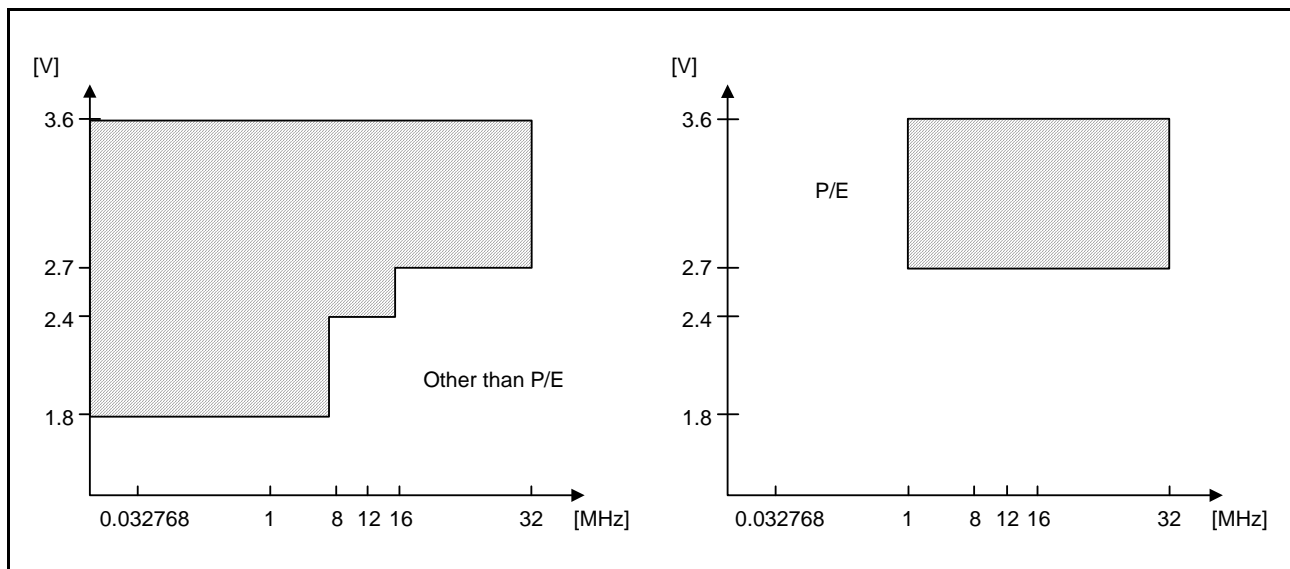
- High-Speed Operating Mode

The maximum operating frequency during FLASH read is 32 MHz for ICLK, FCLK, PCLKD, and PCLKB. The operating voltage range is 1.8 to 3.6 V during FLASH read. However, for ICLK, FCLK, PCLKD, and PCLKB, the maximum operating frequency during FLASH read is 16 MHz when the operating voltage is 2.4 V or larger and smaller than 2.7 V. The maximum operating frequency during FLASH read is 8 MHz for all the clocks when the operating voltage is 1.8 V or larger and smaller than 2.4 V.

During FLASH programming/erasure, the operating frequency range is 1 to 32 MHz and the operating voltage range is 2.7 to 3.6 V.

After a reset is canceled, operation is started from this mode.

Figure 11.3 shows the operating voltages and frequencies in high-speed operating mode.



**Figure 11.3** Operating Voltages and Frequencies in High-Speed Operating Mode

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

- Middle-Speed Operating Mode

As compared to high-speed operating mode, this mode reduces power consumption for low-speed operation.

The maximum operating frequency during FLASH read is 12 MHz for ICLK, FCLK, PCLKD, and PCLKB. The

operating voltage range is 1.8 to 3.6 V during FLASH read. The maximum operating frequency during FLASH read is 8 MHz for all the clocks when the operating voltage is 1.8 V or larger and smaller than 2.4 V.

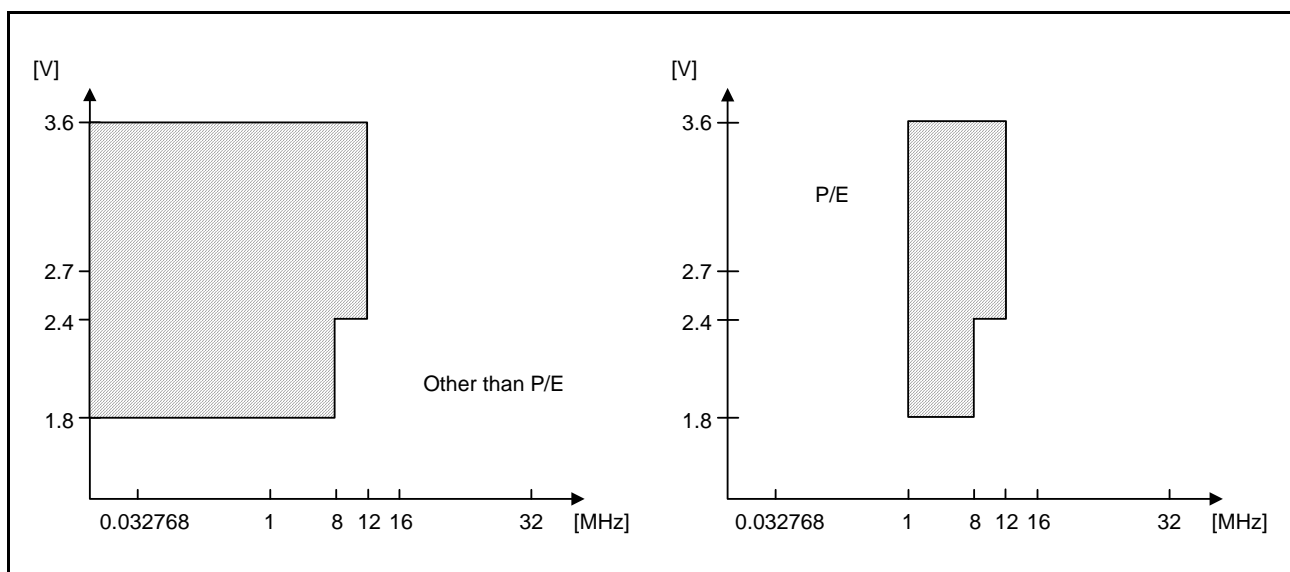
During FLASH programming/erasure, the operating frequency range is 1 to 12 MHz and the operating voltage range is 1.8 to 3.6 V. The maximum operating frequency during FLASH programming/erasure is 8 MHz when the operating voltage is 1.8 V or larger and smaller than 2.4 V.

The power consumption of this mode is lower than that of high speed mode under the same conditions.

The following restriction applies when middle-speed operating mode is selected:

- The PLL and USB-dedicated PLL can be used when the operating voltage is 2.4 V or above.

Figure 11.4 shows the operating voltages and frequencies in middle-speed operating mode.



**Figure 11.4 Operating Voltages and Frequencies in Middle-Speed Operating Mode**

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

- Low-Speed Operating Mode

A transition to low-speed operating mode is set by writing 1 to the SOPCM bit in the SOPCCR register. The setting of the OPCM[2:0] bits cannot be modified during low-speed operating mode. This mode is used only for the sub oscillator of 32.768 kHz.

During reading the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKD, and PCLKB is 32.768 kHz. The operating voltage is in the range of 1.8 to 3.6 V.

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

- P/E operations for flash memory are prohibited.
- The PLL/USB-dedicated PLL, main clock oscillator, LOCO, and HOCO cannot be used.

Note: The SOPCM bit cannot be set to 1 when the PLLCR2.PLEN bit is 0 (PLL is operating).  
 The SOPCM bit cannot be set to 1 when the UPLLCR2.UPLEN bit is 0 (USB-dedicated PLL is operating).  
 The SOPCM bit cannot be set to 1 when the HOCOCR.HCSTP bit is 0 (HOCO is operating).  
 The SOPCM bit cannot be set to 1 when the MOSCCR.MOSTP bit is 0 (MOSC is operating).  
 The SOPCM should not be set to 1 when the LOCOCR.LCSTP bit is 0 (LOCO is operating).

Figure 11.5 shows the operating voltages and frequencies in low-speed operating mode.

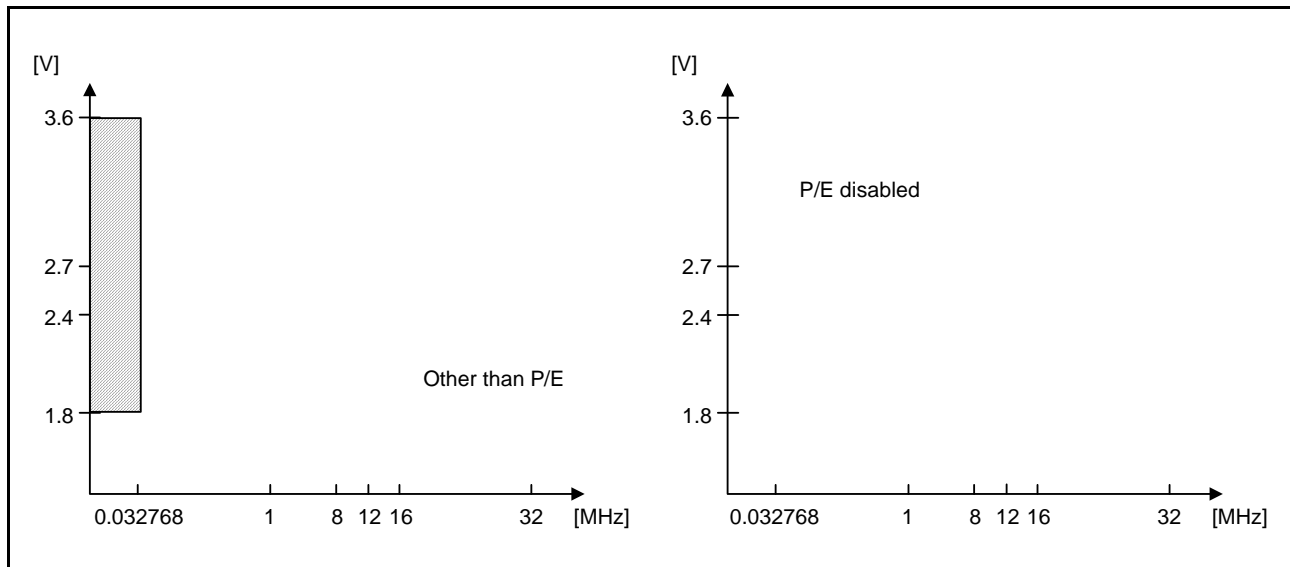
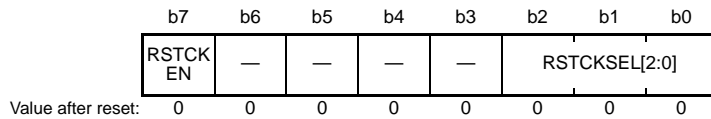


Figure 11.5 Operating Voltages and Frequencies in Low-Speed Operating Mode

## 11.2.8 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 0: LOCO is selected 0 0 1: HOCO is selected*1 0 1 0: Main clock oscillator is selected Settings other than above are prohibited when 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 exit from sleep mode is disabled 1: Clock source switching at exit from sleep mode is enabled	R/W

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

Note 1. HOCO can only be selected when entering high-speed operating mode.

RSTCKCR is used to control clock source switching at exit from sleep mode.

When exit from sleep mode is initiated by setting RSTCKCR, the main clock oscillator stop bit in the main clock oscillator control register (MOSCCR.MOSTP), the HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCCR.HCSTP), and the LOCO stop bit in the low-speed on-chip oscillator control register (LOCOCCR.LCSTP) are automatically modified to the operating state corresponding to the clock source to be used after transition. The value of the RSTCKSEL[2:0] bits is automatically reloaded to the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

### RSTCKSEL[2:0] Bits (Sleep Mode Return Clock Source Select)

The RSTCKSEL[2:0] bits select the clock source to be used at exit from sleep mode.

The clock source selected by the RSTCKSEL[2:0] bits is enabled only when the RSTCKEN bit is 1.

As shown in Figure 11.2, Operating Modes, when returning from sleep mode to high-speed operating mode, the LOCO, HOCO, or main clock oscillator can be selected. When returning from sleep mode to middle-speed operating mode, the LOCO or main clock oscillator can be selected. However, in this case, the frequency of each clock (ICLK, FCLK, PCLKD, and PCLKB) must be lower than 12 MHz when the power supply voltage is 2.4 V or above, and lower than 8 MHz when the voltage is below 2.4 V.

**Table 11.5 When Exiting Sleep Mode to High-Speed Operating Mode and Middle-Speed Operating Mode**

Operating Mode during Sleep	Clock Source during Sleep	RSTCKSEL	Operating Mode after Exiting	Clock Source after Exiting
High-speed operating mode or low-speed operating mode after exit from high-speed operating mode	Sub-clock oscillator	000b (LOCO)	High-speed operating mode	LOCO
		001b (HOCO)		HOCO
		010b (main clock oscillator)		Main clock oscillator
Middle-speed operating mode or low-speed operating mode after exit from middle-speed operating mode	Sub-clock oscillator	000b (LOCO)	Middle-speed operating mode	LOCO
		010b (main clock oscillator)		Main clock oscillator*1

Note 1. The frequency of each clock (ICLK, FCLK, PCLKD, and PCLKB) must be lower than 12 MHz when the power supply voltage is 2.4 V or above, and lower than 8 MHz when the voltage is below 2.4 V.

**RSTCKEN Bit (Sleep Mode Return Clock Source Switching Enable)**

The RSTCKEN bit enables or disables clock source switching when sleep mode is exited.

When returning from sleep mode while this bit is enabled, the SOPCM bit in the SOPCCR register is automatically rewritten to 0 (middle-speed operating mode or high-speed operating mode).

The value of the frequency division setting (in the SCKCR register) is retained.

To exit sleep mode to middle-speed operating mode when the main clock oscillator is selected, the frequency of each clock must be lower than 12 MHz when the power supply voltage is 2.4 V or above, and lower than 8 MHz when the voltage is below 2.4 V.

### 11.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency can change by setting the SCKCR.FCK[3:0], ICK[3:0], PCKB[3:0], and PCKD[3:0] bits. The CPU, DTC, ROM, and RAM clocks can be set by the ICK[3:0] bits. The peripheral module clocks can be set by the PCKB[3:0] and PCKD[3:0] bits.

The flash memory clock can be set by the FCK[3:0] bits.

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

### 11.4 Module Stop Function

The module stop function can be set for each on-chip peripheral module.

When the MSTPmi bit (m = A to D; i = 0 to 31) in MSTPCRA to MSTPCRD is set to 1, the specified module stops operating and enters the module stop state, but the CPU continues to operate independently. When the corresponding MSTPmi bit is set to 0, the module exits the module state and restarts operating at the end of the bus cycle. The internal states of modules are retained in the module stop state.

After a reset is canceled, all modules other than the DTC, and on-chip RAM are in the module stop state. Basically the registers in the module stop state cannot be read or written. However, note that data may be written to these registers if write access is made immediately after the setting of the module stop state. To avoid this, always write to the module stop registers after confirming that the last register setting is done.

### 11.5 Function for Lower Operating Power Consumption

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

#### 11.5.1 Setting Operating Power Control Mode

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

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

- Example 1: From high-speed operating mode to middle-speed operating mode

(High-speed operation in high-speed operating mode)



Set the frequency of each clock to lower than the maximum operating frequency for middle-speed operating mode



Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)



Set the OPCCR.OPCM[2:0] bits to 010b (middle-speed operating mode)



Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)



(Middle-speed operation in middle-speed operating mode)



- Example 2: From high-speed/middle-speed operating mode to low-speed operating mode

(High-speed operation in high-speed operating mode/middle-speed operation in middle-speed operating mode)

↓

Set the frequency of each clock to lower than the maximum operating frequency for low-speed operating mode

↓

Confirm that all clock sources but the sub-clock oscillator are stopped

↓

Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)

↓

Set the SOPCCR.SOPCM bit to 1 (low-speed operating mode)

↓

Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)

↓

Low-speed operation in low-speed operating mode

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

- Example 1: From low-speed operating mode to high-speed/middle-speed operating mode

Low-speed operation in low-speed operating mode

↓

Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)

↓

Set the SOPCCR.SOPCM bit to 0 (high-speed operating mode or middle-speed operating mode)

↓

Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)

↓

Set the frequency of each clock to lower than the maximum operating frequency for high-speed/middle-speed operating mode

↓

(High-speed operation in high-speed operating mode/middle-speed operation in middle-speed operating mode)

- Example 2: From middle-speed operating mode to high-speed operating mode

Middle-speed operation in middle-speed operating mode

↓

Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)

↓

Set the OPCCR.OPCM[2:0] bit to 0 (high-speed operating mode)

↓

Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)

↓

Set the frequency of each clock to lower than the maximum operating frequency for high-speed operating mode

↓

High-speed operation in high-speed operating mode

## 11.6 Low Power Consumption Modes

### 11.6.1 Sleep Mode

#### 11.6.1.1 Entry to Sleep Mode

When the WAIT instruction is executed while the SBYCR.SSBY bit is 0, the CPU enters sleep mode. In sleep mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop.

Counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in register start mode and the IWDTCSSTPR.SLCSTP bit is 1.

Furthermore, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low power consumption modes). In the same way, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in register start mode and the IWDTCSSTPR.SLCSTP bit is 0.

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

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

Note 1. For details, refer to section 2, CPU.

Note 2. For details, refer to section 14.4.3, Selecting Interrupt Request Destinations.

Note 3. For details, refer to section 14, Interrupt Controller (ICUb).

### 11.6.1.2 Exit from Sleep Mode

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

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

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

Note 2. For details, refer to section 2, CPU.

### 11.6.1.3 Sleep Mode Return Clock Source Switching Function

To switch the clock source used for exit from sleep mode, set the sleep mode return clock source switching register (RSTCKCR) and the wait control register for each clock. 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 exits sleep mode. At this time, the registers related to clock source switching are automatically rewritten.

For details, refer to section 11.2.8, Sleep Mode Return Clock Source Switching Register (RSTCKCR).

For details on settings the oscillation stabilization wait time, refer to section 9.2.17, Main Clock Oscillator Wait Control Register (MOSCWTCR) and section 9.2.18, High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR).

## 11.6.2 Deep Sleep Mode

### 11.6.2.1 Entry to Deep Sleep Mode

When a WAIT instruction is executed with the MSTPCRC.DSLPE bit set to 1, the MSTPCRA.MSTPA28 bit set to 1, and the SBYCR.SSBY bit cleared to 0, a transition to deep sleep mode is made.

In deep sleep mode, the CPU and the DTC, ROM, and RAM clocks stop. Peripheral functions do not stop.

Counting by the IWDT stops if a transition to deep sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to deep sleep mode is made while the IWDT is being used in register start mode and the IWDTCSSTPR.SLCSTP bit is 1.

Furthermore, counting by the IWDT continues if a transition to deep sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low power consumption modes). In the same way, counting by the IWDT continues if a transition to deep sleep mode is made while the IWDT is being used in register start mode and the IWDTCSSTPR.SLCSTP bit is 0.

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

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

Note 1. For details, refer to section 2, CPU.

Note 2. For details, refer to section 14.4.3, Selecting Interrupt Request Destinations.

Note 3. For details, refer to section 14, Interrupt Controller (ICUb).

### 11.6.2.2 Exit from Deep Sleep Mode

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

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

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

Note 2. For details, refer to section 2, CPU.

### 11.6.3 Software Standby Mode

#### 11.6.3.1 Entry to Software Standby Mode

When a WAIT instruction is executed with the SBYCR.SSBY bit set to 1, a transition to software standby mode is made. In this mode, the CPU, on-chip peripheral functions, and all the other functions except the sub-clock oscillator stop. However, the contents of the CPU internal registers, RAM data, the states of on-chip peripheral functions, the I/O ports, and the sub-clock oscillator are retained. Software standby mode allows significant reduction in power consumption because the oscillator stops in this mode.

Set the DTCST.DTCST bit to 0 before executing the WAIT instruction.

Counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in register start mode and the IWDTCSSTPR.SLCSTP bit is 1.

Furthermore, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low power consumption modes). In the same way, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in register start mode and the IWDTCSSTPR.SLCSTP bit is 0.

To use software standby mode, make the following settings and then execute a WAIT instruction.

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

Note 1. For details, refer to section 2, CPU.

Note 2. For details, refer to section 14.4.3, Selecting Interrupt Request Destinations.

Note 3. For details, refer to section 14, Interrupt Controller (ICUb).

### 11.6.3.2 Exit from Software Standby Mode

Exit from software standby mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ7), peripheral interrupts (the RTC alarm, RTC interval, IWDT, voltage monitoring, and USB interrupts), a RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset. When any trigger which initiates exit from software standby mode is asserted, the oscillators which were operating before entry to software standby mode restart operation. After the oscillation of all these oscillators has been stabilized, operation returns from software standby mode.

- Initiated by an interrupt  
When an interrupt request from among the NMI, IRQ0 to IRQ7, RTC alarm, RTC interval, IWDT, voltage monitoring, and USB interrupts is generated, each of the oscillators which was operating before the transition to software standby mode resumes oscillation. After the oscillation stabilization wait time of each oscillator set by the MOSCWTCR.MSTS[4:0] bits or HOCOWTCR.HSTS[4:0] bits has elapsed, the MCU exits software standby mode and interrupt exception processing starts.
- Initiated by a RES# pin reset  
Clock oscillation starts when the low level is applied to the RES# pin. Clock supply for the MCU starts at the same time. Keep the level on the RES# pin low over the time required for oscillation of the clocks to become stable. Reset exception processing starts when the high level is applied to the RES# pin.
- Initiated by a power-on reset  
A power-on reset asserts a reset to the MCU.  
When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset  
A voltage monitoring reset asserts a reset to the MCU.  
When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by an independent watchdog timer reset  
An internal reset generated by an IWDT underflow asserts a reset to the MCU.  
Note that the independent watchdog timer is stopped in software standby mode due to the register settings (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) in software standby mode. In that case, exit from software standby mode by the independent watchdog timer reset cannot be done.

### 11.6.3.3 Example of Software Standby Mode Application

Figure 11.6 shows an example of entry to software standby mode by the falling edge of the IRQn pin, and exit from software standby mode by the rising edge of the IRQn pin.

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

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

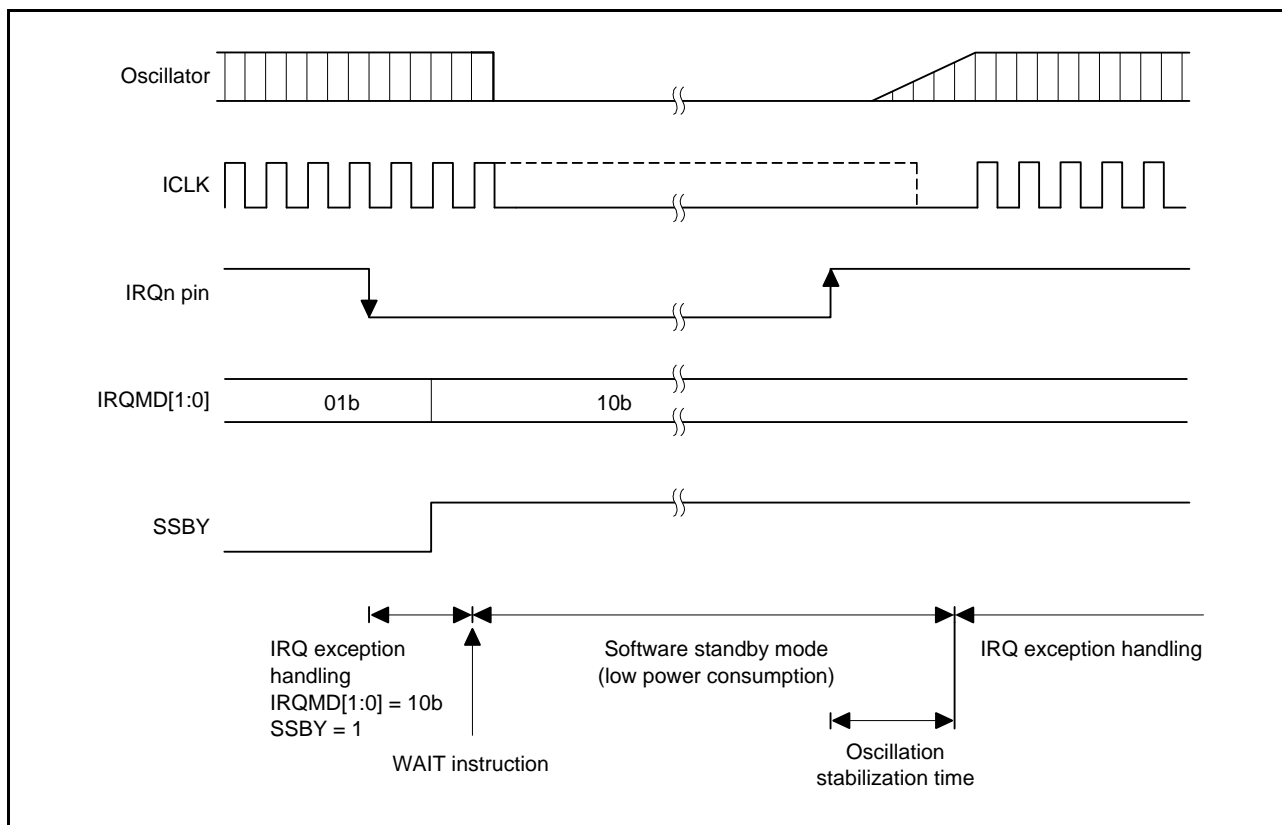


Figure 11.6 Example of Software Standby Mode Application



## 11.7 Usage Notes

### 11.7.1 I/O Port States

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

### 11.7.2 Module Stop State of DTC

Before setting the MSTPCRA.MSTPA28 bit to 1, set the DTCST.DTCST bit of the DTC to 0 to avoid activating the DTC module.

For details, refer to section 16, 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 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, MSTPCRC, and MSTPCRD

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

### 11.7.5 Timing of WAIT Instructions

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

### 11.7.6 Rewrite the Register by DTC in Sleep Mode

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

The RSTCKCR register is a register that switches the clock source at exit from sleep mode. Changing the RSTCKCR register in sleep mode causes unintended operation, so do not write to this register in sleep mode.

## 12. Register Write Protection Function

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

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

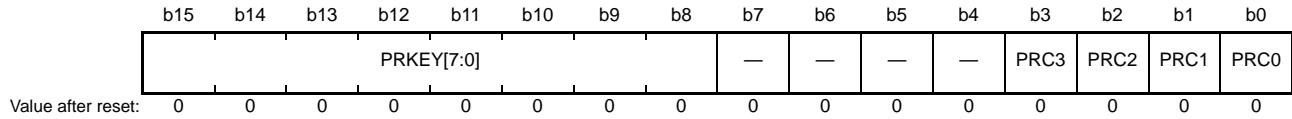
**Table 12.1 Association between PRCR Bits and Registers to be Protected**

PRCR Bit	Register to be Protected
PRC0	<ul style="list-style-type: none"> <li>Registers related to the clock generation circuit: SCKCR, SCKCR3, PLLCR, PLLCR2, MOSCCR, SOSCCR, LOCOCR, ILOCOCR, HOCOGR, OSTDCR, OSTDSR, CKOCR, UPLLCR, UPLLCR2, LCDSCLKCR, LCDSCLKCR2</li> </ul>
PRC1	<ul style="list-style-type: none"> <li>Register related to the operating modes: SYSCR1</li> <li>Registers related to low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, OPCCR, RSTCKCR, SOPCCR</li> <li>Registers related to the clock generation circuit: MOFCR, MOSCWTCR</li> <li>Software reset register: SWRR</li> </ul>
PRC2	<ul style="list-style-type: none"> <li>Register related to the clock generation circuit: HOCOWTCR</li> </ul>
PRC3	<ul style="list-style-type: none"> <li>Registers related to the LVD: LVCMPCR, LVDLVLR, LVD1CR0, LVD1CR1, LVD1SR, LVD2CR0, LVD2CR1, LVD2SR</li> </ul>

## 12.1 Register Descriptions

### 12.1.1 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 functions, the clock generation circuit, and software reset. 0: Write disabled 1: Write enabled	R/W
b2	PRC2	Protect Bit 2	Enables writing to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
b3	PRC3	Protect Bit 3	Enables writing to the registers related to the LVD. 0: Write disabled 1: Write enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	PRC Key Code	These bits control permission and prohibition of writing to the PRCR register. To modify the PRCR register, write A5h to the 8 higher-order bits and the desired value to the 8 lower-order bits as a 16-bit unit.	R/W*1

Note 1. Write data is not retained.

#### PRCi Bits (Protect Bit i) (i = 0 to 3)

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

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

## 13. Exception Handling

### 13.1 Exception Events

During execution of a program by the CPU, the occurrence of a certain event may cause execution of that program to be suspended and execution of another program to be started. Such kinds of events are called exception events.

The RX CPU supports six types of exceptions. The types of exception events are shown in Figure 13.1.

The occurrence of an exception causes the processor mode to switch to supervisor mode.

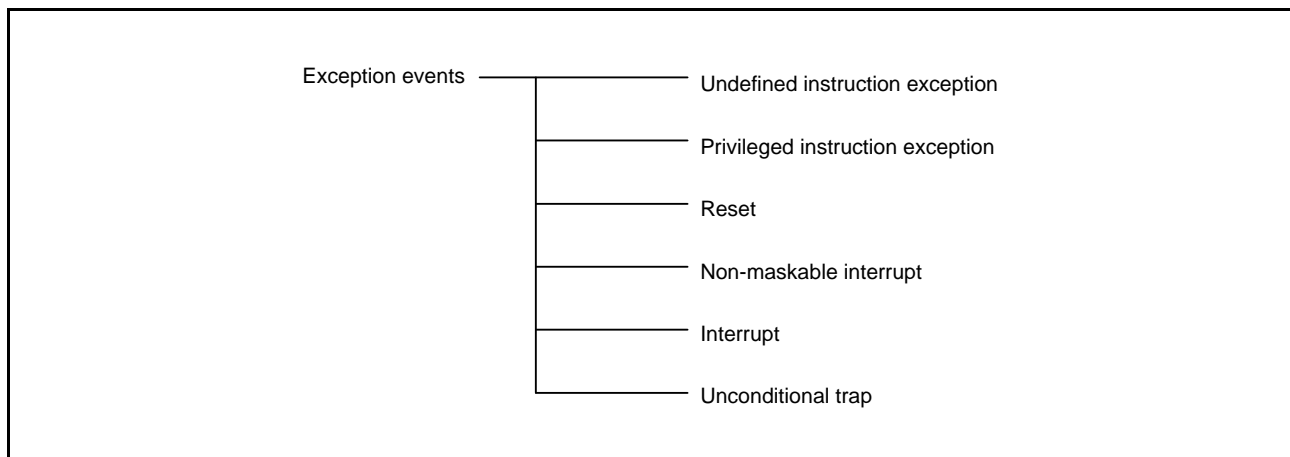


Figure 13.1 Types of Exception Events

### 13.1.1 Undefined Instruction Exception

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

### 13.1.2 Privileged Instruction Exception

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

### 13.1.3 Reset

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

### 13.1.4 Non-Maskable Interrupt

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

### 13.1.5 Interrupts

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

### 13.1.6 Unconditional Trap

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

### 13.2 Exception Handling Procedure

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

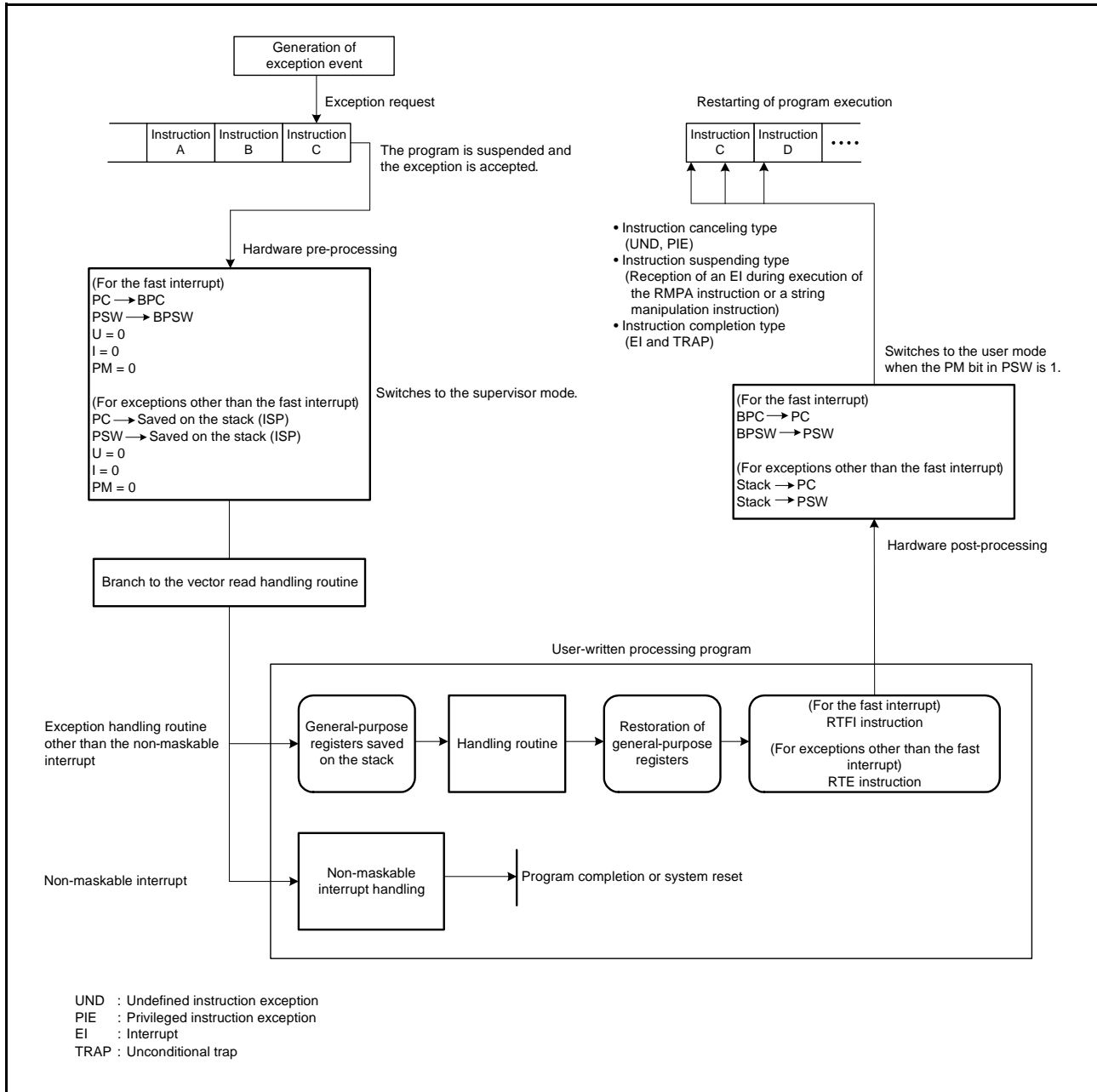


Figure 13.2 Outline of Exception Handling Procedure

When an exception is accepted, hardware processing by the RX CPU is followed by access to the vector to acquire the address of the branch destination. In the vector, a vector address is allocated to each exception, and the branch destination address of the exception handling routine is written to each vector address.

Hardware pre-processing by the RX CPU handles saving of the contents of the program counter (PC) and processor status word (PSW). In the case of a fast interrupt, the contents are saved in the backup PC (BPC) and the backup PSW (BPSW), respectively. In the case of exceptions other than a fast interrupt, the contents are saved in the stack area.

General purpose registers and control registers other than the PC and PSW that are to be used within the exception handling routine must be saved on the stack by a user program at the start of the exception handling routine.

On completion of processing by an exception handling routine, execution is restored from the exception handling routine to the original program by saving the registers saved on the stack and executing the RTE instruction. For return from a fast interrupt, the RTFI instruction is used instead. In the case of a non-maskable interrupt, however, finish the program or reset the system without returning to the original program.

Hardware post-processing by the RX CPU handles restoration of the contents of PC and PSW. In the case of a fast interrupt, the values of BPC and BPSW are restored to PC and PSW, respectively. In the case of exceptions other than a fast interrupt, the values are restored from the stack to PC and PSW.

### 13.3 Acceptance of Exception Events

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

#### 13.3.1 Acceptance Timing and Saved PC Value

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

**Table 13.1 Acceptance Timing and Saved PC Value**

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

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

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

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

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



## 13.4 Hardware Processing for Accepting and Returning from Exceptions

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

### (1) Hardware Pre-Processing for Accepting an Exception

#### (a) Saving PSW

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

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

I: Set to 0

U: Set to 0

PM: Set to 0

#### (c) Saving PC

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

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

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

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

#### (a) Restoring PSW

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

#### (b) Restoring PC

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

## 13.5 Hardware Pre-Processing

The hardware pre-processing from reception of each exception request to execution of the associated exception handling routine are explained below.

### 13.5.1 Undefined Instruction Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFDCh.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

### 13.5.2 Privileged Instruction Exception

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFD0h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

### 13.5.3 Reset

1. The control registers are initialized.
2. The vector is fetched from address FFFF FFFCh.
3. The fetched vector is set to the PC.

### 13.5.4 Non-Maskable Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved on the stack (ISP). For other instructions, the PC value of the next instruction is saved.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW are set to Fh.
5. The vector is fetched from address FFFF FFF8h.
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

### 13.5.5 Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP) or, for the fast interrupt, in the backup PSW (BPSW).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved. For other instructions, the PC value of the next instruction is saved. Saving of the PC is in the backup PC (BPC) for fast interrupts.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW indicate the interrupt priority level of the interrupt.
5. The vector for an interrupt source other than the fast interrupt is fetched from the relocatable vector table. For the fast interrupt, the address is fetched from the fast interrupt vector register (FINTV).
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

### 13.5.6 Unconditional Trap

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) for the next instruction is saved on the stack (ISP).
4. For the INT instruction, the value at the vector corresponding to the INT instruction number is fetched from the relocatable vector table.  
For the BRK instruction, the value at the vector from the start address is fetched from the relocatable vector table.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

### 13.6 Return from Exception Handling Routine

Executing the instruction listed in Table 13.3 at the end of the corresponding exception handling routine restores the values of the program counter (PC) and processor status word (PSW) that were saved on the stack or in the control registers (BPC and BPSW) immediately before the exception handling sequence.


**Table 13.3 Return from Exception Handling Routine**

Exception		Instruction for Return
Undefined instruction exception		RTE
Privileged instruction exception		RTE
Reset		Return is impossible
Non-maskable interrupt		Return is impossible
Interrupt	Fast interrupt	RTFI
	Other than above	RTE
Unconditional trap		RTE

### 13.7 Priority of Exception Events

The priority of exception events is listed in Table 13.4. When multiple exceptions are generated at the same time, the exception with the highest priority is accepted first.

**Table 13.4 Priority of Exception Events**

Priority	Exception Event
High  Low	1 Reset
	2 Non-maskable interrupt
	3 Interrupt
	4 Undefined instruction exception Privileged instruction exception
	5 Unconditional trap

## 14. Interrupt Controller (ICUb)

In this section, “PCLK” is used to refer to PCLKB.

### 14.1 Overview

The interrupt controller receives interrupt signals from peripheral modules and external pins, sends interrupts to the CPU, and activates the DTC.

Table 14.1 lists the specifications of the interrupt controller, and Figure 14.1 shows a block diagram of the interrupt controller.

**Table 14.1 Specifications of Interrupt Controller**

Item	Description
Interrupts	Peripheral function interrupts <ul style="list-style-type: none"> <li>• Interrupts from peripheral modules</li> <li>• Interrupt detection: Edge detection/level detection</li> </ul> Edge detection or level detection is fixed for each source of connected peripheral modules.
	External pin interrupts <ul style="list-style-type: none"> <li>• Interrupts from pins IRQ0 to IRQ7</li> <li>• Number of sources: 8</li> <li>• Interrupt detection: Low/falling edge/rising edge/rising and falling edges</li> <li>• 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>
	Event link interrupt <ul style="list-style-type: none"> <li>• The ELSR18I interrupt is generated by an ELC event</li> </ul>
	Interrupt priority <ul style="list-style-type: none"> <li>• Specified by registers.</li> </ul>
	Fast interrupt function <ul style="list-style-type: none"> <li>• Faster interrupt handling of the CPU can be set only for a single interrupt source.</li> </ul>
	DTC control <ul style="list-style-type: none"> <li>• The DTC can be activated by interrupt sources.*1</li> </ul>
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 <ul style="list-style-type: none"> <li>• Interrupt on detection of oscillation having stopped</li> </ul>
	IWDT underflow/refresh error <ul style="list-style-type: none"> <li>• Interrupt on an underflow of the down counter or occurrence of a refresh error</li> </ul>
	Voltage monitoring 1 interrupt <ul style="list-style-type: none"> <li>• Voltage monitoring interrupt of voltage monitoring circuit 1 (LVD1)</li> </ul>
	Voltage monitoring 2 interrupt <ul style="list-style-type: none"> <li>• Voltage monitoring interrupt of voltage monitoring circuit 2 (LVD2)</li> </ul>
Return from power-down modes <ul style="list-style-type: none"> <li>• Sleep mode, deep sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source.</li> <li>• Software standby mode: Return is initiated by non-maskable interrupts, IRQ0 to IRQ7 interrupts, or RTC alarm/periodic interrupts.</li> </ul>	

Note 1. For the DTC activation source, see Table 14.3, Interrupt Vector Table.

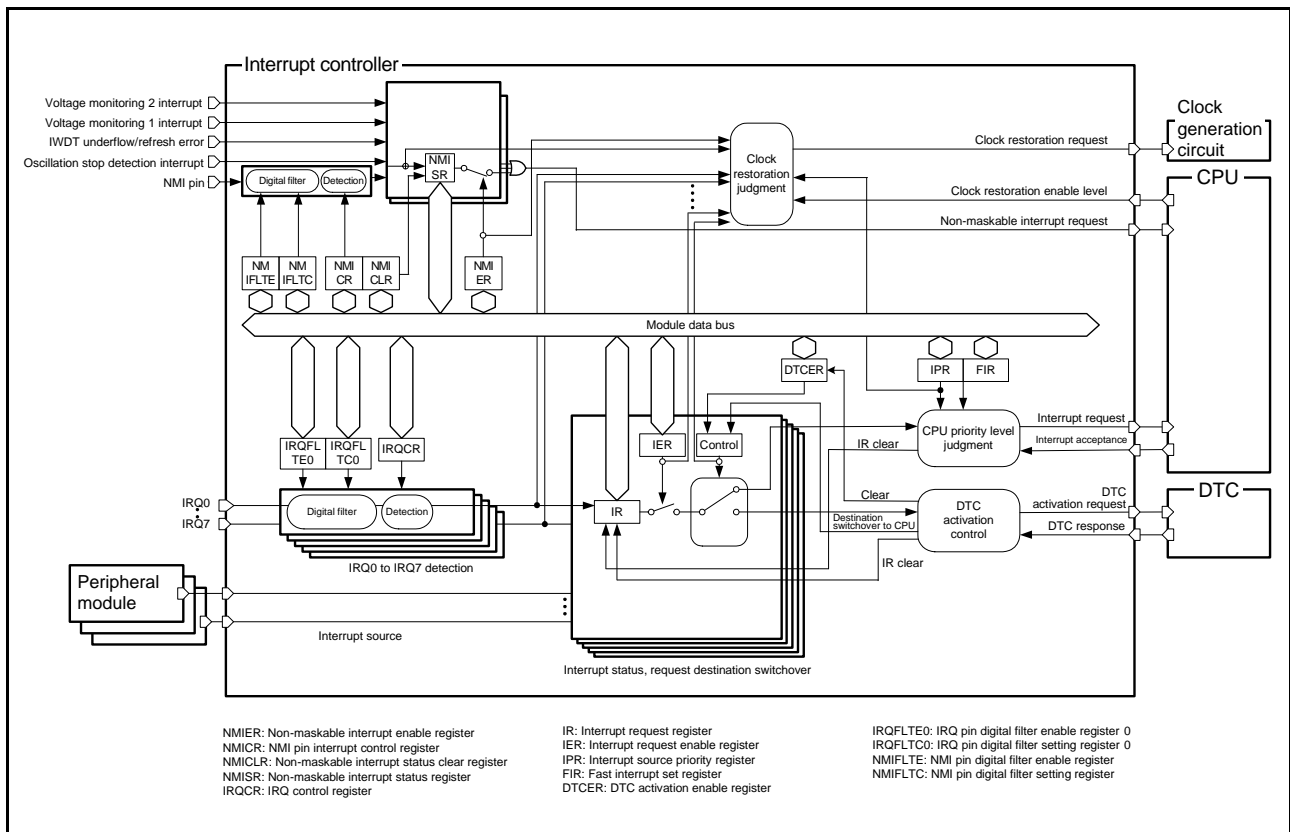


Figure 14.1 Block Diagram of Interrupt Controller

Table 14.2 lists the I/O pins of the interrupt controller.

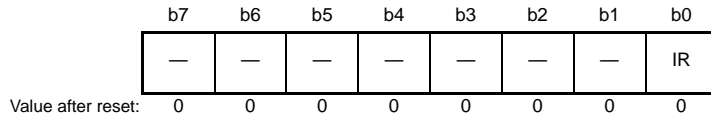
Table 14.2 Pin Configuration of Interrupt Controller

Pin Name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ7	Input	External interrupt request pins

## 14.2 Register Descriptions

### 14.2.1 Interrupt Request Register n (IRn) (n = interrupt vector number)

Address(es): 0008 7010h to 0008 70F9h



Bit	Symbol	Bit Name	Description	R/W
b0	IR	Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. For an edge detection interrupt, only 0 can be written to this bit; do not write 1.  
For a level detection interrupt, neither 0 nor 1 can be written.

IRn is provided for each interrupt source, where “n” indicates the interrupt vector number.

For the correspondence between interrupt sources and interrupt vector numbers, see Table 14.3, Interrupt Vector Table.

#### IR Flag (Interrupt Status Flag)

This 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 for each interrupt source. For interrupts from IRQi pins, edge detection or level detection is selected by setting the corresponding IRQCRi.IRQMD[1:0] bits (i = 0 to 7). For detection of the various interrupt sources, see Table 14.3, Interrupt Vector Table.

#### (1) Edge detection

[Setting condition]

- The flag is set to 1 in response to the generation of an interrupt request from the corresponding peripheral module or IRQi pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.

[Clearing conditions]

- The flag is set to 0 when the interrupt request destination accepts the interrupt request.
- The IR flag is set to 0 by writing 0 to it. Note, however, that writing 0 to the IR flag is prohibited if the destination of the interrupt request is the DTC.

#### (2) Level detection

[Setting condition]

- The flag remains set to 1 while an interrupt request is being sent from the corresponding peripheral module or IRQi pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.

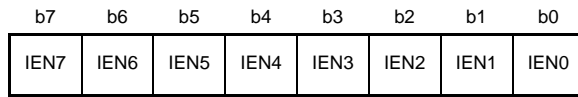
[Clearing condition]

- The flag is set to 0 when the source of the interrupt request is cleared (it is not cleared when the interrupt request destination accepts the interrupt request). For clearing interrupts from the various peripheral modules, refer to the sections describing the modules.

When level detection has been selected for an IRQi pin, the interrupt request is withdrawn by driving the IRQi pin high. Do not write 0 or 1 to the IR flag while level detection is selected.

## 14.2.2 Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh)

Address(es): 0008 7202h to 0008 721Fh



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

Bit	Symbol	Bit Name	Description	R/W
b0	IEN0	Interrupt Request Enable 0	0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
b1	IEN1	Interrupt Request Enable 1		R/W
b2	IEN2	Interrupt Request Enable 2		R/W
b3	IEN3	Interrupt Request Enable 3		R/W
b4	IEN4	Interrupt Request Enable 4		R/W
b5	IEN5	Interrupt Request Enable 5		R/W
b6	IEN6	Interrupt Request Enable 6		R/W
b7	IEN7	Interrupt Request Enable 7		R/W

Note: Write 0 to the bit that corresponds to the vector number for reservation. These bits are read as 0.

### IENj Bit (Interrupt Request Enable j) (j = 0 to 7)

When an IENj bit is 1, the corresponding interrupt request will be output to the destination selected for the request. When an IENj bit is 0, the corresponding interrupt request will not be output to the destination selected for the request. The setting of an IENj bit does not affect the IRn.IR flag. Even if the corresponding IENj bit is 0, the IR flag value changes according to the descriptions in section 14.2.1, Interrupt Request Register n (IRn) (n = interrupt vector number).

The IERm.IENj bit is set for each request source (vector number).

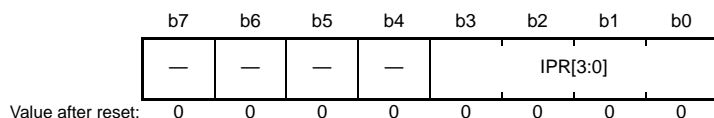
For the correspondence between interrupt sources and IERm.IENj bit, see Table 14.3, Interrupt Vector Table.

For the procedure for setting IERm.IENj bit during the selection of destinations for interrupt requests, refer to section 14.4.3, Selecting Interrupt Request Destinations.



### 14.2.3 Interrupt Source Priority Register n (IPRn) (n = 000 to 249)

Address(es): 0008 7300h to 0008 73F9h



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</td> <td>0</td> <td>: Level 0 (interrupt disabled)*1</td> </tr> <tr> <td>0 0 0</td> <td>1</td> <td>: Level 1</td> </tr> <tr> <td>0 0 1</td> <td>0</td> <td>: Level 2</td> </tr> <tr> <td>0 0 1</td> <td>1</td> <td>: Level 3</td> </tr> <tr> <td>0 1 0</td> <td>0</td> <td>: Level 4</td> </tr> <tr> <td>0 1 0</td> <td>1</td> <td>: Level 5</td> </tr> <tr> <td>0 1 1</td> <td>0</td> <td>: Level 6</td> </tr> <tr> <td>0 1 1</td> <td>1</td> <td>: Level 7</td> </tr> <tr> <td>1 0 0</td> <td>0</td> <td>: Level 8</td> </tr> <tr> <td>1 0 0</td> <td>1</td> <td>: Level 9</td> </tr> <tr> <td>1 0 1</td> <td>0</td> <td>: Level 10</td> </tr> <tr> <td>1 0 1</td> <td>1</td> <td>: Level 11</td> </tr> <tr> <td>1 1 0</td> <td>0</td> <td>: Level 12</td> </tr> <tr> <td>1 1 0</td> <td>1</td> <td>: Level 13</td> </tr> <tr> <td>1 1 1</td> <td>0</td> <td>: Level 14</td> </tr> <tr> <td>1 1 1</td> <td>1</td> <td>: Level 15 (highest)</td> </tr> </table>	b3	b0		0 0 0	0	: Level 0 (interrupt disabled)*1	0 0 0	1	: Level 1	0 0 1	0	: Level 2	0 0 1	1	: Level 3	0 1 0	0	: Level 4	0 1 0	1	: Level 5	0 1 1	0	: Level 6	0 1 1	1	: Level 7	1 0 0	0	: Level 8	1 0 0	1	: Level 9	1 0 1	0	: Level 10	1 0 1	1	: Level 11	1 1 0	0	: Level 12	1 1 0	1	: Level 13	1 1 1	0	: Level 14	1 1 1	1	: Level 15 (highest)	R/W
b3	b0																																																						
0 0 0	0	: Level 0 (interrupt disabled)*1																																																					
0 0 0	1	: Level 1																																																					
0 0 1	0	: Level 2																																																					
0 0 1	1	: Level 3																																																					
0 1 0	0	: Level 4																																																					
0 1 0	1	: Level 5																																																					
0 1 1	0	: Level 6																																																					
0 1 1	1	: Level 7																																																					
1 0 0	0	: Level 8																																																					
1 0 0	1	: Level 9																																																					
1 0 1	0	: Level 10																																																					
1 0 1	1	: Level 11																																																					
1 1 0	0	: Level 12																																																					
1 1 0	1	: Level 13																																																					
1 1 1	0	: Level 14																																																					
1 1 1	1	: Level 15 (highest)																																																					
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 register, see Table 14.3, Interrupt Vector Table.

#### IPR[3:0] Bits (Interrupt Priority Level Select)

These bits specify the priority level of the corresponding interrupt source.

Priority levels specified by the IPR[3:0] bits are used only to determine the priority of interrupt requests to be transferred to the CPU, and do not affect activation requests to the DTC.

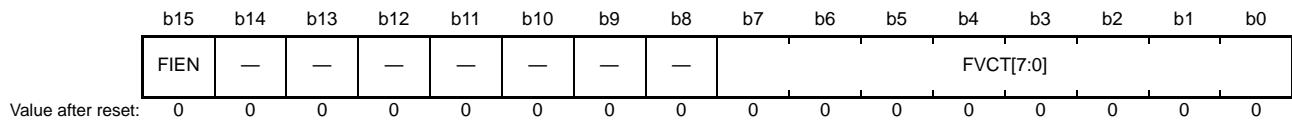
The CPU accepts only interrupt requests higher than the priority level specified by the PSW.IPL[3:0] bits, and handles accepted interrupts.

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).

## 14.2.4 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.

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, refer to section 14.6.2, Return from Software Standby Mode.

If the setting of the IERm.IENj (m = 02h to 1Fh; j = 0 to 7) bit has disabled interrupt requests from the interrupt source with the vector number in this register, fast interrupt requests are not output to the CPU.

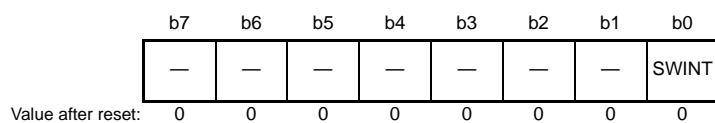
For settable vector numbers, see Table 14.3, Interrupt Vector Table.

Do not write any reserved vector numbers to the FVCT[7:0] bits.

For details on the fast interrupt, refer to section 13, Exception Handling, and section 14.4.6, Fast Interrupt.

### 14.2.5 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)

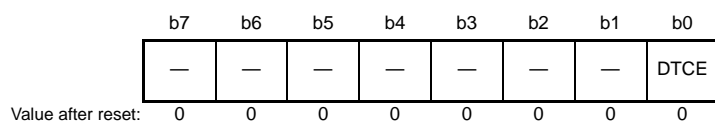
When 1 is written to the SWINT bit, interrupt request register 027 (IR027) is set to 1.

If 1 is written to the SWINT bit when 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 DTC activation enable register 027 (DTCER027) is set to 1, a DTC activation request is issued.

### 14.2.6 DTC Activation Enable Register n (DTCERn) (n = interrupt vector number)

Address(es): 0008 711Bh to 0008 71F8h



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

See Table 14.3, 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]

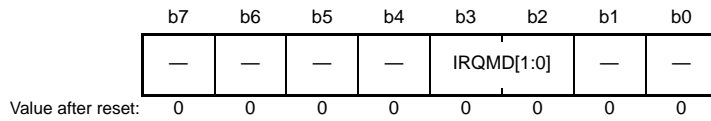
- When 1 is written to the DTCE bit

[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

### 14.2.7 IRQ Control Register i (IRQCRi) (i = 0 to 7)

Address(es): 0008 7500h to 0008 7507h



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	<div style="display: flex; align-items: flex-start;"> <div style="margin-right: 5px;">b3</div> <div style="margin-right: 5px;">b2</div> <div>           0 0: Low level            0 1: Falling edge            1 0: Rising edge            1 1: Rising and falling edges         </div> </div>	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 (IERm.IENj bit is 0). After changing the setting, clear the IR flag 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 IRQ7.

For the external pin interrupt detection setting, refer to section 14.4.8, External Pin Interrupts.

### 14.2.8 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 disabled 1: Digital filter 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> Bit (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.

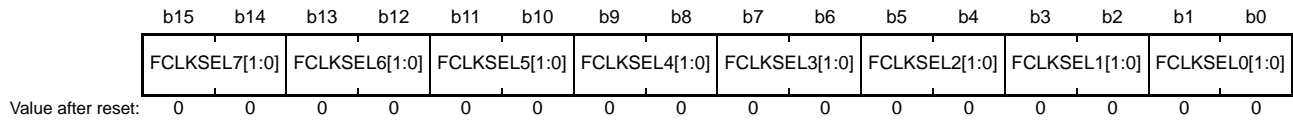
The digital filter is enabled when the FLTEN<sub>i</sub> bit is 1, and disabled when the FLTEN<sub>i</sub> bit is 0.

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, refer to section 14.4.7, Digital Filter.

### 14.2.9 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), PCLK/8 (once every eight cycles), PCLK/32 (once every 32 cycles), and PCLK/64 (once every 64 cycles).

For details of the digital filter, refer to section 14.4.7, Digital Filter.

### 14.2.10 Non-Maskable Interrupt Status Register (NMISR)

Address(es): 0008 7580h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2S T	LVD1S T	IWDTS T	—	OSTST	NMIST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMIST	NMI Status Flag	0: NMI pin interrupt is not requested 1: NMI pin interrupt is requested	R
b1	OSTST	Oscillation Stop Detection Interrupt Status Flag	0: Oscillation stop detection interrupt is not requested 1: Oscillation stop detection interrupt is requested	R
b2	—	Reserved	This bit is read as 0 and cannot be modified.	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



**IWDTST Flag (IWDT Underflow/Refresh Error Status Flag)**

This flag indicates the IWDT underflow/refresh error interrupt request.

The IWDTST flag is read-only, and cleared by the NMICLR.IWDTCLR bit.

[Setting condition]

- When the IWDT underflow/refresh error interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.IWDTCLR bit

**LVD1ST Flag (Voltage Monitoring 1 Interrupt Status Flag)**

This flag indicates the request for voltage monitoring 1 interrupt.

The LVD1ST flag is read-only, and cleared by the NMICLR.LVD1CLR bit.

[Setting condition]

- When the voltage monitoring 1 interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.LVD1CLR bit

**LVD2ST Flag (Voltage Monitoring 2 Interrupt Status Flag)**

This flag indicates the request for voltage monitoring 2 interrupt.

The LVD2ST flag is read-only, and cleared by the NMICLR.LVD2CLR bit.

[Setting condition]

- When the voltage monitoring 2 interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.LVD2CLR bit

### 14.2.11 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): 0008 7581h

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

Bit	Symbol	Bit Name	Description	R/W
b0	NMIEN	NMI Pin Interrupt Enable	0: NMI pin interrupt is disabled 1: NMI pin interrupt is enabled	R/(W) *1
b1	OSTEN	Oscillation Stop Detection Interrupt Enable	0: Oscillation stop detection interrupt is disabled 1: Oscillation stop detection interrupt is enabled	R/(W) *1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/(W)
b3	IWDTEN	IWDT Underflow/Refresh Error Enable	0: IWDT underflow/refresh error interrupt is disabled 1: IWDT underflow/refresh error interrupt is enabled	R/(W) *1
b4	LVD1EN	Voltage Monitoring 1 Interrupt Enable	0: Voltage monitoring 1 interrupt is disabled 1: Voltage monitoring 1 interrupt is enabled	R/(W) *1
b5	LVD2EN	Voltage Monitoring 2 Interrupt Enable	0: Voltage monitoring 2 interrupt is disabled 1: Voltage monitoring 2 interrupt is enabled	R/(W) *1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

#### NMIEN Bit (NMI Pin Interrupt Enable)

This bit enables the NMI pin interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

#### OSTEN Bit (Oscillation Stop Detection Interrupt Enable)

This bit enables the oscillation stop detection interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

#### IWDTEN Bit (IWDT Underflow/Refresh Error Enable)

This bit enables the IWDT underflow/refresh error interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

#### LVD1EN Bit (Voltage Monitoring 1 Interrupt Enable)

This bit enables the voltage monitoring 1 interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

#### LVD2EN Bit (Voltage Monitoring 2 Interrupt Enable)

This bit enables the voltage monitoring 2 interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

## 14.2.12 Non-Maskable Interrupt Status Clear Register (NMICLR)

Address(es): 0008 7582h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2C LR	LVD1C LR	IWDTC LR	—	OSTCL R	NMICL R
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMICLR	NMI Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.NMIST flag. Writing 0 to this bit has no effect.	R/(W) *1
b1	OSTCLR	OST Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.OSTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/(W)
b3	IWDTCLR	IWDT Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.IWDTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b4	LVD1CLR	LVD1 Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.LVD1ST flag. Writing 0 to this bit has no effect.	R/(W) *1
b5	LVD2CLR	LVD2 Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.LVD2ST flag. Writing 0 to this bit has no effect.	R/(W) *1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written to this bit.

### NMICLR Bit (NMI Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. This bit is read as 0.

### OSTCLR Bit (OST Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. This bit is read as 0.

### IWDTCLR Bit (IWDT Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. This bit is read as 0.

### LVD1CLR Bit (LVD1 Clear)

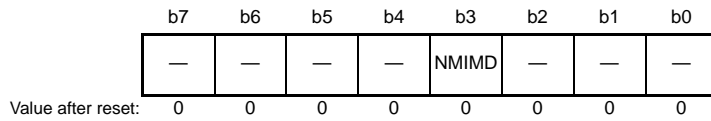
Writing 1 to the LVD1CLR bit clears the NMISR.LVD1ST flag. This bit is read as 0.

### LVD2CLR Bit (LVD2 Clear)

Writing 1 to the LVD2CLR bit clears the NMISR.LVD2ST flag. This bit is read as 0.

### 14.2.13 NMI Pin Interrupt Control Register (NMICR)

Address(es): 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

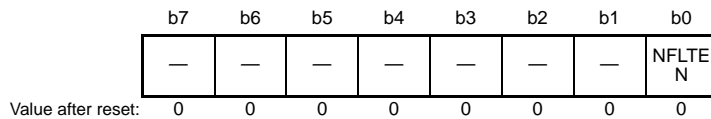
Change the setting of the NMICR register before the NMI pin interrupt is enabled (before setting the NMIER.NMIEN bit to 1).

#### NMIMD Bit (NMI Detection Set)

This bit specifies the detection edge of the NMI pin interrupt.

### 14.2.14 NMI Pin Digital Filter Enable Register (NMIFLTE)

Address(es): 0008 7590h



Bit	Symbol	Bit Name	Description	R/W
b0	NFLTEN	NMI Digital Filter Enable	0: Digital filter disabled 1: Digital filter 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.

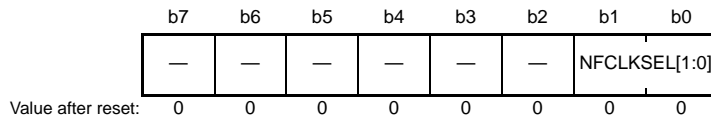
The digital filter is enabled when the NFLTEN bit is 1, and disabled when the NFLTEN bit is 0.

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, refer to section 14.4.7, Digital Filter.

### 14.2.15 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), PCLK/8 (once every eight cycles), PCLK/32 (once every 32 cycles), and PCLK/64 (once every 64 cycles).

For details of the digital filter, refer to section 14.4.7, Digital Filter.

## 14.3 Vector Table

There are two types of interrupts detected by the interrupt controller: maskable interrupts and non-maskable interrupts. When the CPU accepts an interrupt or non-maskable interrupt, it acquires a 4-byte vector address from the vector table.

### 14.3.1 Interrupt Vector Table

The interrupt vector table is placed in the 1024-byte range (4 bytes × 256 sources) beginning at the address specified in the interrupt table register (INTB) of the CPU. Write a value to the INTB register before enabling interrupts. The value written to the INTB register should be a multiple of 4.

Executing an INT instruction or BRK instruction leads to the generation of an unconditional trap. The same range of memory as shown in Table 14.3, Interrupt Vector Table, is used for the vectors for unconditional traps. The vector for BRK instructions is vector 0 while the vector numbers for INT instructions are specifiable as numbers in the range from 0 to 255.

Table 14.3 lists details of the interrupt vectors. Details of the headings in Table 14.3 are listed below.

Item	Description
Source of interrupt request generation	Name of the source for generation of the interrupt request
Name	Name of the interrupt
Vector no.	Vector number for the interrupt
Vector address offset	Value of the offset from the base address for the vector table
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.
ssbt return	"o" in this column indicates usability as a request for return from software standby 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 14.3 Interrupt Vector Table (1/7)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	sstb Return	IER	IPR	DTCER
—	For an unconditional trap	0	0000h	—	x	x	x	—	—	—
—	For an unconditional trap	1	0004h	—	x	x	x	—	—	—
—	For an unconditional trap	2	0008h	—	x	x	x	—	—	—
—	For an unconditional trap	3	000Ch	—	x	x	x	—	—	—
—	For an unconditional trap	4	0010h	—	x	x	x	—	—	—
—	For an unconditional trap	5	0014h	—	x	x	x	—	—	—
—	For an unconditional trap	6	0018h	—	x	x	x	—	—	—
—	For an unconditional trap	7	001Ch	—	x	x	x	—	—	—
—	For an unconditional trap	8	0020h	—	x	x	x	—	—	—
—	For an unconditional trap	9	0024h	—	x	x	x	—	—	—
—	For an unconditional trap	10	0028h	—	x	x	x	—	—	—
—	For an unconditional trap	11	002Ch	—	x	x	x	—	—	—
—	For an unconditional trap	12	0030h	—	x	x	x	—	—	—
—	For an unconditional trap	13	0034h	—	x	x	x	—	—	—
—	For an unconditional trap	14	0038h	—	x	x	x	—	—	—
—	For an unconditional trap	15	003Ch	—	x	x	x	—	—	—
BSC	BUSERR	16	0040h	Level	o	x	x	IER02.IEN0	IPR000	—
—	Reserved	17	0044h	—	x	x	x	—	—	—
—	Reserved	18	0048h	—	x	x	x	—	—	—
—	Reserved	19	004Ch	—	x	x	x	—	—	—
—	Reserved	20	0050h	—	x	x	x	—	—	—

Table 14.3 Interrupt Vector Table (2/7)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	sstb Return	IER	IPR	DTCER
—	Reserved	21	0054h	—	x	x	x	—	—	—
—	Reserved	22	0058h	—	x	x	x	—	—	—
—	Reserved	23	005Ch	—	x	x	x	—	—	—
—	Reserved	24	0060h	—	x	x	x	—	—	—
—	Reserved	25	0064h	—	x	x	x	—	—	—
—	Reserved	26	0068h	—	x	x	x	—	—	—
ICU	SWINT	27	006Ch	Edge	o	o	x	IER03.IEN3	IPR003	DTCER027
CMT0	CMI0	28	0070h	Edge	o	o	x	IER03.IEN4	IPR004	DTCER028
CMT1	CMI1	29	0074h	Edge	o	o	x	IER03.IEN5	IPR005	DTCER029
CMT2	CMI2	30	0078h	Edge	o	o	x	IER03.IEN6	IPR006	DTCER030
CMT3	CMI3	31	007Ch	Edge	o	o	x	IER03.IEN7	IPR007	DTCER031
CAC	FERRF	32	0080h	Level	o	x	x	IER04.IEN0	IPR032	—
	MENDF	33	0084h	Level	o	x	x	IER04.IEN1	IPR033	—
	OVFF	34	0088h	Level	o	x	x	IER04.IEN2	IPR034	—
—	Reserved	35	008Ch	—	x	x	x	—	—	—
USB0	D0FIFO0	36	0090h	Edge	o	o	x	IER04.IEN4	IPR036	DTCER036
	D1FIFO0	37	0094h	Edge	o	o	x	IER04.IEN5	IPR037	DTCER037
	USBIO	38	0098h	Edge	o	x	x	IER04.IEN6	IPR038	—
—	Reserved	39	009Ch	—	x	x	x	—	—	—
—	Reserved	40	00A0h	—	x	x	x	—	—	—
—	Reserved	41	00A4h	—	x	x	x	—	—	—
—	Reserved	42	00A8h	—	x	x	x	—	—	—
—	Reserved	43	00ACh	—	x	x	x	—	—	—
RSPIO	SPEI0	44	00B0h	Level	o	x	x	IER05.IEN4	IPR044	—
	SPRI0	45	00B4h	Edge	o	o	x	IER05.IEN5		DTCER045
	SPTI0	46	00B8h	Edge	o	o	x	IER05.IEN6		DTCER046
	SPII0	47	00BCh	Level	o	x	x	IER05.IEN7		—
—	Reserved	48	00D0h	—	x	x	x	—	—	—
—	Reserved	49	00D4h	—	x	x	x	—	—	—
—	Reserved	50	00D8h	—	x	x	x	—	—	—
—	Reserved	51	00DCh	—	x	x	x	—	—	—
—	Reserved	52	00D0h	—	x	x	x	—	—	—
—	Reserved	53	00D4h	—	x	x	x	—	—	—
—	Reserved	54	00D8h	—	x	x	x	—	—	—
—	Reserved	55	00DCh	—	x	x	x	—	—	—
—	Reserved	56	00E0h	—	x	x	x	—	—	—
DOC	DOPCF	57	00E4h	Level	o	x	x	IER07.IEN1	IPR057	—
CMPB	CMPB0	58	00E8h	Edge	o	o	x	IER07.IEN2	IPR058	DTCER058
	CMPB1	59	00ECh	Edge	o	o	x	IER07.IEN3	IPR059	DTCER059
CTSU	CTSUWR	60	00F0h	Edge	o	o	x	IER07.IEN4	IPR060	DTCER060
	CTSURD	61	00F4h	Edge	o	o	x	IER07.IEN5		DTCER061
	CTSUFN	62	00F8h	Edge	o	x	x	IER07.IEN6		—
RTC	CUP	63	00FCh	Edge	o	x	x	IER07.IEN7	IPR063	—



Table 14.3 Interrupt Vector Table (3/7)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	sstb Return	IER	IPR	DTCER
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
—	Reserved	72	0120h	—	×	×	×	—	—	—
—	Reserved	73	0124h	—	×	×	×	—	—	—
—	Reserved	74	0128h	—	×	×	×	—	—	—
—	Reserved	75	012Ch	—	×	×	×	—	—	—
—	Reserved	76	0130h	—	×	×	×	—	—	—
—	Reserved	77	0134h	—	×	×	×	—	—	—
—	Reserved	78	0138h	—	×	×	×	—	—	—
—	Reserved	79	013Ch	—	×	×	×	—	—	—
—	Reserved	80	0140h	—	×	×	×	—	—	—
—	Reserved	81	0144h	—	×	×	×	—	—	—
—	Reserved	82	0148h	—	×	×	×	—	—	—
—	Reserved	83	014Ch	—	×	×	×	—	—	—
—	Reserved	84	0150h	—	×	×	×	—	—	—
—	Reserved	85	0154h	—	×	×	×	—	—	—
—	Reserved	86	0158h	—	×	×	×	—	—	—
—	Reserved	87	015Ch	—	×	×	×	—	—	—
LVD	LVD1	88	0160h	Edge	○	×	○	IER0B.IEN0	IPR088	—
	LVD2	89	0164h	Edge	○	×	○	IER0B.IEN1	IPR089	—
USB0	USB0	90	0168h	Level	○	×	○	IER0B.IEN2	IPR090	—
—	Reserved	91	016Ch	—	×	×	×	—	—	—
RTC	ALM	92	0170h	Edge	○	×	○	IER0B.IEN4	IPR092	—
	PRD	93	0174h	Edge	○	×	○	IER0B.IEN5	IPR093	—
—	Reserved	94	0178h	—	×	×	×	—	—	—
—	Reserved	95	017Ch	—	×	×	×	—	—	—
—	Reserved	96	0180h	—	×	×	×	—	—	—
—	Reserved	97	0184h	—	×	×	×	—	—	—
—	Reserved	98	0188h	—	×	×	×	—	—	—
—	Reserved	99	018Ch	—	×	×	×	—	—	—
—	Reserved	100	0190h	—	×	×	×	—	—	—
—	Reserved	101	0194h	—	×	×	×	—	—	—
S12AD	S12ADI0	102	0198h	Edge	○	○	×	IER0C.IEN6	IPR102	DTCER102
	GBADI	103	019Ch	Edge	○	○	×	IER0C.IEN7	IPR103	DTCER103
—	Reserved	104	01A0h	—	×	×	×	—	—	—
—	Reserved	105	01A4h	—	×	×	×	—	—	—
ELC	ELSR18I	106	01A8h	Edge	○	○	×	IER0D.IEN2	IPR106	DTCER106

Table 14.3 Interrupt Vector Table (4/7)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	sstb Return	IER	IPR	DTCER
—	Reserved	107	01ACh	—	x	x	x	—	—	—
SSI0	SSIF0	108	01B0h	Level	o	x	x	IER0D.IEN4	IPR108	—
	SSIRX10	109	01B4h	Edge	o	o	x	IER0D.IEN5		DTCER109
	SSITX10	110	01B8h	Edge	o	o	x	IER0D.IEN6		DTCER110
—	Reserved	111	01BCCh	—	x	x	x	—	—	—
—	Reserved	112	01C0h	—	x	x	x	—	—	—
—	Reserved	113	01C4h	—	x	x	x	—	—	—
MTU0	TGIA0	114	01C8h	Edge	o	o	x	IER0E.IEN2	IPR114	DTCER114
	TGIB0	115	01CCh	Edge	o	o	x	IER0E.IEN3		DTCER115
	TGIC0	116	01D0h	Edge	o	o	x	IER0E.IEN4		DTCER116
	TGID0	117	01D4h	Edge	o	o	x	IER0E.IEN5		DTCER117
	TCIV0	118	01D8h	Edge	o	x	x	IER0E.IEN6	IPR118	—
	TGIE0	119	01DCh	Edge	o	x	x	IER0E.IEN7		—
	TGIF0	120	01E0h	Edge	o	x	x	IER0F.IEN0		—
MTU1	TGIA1	121	01E4h	Edge	o	o	x	IER0F.IEN1	IPR121	DTCER121
	TGIB1	122	01E8h	Edge	o	o	x	IER0F.IEN2		DTCER122
	TCIV1	123	01ECh	Edge	o	x	x	IER0F.IEN3	IPR123	—
	TCIU1	124	01F0h	Edge	o	x	x	IER0F.IEN4		—
MTU2	TGIA2	125	01F4h	Edge	o	o	x	IER0F.IEN5	IPR125	DTCER125
	TGIB2	126	01F8h	Edge	o	o	x	IER0F.IEN6		DTCER126
	TCIV2	127	01FCh	Edge	o	x	x	IER0F.IEN7	IPR127	—
	TCIU2	128	0200h	Edge	o	x	x	IER10.IEN0		—
—	Reserved	129	0204h	—	x	x	x	—	—	—
—	Reserved	130	0208h	—	x	x	x	—	—	—
—	Reserved	131	020Ch	—	x	x	x	—	—	—
—	Reserved	132	0210h	—	x	x	x	—	—	—
—	Reserved	133	0214h	—	x	x	x	—	—	—
—	Reserved	134	0218h	—	x	x	x	—	—	—
—	Reserved	135	021Ch	—	x	x	x	—	—	—
—	Reserved	136	0220h	—	x	x	x	—	—	—
—	Reserved	137	0224h	—	x	x	x	—	—	—
—	Reserved	138	0228h	—	x	x	x	—	—	—
MTU3	TGIA3	129	0204h	Edge	o	o	x	IER10.IEN1	IPR129	DTCER129
	TGIB3	130	0208h	Edge	o	o	x	IER10.IEN2		DTCER130
	TGIC3	131	020Ch	Edge	o	o	x	IER10.IEN3		DTCER131
	TGID3	132	0210h	Edge	o	o	x	IER10.IEN4		DTCER132
	TCIV3	133	0214h	Edge	o	x	x	IER10.IEN5	IPR133	—
MTU4	TGIA4	134	0218h	Edge	o	o	x	IER10.IEN6	IPR134	DTCER134
	TGIB4	135	021Ch	Edge	o	o	x	IER10.IEN7		DTCER135
	TGIC4	136	0220h	Edge	o	o	x	IER11.IEN0		DTCER136
	TGID4	137	0224h	Edge	o	o	x	IER11.IEN1		DTCER137
	TCIV4	138	0228h	Edge	o	o	x	IER11.IEN2	IPR138	DTCER138

Table 14.3 Interrupt Vector Table (5/7)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	sstb Return	IER	IPR	DTCER
MTU5	TGIU5	139	022Ch	Edge	○	○	×	IER11.IEN3	IPR139	DTCER139
	TGIV5	140	0230h	Edge	○	○	×	IER11.IEN4		DTCER140
	TGIW5	141	0234h	Edge	○	○	×	IER11.IEN5		DTCER141
—	Reserved	142	0238h	—	×	×	×	—	—	—
—	Reserved	143	023Ch	—	×	×	×	—	—	—
—	Reserved	144	0240h	—	×	×	×	—	—	—
—	Reserved	145	0244h	—	×	×	×	—	—	—
—	Reserved	146	0248h	—	×	×	×	—	—	—
—	Reserved	147	024Ch	—	×	×	×	—	—	—
—	Reserved	148	0250h	—	×	×	×	—	—	—
—	Reserved	149	0254h	—	×	×	×	—	—	—
—	Reserved	150	0258h	—	×	×	×	—	—	—
—	Reserved	151	025Ch	—	×	×	×	—	—	—
—	Reserved	152	0260h	—	×	×	×	—	—	—
—	Reserved	153	0264h	—	×	×	×	—	—	—
—	Reserved	154	0268h	—	×	×	×	—	—	—
—	Reserved	155	026Ch	—	×	×	×	—	—	—
—	Reserved	156	0270h	—	×	×	×	—	—	—
—	Reserved	157	0274h	—	×	×	×	—	—	—
—	Reserved	158	0278h	—	×	×	×	—	—	—
—	Reserved	159	027Ch	—	×	×	×	—	—	—
—	Reserved	160	0280h	—	×	×	×	—	—	—
—	Reserved	161	0284h	—	×	×	×	—	—	—
—	Reserved	162	0288h	—	×	×	×	—	—	—
—	Reserved	163	028Ch	—	×	×	×	—	—	—
—	Reserved	164	0290h	—	×	×	×	—	—	—
—	Reserved	165	0294h	—	×	×	×	—	—	—
—	Reserved	166	0298h	—	×	×	×	—	—	—
—	Reserved	167	029Ch	—	×	×	×	—	—	—
—	Reserved	168	02A0h	—	×	×	×	—	—	—
—	Reserved	169	02A4h	—	×	×	×	—	—	—
POE	OEI1	170	02A8h	Level	○	×	×	IER15.IEN2	IPR170	—
	OEI2	171	02ACh	Level	○	×	×	IER15.IEN3	IPR171	—
—	Reserved	172	02B0h	—	×	×	×	—	—	—
—	Reserved	173	02B4h	—	×	×	×	—	—	—
TMR0	CMIA0	174	02B8h	Edge	○	○	×	IER15.IEN6	IEIPR174	DTCER174
	CMIB0	175	02BCh	Edge	○	○	×	IER15.IEN7		DTCER175
	OVI0	176	02C0h	Edge	○	×	×	IER16.IEN0		—
TMR1	CMIA1	177	02C4h	Edge	○	○	×	IER16.IEN1	IPR177	DTCER177
	CMIB1	178	02C8h	Edge	○	○	×	IER16.IEN2		DTCER178
	OVI1	179	02CCh	Edge	○	×	×	IER16.IEN3		—

Table 14.3 Interrupt Vector Table (6/7)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	sstb Return	IER	IPR	DTCER
TMR2	CMIA2	180	02D0h	Edge	○	○	×	IER16.IEN4	IPR180	DTCER180
	CMIB2	181	02D4h	Edge	○	○	×	IER16.IEN5		DTCER181
	OVI2	182	02D8h	Edge	○	×	×	IER16.IEN6		—
TMR3	CMIA3	183	02DCh	Edge	○	○	×	IER16.IEN7	IPR183	DTCER183
	CMIB3	184	02E0h	Edge	○	○	×	IER17.IEN0		DTCER184
	OVI3	185	02E4h	Edge	○	×	×	IER17.IEN1		—
SCI2	ERI2	186	02E8h	Level	○	×	×	IER17.IEN2	IPR186	—
	RXI2	187	02ECh	Edge	○	○	×	IER17.IEN3		DTCER187
	TXI2	188	02F0h	Edge	○	○	×	IER17.IEN4		DTCER188
	TEI2	189	02F4h	Level	○	×	×	IER17.IEN5		—
—	Reserved	190	02F8h	—	×	×	×	—	—	—
—	Reserved	191	02FCh	—	×	×	×	—	—	—
—	Reserved	192	0300h	—	×	×	×	—	—	—
—	Reserved	193	0304h	—	×	×	×	—	—	—
—	Reserved	194	0308h	—	×	×	×	—	—	—
—	Reserved	195	030Ch	—	×	×	×	—	—	—
—	Reserved	196	0310h	—	×	×	×	—	—	—
—	Reserved	197	0314h	—	×	×	×	—	—	—
—	Reserved	198	0318h	—	×	×	×	—	—	—
—	Reserved	199	031Ch	—	×	×	×	—	—	—
—	Reserved	200	0320h	—	×	×	×	—	—	—
—	Reserved	201	0324h	—	×	×	×	—	—	—
—	Reserved	202	0328h	—	×	×	×	—	—	—
—	Reserved	203	032Ch	—	×	×	×	—	—	—
—	Reserved	204	0330h	—	×	×	×	—	—	—
—	Reserved	205	0334h	—	×	×	×	—	—	—
—	Reserved	206	0338h	—	×	×	×	—	—	—
—	Reserved	207	033Ch	—	×	×	×	—	—	—
—	Reserved	208	0340h	—	×	×	×	—	—	—
—	Reserved	209	0344h	—	×	×	×	—	—	—
—	Reserved	210	0348h	—	×	×	×	—	—	—
—	Reserved	211	034Ch	—	×	×	×	—	—	—
—	Reserved	212	0350h	—	×	×	×	—	—	—
—	Reserved	213	0354h	—	×	×	×	—	—	—
SCI0	ERI0	214	0358h	Level	○	×	×	IER1A.IEN6	IPR214	—
	RXI0	215	035Ch	Edge	○	○	×	IER1A.IEN7		DTCER215
	TXI0	216	0360h	Edge	○	○	×	IER1B.IEN0		DTCER216
	TEI0	217	0364h	Level	○	×	×	IER1B.IEN1		—
SCI1	ERI1	218	0368h	Level	○	×	×	IER1B.IEN2	IPR218	—
	RXI1	219	036Ch	Edge	○	○	×	IER1B.IEN3		DTCER219
	TXI1	220	0370h	Edge	○	○	×	IER1B.IEN4		DTCER220
	TEI1	221	0374h	Level	○	×	×	IER1B.IEN5		—

Table 14.3 Interrupt Vector Table (7/7)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU	DTC	sstb Return	IER	IPR	DTCER
SCI5	ERI5	222	0378h	Level	○	×	×	IER1B.IEN6	IPR222	—
	RXI5	223	037Ch	Edge	○	○	×	IER1B.IEN7		DTCER223
	TXI5	224	0380h	Edge	○	○	×	IER1C.IEN0		DTCER224
	TEI5	225	0384h	Level	○	×	×	IER1C.IEN1		—
SCI6	ERI6	226	0388h	Level	○	×	×	IER1C.IEN2	IPR226	—
	RXI6	227	038Ch	Edge	○	○	×	IER1C.IEN3		DTCER227
	TXI6	228	0390h	Edge	○	○	×	IER1C.IEN4		DTCER228
	TEI6	229	0394h	Level	○	×	×	IER1C.IEN5		—
SCI8	ERI8	230	0398h	Level	○	×	×	IER1C.IEN6	IPR230	—
	RXI8	231	039Ch	Edge	○	○	×	IER1C.IEN7		DTCER231
	TXI8	232	03A0h	Edge	○	○	×	IER1D.IEN0		DTCER232
	TEI8	233	03A4h	Level	○	×	×	IER1D.IEN1		—
SCI9	ERI9	234	03A8h	Level	○	×	×	IER1D.IEN2	IPR234	—
	RXI9	235	03ACh	Edge	○	○	×	IER1D.IEN3		DTCER235
	TXI9	236	03B0h	Edge	○	○	×	IER1D.IEN4		DTCER236
	TEI9	237	03B4h	Level	○	×	×	IER1D.IEN5		—
SCI12	ERI12	238	03B8h	Level	○	×	×	IER1D.IEN6	IPR238	—
	RXI12	239	03BCh	Edge	○	○	×	IER1D.IEN7		DTCER239
	TXI12	240	03C0h	Edge	○	○	×	IER1E.IEN0		DTCER240
	TEI12	241	03C4h	Level	○	×	×	IER1E.IEN1		—
	SCIX0	242	03C8h	Level	○	×	×	IER1E.IEN2	IPR242	—
	SCIX1	243	03CCh	Level	○	×	×	IER1E.IEN3	IPR243	—
	SCIX2	244	03D0h	Level	○	×	×	IER1E.IEN4	IPR244	—
	SCIX3	245	03D4h	Level	○	×	×	IER1E.IEN5	IPR245	—
RIIC0	EEI0	246	03D8h	Level	○	×	×	IER1E.IEN6	IPR246	—
	RXI0	247	03DCh	Edge	○	○	×	IER1E.IEN7	IPR247	DTCER247
	TXI0	248	03E0h	Edge	○	○	×	IER1F.IEN0	IPR248	DTCER248
	TEI0	249	03E4h	Level	○	×	×	IER1F.IEN1	IPR249	—
—	Reserved	250	03E8h	—	×	×	×	—	—	—
—	Reserved	251	03ECh	—	×	×	×	—	—	—
—	Reserved	252	03F0h	—	×	×	×	—	—	—
—	Reserved	253	03F4h	—	×	×	×	—	—	—
—	Reserved	254	03F8h	—	×	×	×	—	—	—
—	Reserved	255	03FCh	—	×	×	×	—	—	—

Note 1. An interrupt source with a smaller vector number takes precedence.

### 14.3.2 Fast Interrupt Vector Table

The address of the entry in the interrupt vector table that corresponds to the vector number of the fast interrupt is placed in the fast interrupt vector register (FINTV) of the CPU.

### 14.3.3 Non-maskable Interrupt Vector Table

The non-maskable interrupt vector table area is FFFF FFF8h.

## 14.4 Interrupt Operation

The interrupt controller performs the following processing.

- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations (CPU interrupt, DTC activation)
- Determining priority

### 14.4.1 Detecting Interrupts

Interrupt requests are detected in either of two ways: the detection of edges of the interrupt signal or the detection of a level of the interrupt signal.

Edge detection or level detection is selected for the IRQ<sub>i</sub> pins ( $i = 0$  to 7) as external interrupt requests by the setting of the IRQCR<sub>i</sub>.IRQMD[1:0] bits.

For interrupts from peripheral modules, either edge detection or level detection is determined for each interrupt source. For the correspondence between interrupt sources and methods of detection, see Table 14.3, Interrupt Vector Table.

#### 14.4.1.1 Operation of Status Flags for Edge-Detected Interrupts

Figure 14.2 shows the operation of the IR<sub>n</sub>.IR flag in the case of edge detection of an interrupt from a peripheral module or on an external pin.

The IR<sub>n</sub>.IR flag is set to 1 immediately after the transition of the interrupt signal due to generation of the interrupt. If the CPU is the request destination for the interrupt, the IR flag is automatically cleared to 0 on acceptance of the interrupt. If the DTC is the request destination for the interrupt, the IR<sub>n</sub>.IR flag operation differs according to the DTC transfer settings and transfer count. For details, see Table 14.4, Operation at DTC Activation. It is not necessary to clear the IR<sub>n</sub>.IR flag by software.

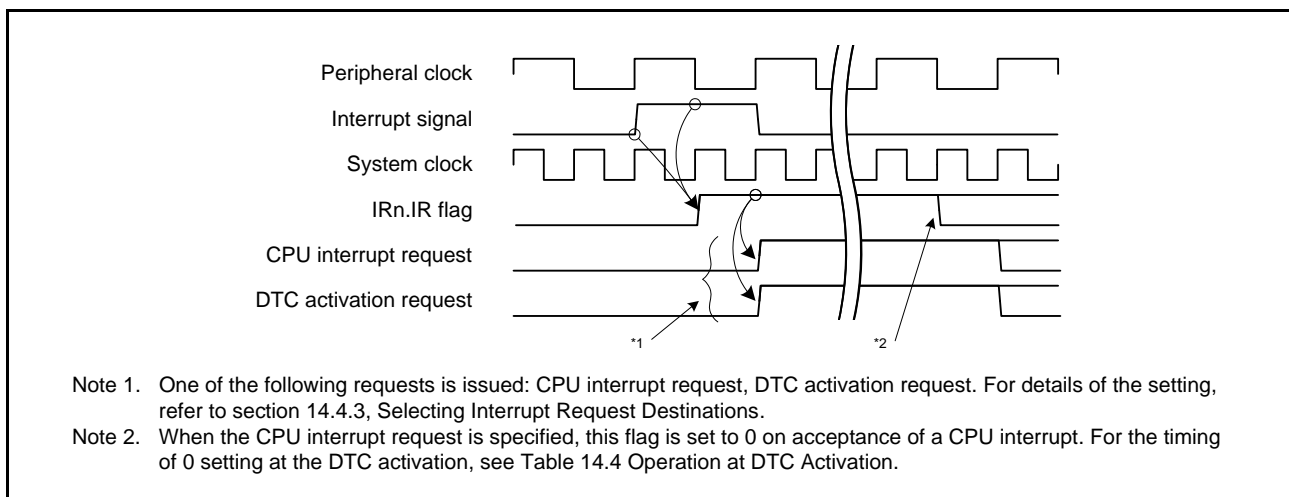


Figure 14.2 IR<sub>n</sub>.IR Flag Operation for Edge Detection Interrupts

Figure 14.3 to Figure 14.5 show the interrupt signals of the interrupt controller. Note that the timings of the interrupts with interrupt vector numbers 64 to 95 are different from those of other interrupts. For the IRQ pin interrupts with interrupt vector numbers 64 to 79, “internal delay + 2 PCLK cycles” of delay is added after the IRQ pin input. For the interrupts with interrupt vector numbers 80 to 95, “2 PCLK cycles” of delay is added.

While the IRn.IR flag is 1 after an interrupt request is generated, the interrupt request that is generated again will be ignored.\*1

Figure 14.3 shows the timing for IRn.IR flag re-setting.

Note 1. When the transmission or reception interrupt of the SCI, RSPI, RIIC, or SSI is generated with the IRn.IR flag being 1, the interrupt request is retained. After the IRn.IR flag is set to 0, the IRn.IR flag is set to 1 again by the retained request. For details, refer to descriptions of the interrupts in section 27, Serial Communications Interface (SC1e, SC1f), section 29, I<sup>2</sup>C Bus Interface (RIIC), and section 31, Serial Peripheral Interface (RSPI).

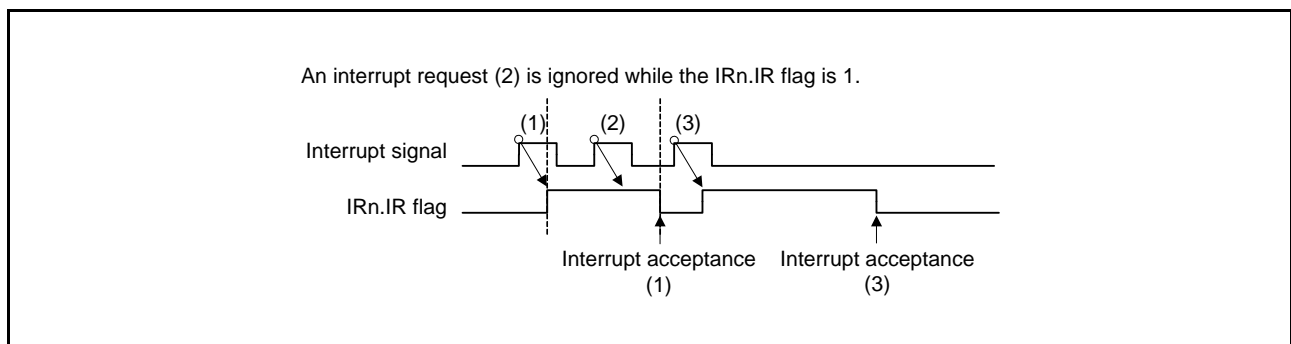


Figure 14.3 Timing for IRn.IR Flag Re-Setting

If an interrupt is disabled after the IRn.IR flag is set to 1 (output of the interrupt request is disabled by the interrupt enable bit of the relevant peripheral module), the IRn.IR flag is not affected but retains its state. Figure 14.4 shows operation when the interrupt is disabled.

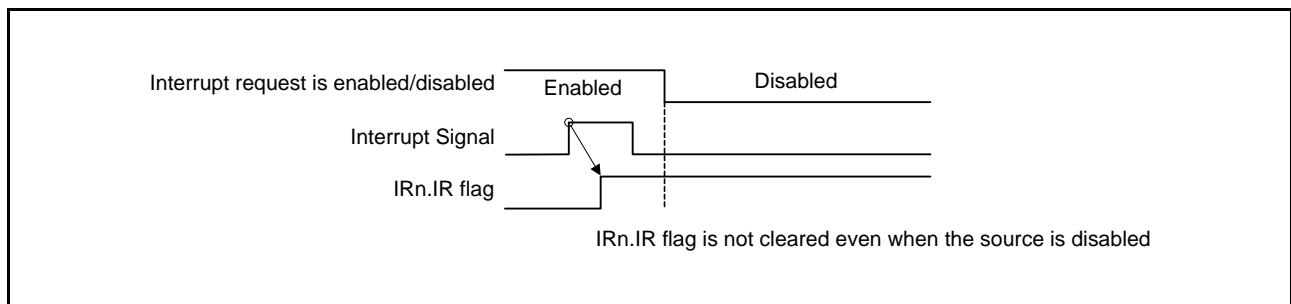
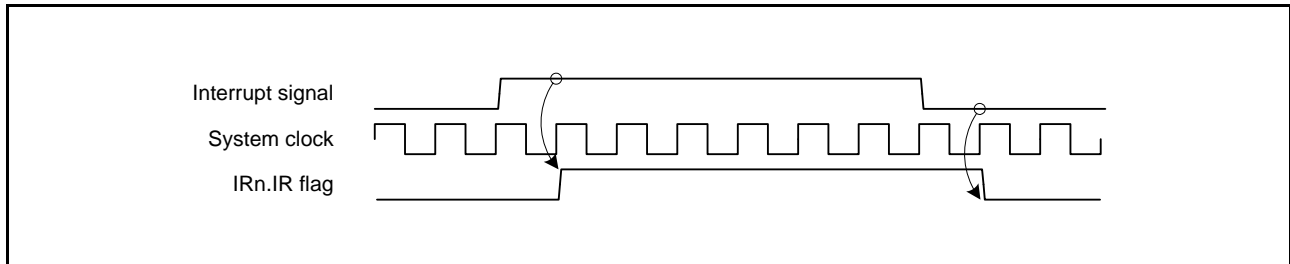


Figure 14.4 Relationship between IRn.IR Flag Operation and Disabling of Interrupt Request

### 14.4.1.2 Operation of Status Flags for Level-Detected Interrupts

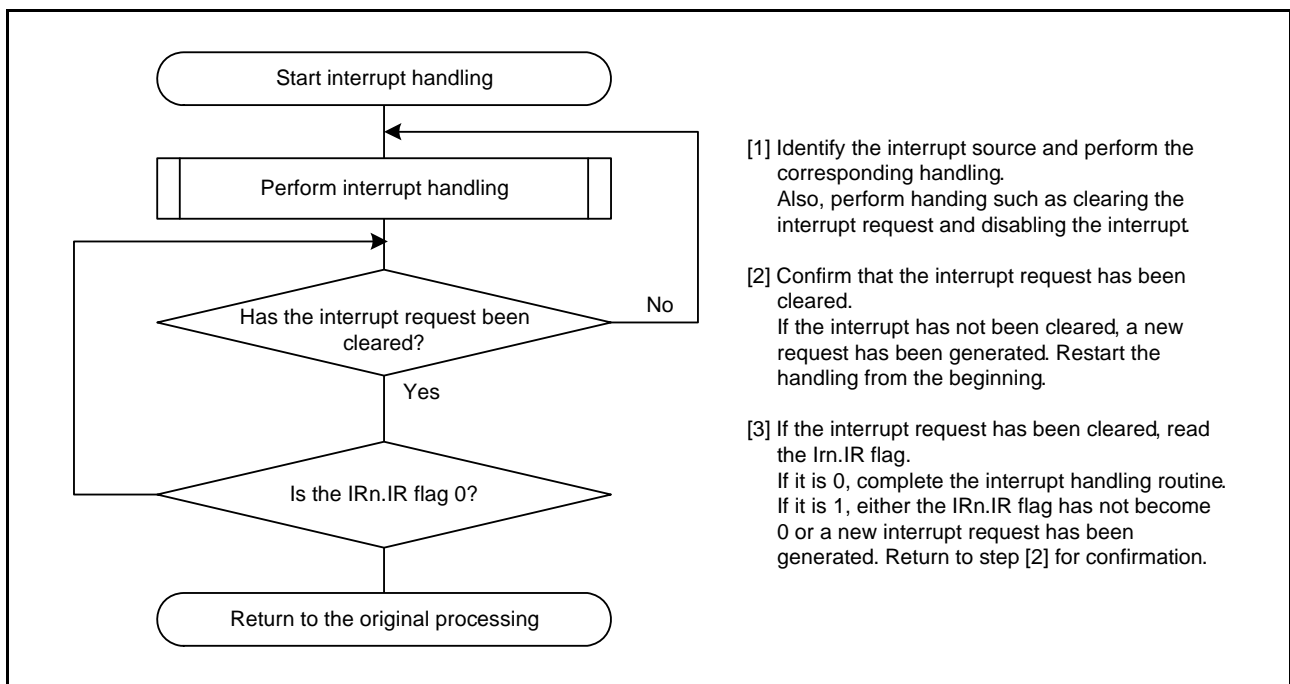
Figure 14.5 shows the operation of the interrupt status flag (IR flag) in IRn in the case of level detection of an interrupt from a peripheral module or an external pin.

The IRn.IR flag remains set to 1 as long as the interrupt signal is asserted. To set 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 set to 0 and that the IRn.IR flag has been set to 0, and then complete the interrupt handling.



**Figure 14.5 IRn.IR Flag Operation for Level Detection Interrupts**

Figure 14.6 shows the procedure for handling level detection interrupts.



**Figure 14.6 Procedure for Handling Level Detection Interrupts**



### 14.4.2 Enabling and Disabling Interrupt Sources

Enabling requests from a given interrupt source requires the following settings.

1. In the case of interrupt requests from peripheral modules, setting the interrupt enable bit for the peripheral module to permit the output of interrupt requests from the source
2. Enabling of the interrupt by the IERm.IENj bit

When an interrupt request that is enabled at the corresponding source is generated, the corresponding IRn.IR flag is set to 1. Setting the IERm.IENj bit to enable an interrupt request allows the interrupt request for which the corresponding IRn.IR is 1 to be output to the interrupt request destination. Setting the IERm.IENj bit to disable an interrupt request suspends the output of the interrupt request for which the corresponding IRn.IR is 1.

The IRn.IR flag is not affected by the IERm.IENj bit.

Use the following procedure to disable interrupt requests.

1. Set the IERm.IENj bit to disable interrupt requests.
2. Set the peripheral module interrupt output enable bit to disable the output. Read the last written register and confirm that writing is completed.
3. Check the IRn.IR flag, and set the IRn.IR flag to 0 if necessary.\*1

Note 1. To disable the transmission or reception interrupt of the SCI, RSPI, or RIIC from the enabled state, set the IRn.IR flag to 0 using the above procedure. For details, refer to descriptions of the interrupts in section 27, Serial Communications Interface (SCIE, SCIF), section 29, I<sup>2</sup>C Bus Interface (RIIC), and section 31, Serial Peripheral Interface (RSPI).

### 14.4.3 Selecting Interrupt Request Destinations

Possible settings for the request destination of each interrupt are fixed. That is, settings for request destination other than those listed in Table 14.3, Interrupt Vector Table, are not possible. Do not make an interrupt request destination setting that is not indicated by a O in Table 14.3.

If the DTC is selected as the destination for requests from an IRQ pin, be sure to set the IRQCRi.IRQMD[1:0] bits for that interrupt to select edge detection.

The following describes how to specify the destinations of interrupt requests.

#### (1) DTC Activation

Make the following settings for interrupt sources 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.

After making the above settings, set the IERm.IENj bit to 1.

In addition, set the DTC module start bit (DTCST.DTCST) to 1. The order of making settings for interrupt sources and enabling the DTC module start bit does not matter.

For the DTC setting procedure, refer to section 16.5, DTC Setting Procedure, in section 16, Data Transfer Controller (DTCa).

#### (2) CPU Interrupt Request

If the interrupt request destination is not the DTC, the interrupt request is sent to the CPU. Set the IERm.IENj bit to 1 while not the DTC activation settings described above are in place.

Table 14.4 shows operation when the DTC is the request destination.

**Table 14.4 Operation at DTC Activation**

Interrupt Request Destination	DISEL	Remaining Number of Transfer Operations	Operation for Each Request	IR*1	Interrupt Request Destination after Transfer
DTC*2	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	Cleared on interrupt acceptance by the CPU	The DTCER.DTCE bit is cleared and the CPU becomes the destination.

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 activation request) that is generated again will be ignored.

Note 2. For chain transfer, DTC transfer continues until the last chain transfer ends. Whether a CPU interrupt is generated at the end of chain transfer, the IRn.IR flag clear timing, and the interrupt request destination after transfer are determined by the state of DISEL and the remaining transfer count at the end of chain transfer. For the chain transfer, see Table 16.3, Chain Transfer Conditions in section 16, 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 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 (1) 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, set the IERm.IENj bit to 0.
2. Check the state of transfer by the DTC. If transfer is in progress, wait for its completion.
3. Make the settings described under (1) DTC Activation.

#### 14.4.4 Determining Priority

Interrupt priority is determined for each interrupt request destination.

The priority for each interrupt request destination is determined as follows.

##### (1) Determining Priority When the CPU is the Request Destination of the Interrupt

A source selected for the fast interrupt has the highest priority. After that, an interrupt source with a larger value of the interrupt priority level select bits (IPRn.IPR[3:0]) takes priority. If interrupts with the same priority level are generated by multiple sources, the source with the smallest vector number takes precedence.

##### (2) Determining Priority When the DTC is the Request Destination of the Interrupt

The IPRn.IPR[3:0] bits have no effect. An interrupt source with a smaller vector number takes precedence.

#### 14.4.5 Multiple Interrupts

To enable multiple interrupts of the CPU, set the PSW.I bit to 1 (interrupt enabled) in the handling routine of accepted interrupts.

The PSW.IPL[3:0] bits immediately after processing branches to the interrupt handling routine are set to the same value as the interrupt priority level of the accepted interrupt request. If an interrupt request which has an interrupt level higher than that of the PSW.IPL[3:0] bits is generated at this time, this interrupt request (for multiple interrupts) is accepted. If the interrupt priority level of the accepted interrupt request is 15 (fast interrupt or interrupt when IPR[3:0] are set to 1111b), multiple interrupts are not generated.

### 14.4.6 Fast Interrupt

The fast interrupt is an interrupt for executing a faster interrupt response by the CPU, so only one of the interrupt sources can be assigned.

The interrupt priority level of the fast interrupt is 15 (highest) regardless of the setting of the IPRn.IPR[3:0] bits. In addition, the fast interrupt is accepted with precedence over other interrupt sources with level 15. However, when the value of the PSW.IPL[3:0] bits are 1111b (priority level 15), even the fast interrupt cannot be accepted.

To assign an interrupt source to the fast interrupt, specify the vector number of the source in the FIR.FVCT[7:0] bits, and set the FIR.FIEN bit to 1 (fast interrupt is enabled).

For details on the fast interrupt, see section 2, CPU and section 13, Exception Handling.

### 14.4.7 Digital Filter

The digital filter function is provided for the external interrupt request IRQi pins (i = 0 to 7) and NMI pin interrupt. The digital filter samples input signals at the filter sampling clock (PCLK) and removes the pulses of which length is less than three sampling cycles.

To use the digital filter for the IRQi pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the IRQFLTC0.FCLKSELi[1:0] bits (i = 0 to 7) and set the IRQFLTE0.FLTENi bit to 1 (digital filter enabled).

To use the digital filter for the NMI pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the NMIFLTC.NFCLKSEL[1:0] bits and set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).

Figure 14.7 shows an example of digital filter operation.

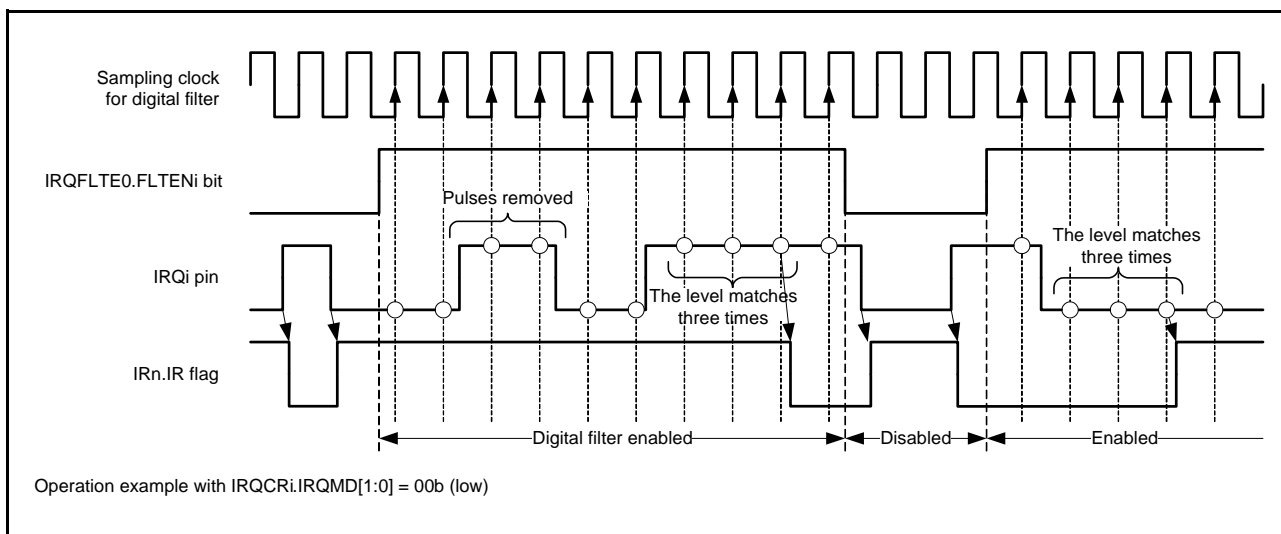


Figure 14.7 Digital Filter Operation Example

Before software standby mode is entered, set the IRQFLTE0.FLTENi and NMIFLTE.NFLTEN bits to 0 (digital filter disabled). To use the digital filter again after return from software standby mode, set the IRQFLTE0.FLTENi or NMIFLTE.NFLTEN bit to 1 (digital filter enabled).

### 14.4.8 External Pin Interrupts

The procedure for using the signal on an external pin as an interrupt is as follows.

1. Set the IERm.IENj bit to 0 (interrupt request disabled).
2. Set the IRQFLTE0.FLTENi bit (i = 0 to 7) to 0 (digital filter disabled).
3. Set the digital filter sampling clock with the IRQFLTC0.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. Set the corresponding IRn.IR flag to 0 (if edge detection is in use).
7. Set the IRQFLTE0.FLTENi bit to 1 (digital filter enabled).
8. If the interrupt is to be used for DTC activation, set the DTCERn.DTCE bit. (The interrupt will be a CPU interrupt if settings is not made.)
9. Set the IERm.IENj bit to 1 (interrupt request enabled).

## 14.5 Non-maskable Interrupt Operation

There are six types of non-maskable interrupt: the NMI pin interrupt, oscillation stop detection interrupt, IWDT underflow/refresh error, voltage monitoring 1 interrupt, and voltage monitoring 2 interrupt. Non-maskable interrupts are only usable as interrupts for the CPU; that is, they are not capable of DTC 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 in the non-maskable interrupt handler.

Non-maskable interrupts are disabled by default. If a system is to use non-maskable interrupts, the following procedure must be followed at the beginning of program processing.

Non-maskable interrupt usage procedure:

1. Set the stack pointer (SP).
2. To use the NMI pin, set 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 interrupt detection method with the NMICR.NMIMD bit.
5. To use the NMI pin, write 1 to the NMICLR.NMICLR bit to set 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 handling, refer to [section 13, Exception Handling](#).

Writing 1 to the NMICLR.NMICLR bit sets the NMI status flag (NMISR.NMIST) to 0.

Writing 1 to the NMICLR.OSTCLR bit sets the oscillation stop detection interrupt status flag (NMISR.OSTST) to 0.

Writing 1 to the NMICLR.IWDTCLR bit sets the IWDT underflow/refresh error status flag (NMISR.IWDTST) to 0.

Writing 1 to the NMICLR.LVD1CLR bit sets the voltage monitoring 1 interrupt status flag (NMISR.LVD1ST) to 0.

Writing 1 to the NMICLR.LVD2CLR bit sets the voltage monitoring 2 interrupt status flag (NMISR.LVD2ST) to 0.

## 14.6 Return from Power-Down States

The interrupt sources that can be used to return operation from sleep mode, deep sleep mode, or software standby mode are listed in Table 14.3, Interrupt Vector Table.

For details, refer to section 11, Low Power Consumption. The following describes how to use an interrupt to return operation from each low power consumption mode.

### 14.6.1 Return from Sleep Mode or Deep Sleep Mode

If the interrupt controller is to return operation from sleep mode in response to an interrupt or non-maskable interrupt, make the following settings for the interrupt.

- Interrupts
  1. Select the CPU as the interrupt request destination.
  2. Use the IERm.IENj bit to enable the given interrupt request.
  3. Set a priority level higher than that set in the CPU.PSW.IPL[3:0] bits.
- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

### 14.6.2 Return from Software Standby Mode

The interrupt controller can return operation from a non-maskable interrupt or an interrupt that enables the return from the software standby mode.

The conditions for the return are listed below.

- Interrupts
  1. Select the interrupt source that enables the return from the software standby mode.
  2. Select the CPU as the interrupt request destination.
  3. Use the IERm.IENj bit to enable the given interrupt request.
  4. Set a priority level higher than that set in the CPU.PSW.IPL[3:0] bits.  
(For the interrupt source specified as a fast interrupt, as well as setting the fast interrupt set register (FIR), the interrupt priority level (IPRn) should be set above the level set by CPU.PSW.IPL.)

Interrupt requests through the IRQ pins that do not satisfy the above conditions are not detected while the clock is stopped in software standby mode.

- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

- Procedure to make a transition to/from software standby mode
  1. Before software standby mode is entered, disable the digital filter for the interrupt source as a return target (IRQFLTE0.FLTENi = 0, NMIFLTE.NFLTEN = 0).
  2. To use the digital filter again after return from software standby mode, enable the digital filter (IRQFLTE0.FLTENi = 1, NMIFLTE.NFLTEN = 1).

## 14.7 Usage Note

### 14.7.1 Note on WAIT Instruction Used with Non-Maskable Interrupt

Before executing the WAIT instruction, check to see that all the status flags in NMISR are 0.

## 15. Buses

### 15.1 Overview

Table 15.1 lists the bus specifications, Figure 15.1 shows the bus configuration, and Table 15.2 lists the addresses assigned to each bus.

**Table 15.1 Bus Specifications**

Bus Type		Description
CPU bus	Instruction bus	<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 buses	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 DTC</li> <li>• Connected to on-chip memory (RAM, ROM)</li> <li>• Operates in synchronization with the system clock (ICLK)</li> </ul>
Internal peripheral buses	Internal peripheral bus 1	<ul style="list-style-type: none"> <li>• Connected to peripheral modules (DTC, 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</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 6	<ul style="list-style-type: none"> <li>• Connected to ROM (P/E) and E2 DataFlash</li> <li>• Operates in synchronization with the FlashIF clock (FCLK)</li> </ul>

P/E: Programming/Erase

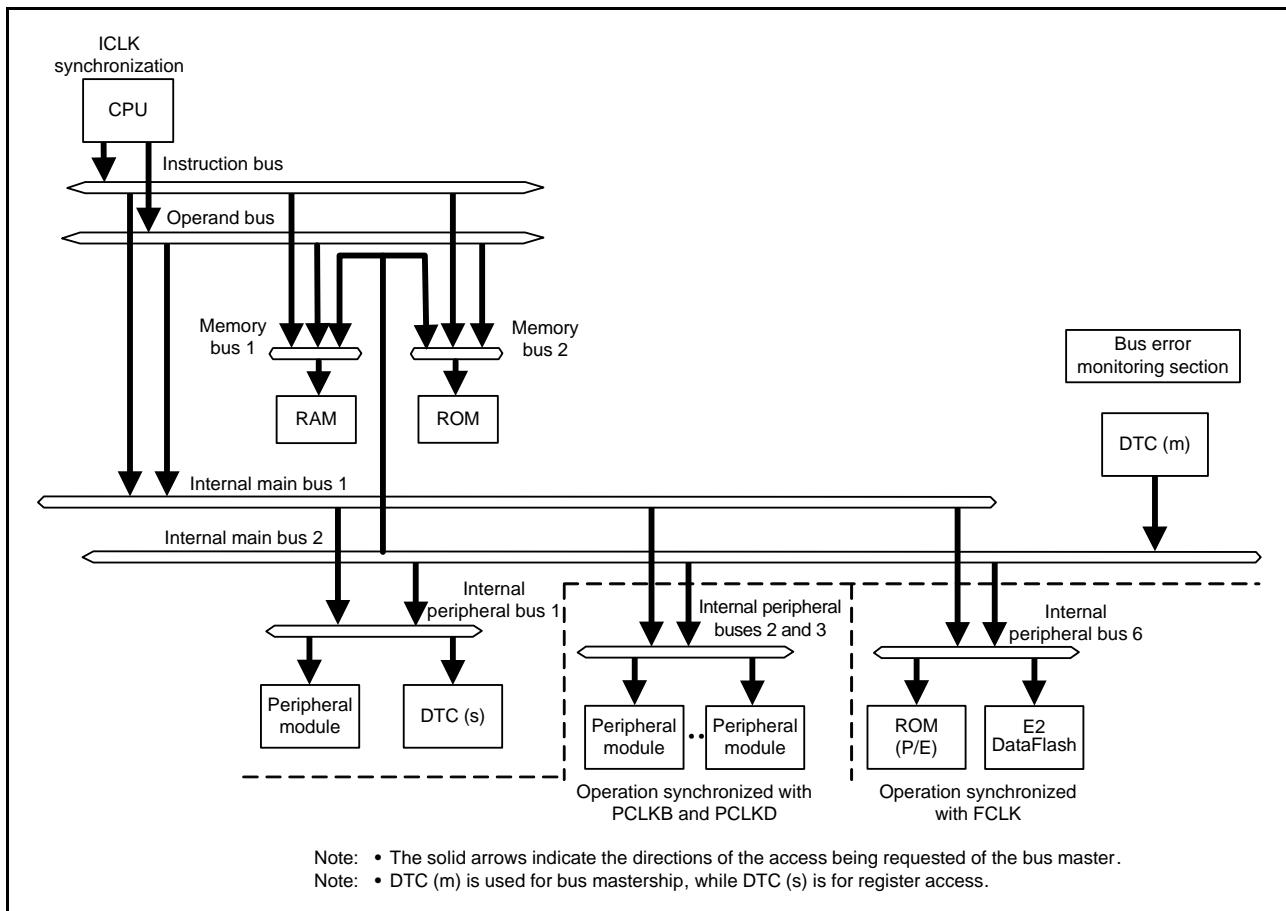


Figure 15.1 Bus Configuration

Table 15.2 Addresses Assigned for Each Bus

Address	Bus	Area
0000 0000h to 0000 FFFFh	Memory bus 1	RAM
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	Peripheral I/O registers
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	
000A 0000h to 000B FFFFh	Internal peripheral bus 3	
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	E2 DataFlash memory and ROM (for programming/erasure)
8000 0000h to FEFF FFFFh	Memory bus 2	ROM (for reading only)
FF00 0000h to FFFF FFFFh		



## 15.2 Description of Buses

### 15.2.1 CPU Buses

The CPU buses consist of the instruction and operand buses, which are connected to internal main bus 1. As the names suggest, the instruction bus is used to fetch instructions for the CPU, while the operand bus is used for operand access. Connection of the instruction and operand buses to RAM and ROM provides the CPU with direct access to these areas, i.e. access is not via internal main bus 1. However, only reading is possible in direct access to ROM by the CPU; programming and erasure are handled via an internal peripheral bus.

Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.

If instruction fetching and operand access are requested for different buses (memory bus 1, memory bus 2, and internal main bus 1), the bus-access operations can proceed simultaneously. For example, parallel access to ROM and RAM is possible.

### 15.2.2 Memory Buses

The memory buses consist of memory bus 1 and memory bus 2. RAM is connected to memory bus 1 and ROM is connected to memory bus 2. Requests for bus mastership from the CPU buses (instruction fetching and operand) and internal main bus 2 are arbitrated through memory buses 1 and 2.

The priority order of CPU bus and internal main bus 2 can be set using the memory bus 1 (RAM) priority control bits (BPRA[1:0]) and memory bus 2 (ROM) priority control bits (BPRO[1:0]) in the bus priority control register (BUSPRI) for the corresponding memory buses. When the priority order is fixed, internal main bus 2 has priority over the CPU bus (operand over instruction fetching). When the priority order is toggled, the bus for which a request has been accepted has lower priority.

### 15.2.3 Internal Main Buses

The internal main buses consist of a bus for use by the CPU (internal main bus 1) and a bus for use by the other bus-master modules, i.e. the DTC (internal main bus 2).

Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.

Requests for bus mastership from the DTC is arbitrated by internal main bus 2. The order of priority is as shown in Table 15.3.

If the CPU and another bus master are requesting access to different buses (on-chip memory, internal peripheral buses 1 to 3 and 6), the respective bus-access operations can proceed simultaneously.

However, when the CPU executes the XCHG instruction, requests for bus access from masters other than the CPU are not accepted until data transfer for the XCHG instruction is completed regardless of the bus priority control register (BUSPRI) setting. Furthermore, requests for bus access from masters other than the DTC are not accepted during reading and writing-back of transfer control information for the DTC.

**Table 15.3 Order of Priority for Bus Masters**

Priority	Bus Master
High	DTC
↑	
Low	CPU

### 15.2.4 Internal Peripheral Buses

Connection of peripheral modules to the internal peripheral buses is as described in Table 15.4.

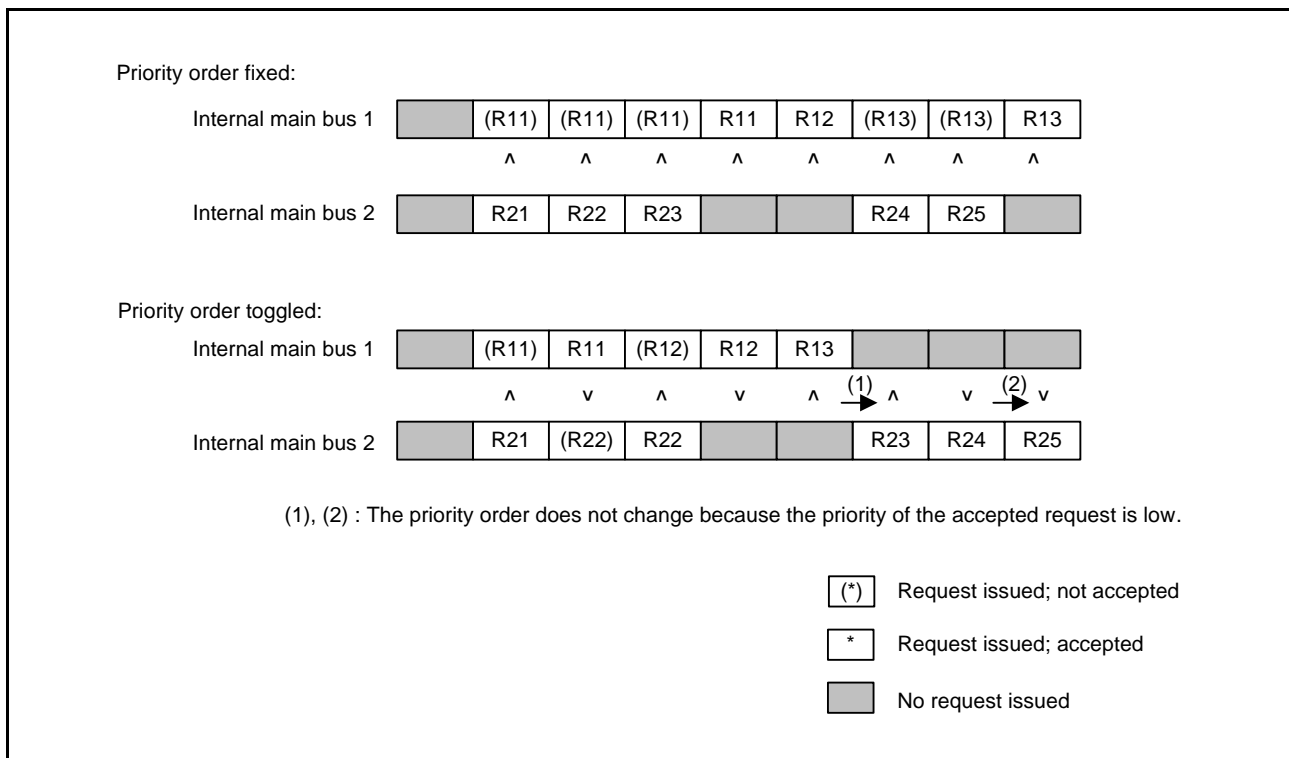
**Table 15.4 Connection of Peripheral Modules to the Internal Peripheral Buses**

Type of Bus	Peripheral Modules
Internal peripheral bus 1	DTC, interrupt controller, and bus error monitoring section
Internal peripheral bus 2	Peripheral modules other than those connected to internal peripheral buses 1 and 3
Internal peripheral bus 3	USB
Internal peripheral bus 6	ROM (P/E)/E2 DataFlash

Requests for bus mastership from the CPU (internal main bus 1) and other bus masters (internal main bus 2) are arbitrated through internal peripheral buses 1 to 3 and 6.

The priority order of the two internal main buses can be set using the bus priority control register (BUSPRI). The priority order can be set with the internal peripheral bus 1 priority control bits (BUSPRI.BPIB[1:0]), internal peripheral buses 2 and 3 priority control bits (BUSPRI.BPGB[1:0]), and internal peripheral bus 6 priority control bits (BUSPRI.BPFB[1:0]) for the corresponding internal peripheral buses. When the priority order is fixed, internal main bus 2 has priority over internal main bus 1. When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

The order of accepting requests may change depending on the BUSPRI setting (see Figure 15.2).



**Figure 15.2 Priority Order Between Internal Peripheral Bus Accesses**

### 15.2.5 Write Buffer Function (Internal Peripheral Bus)

The internal peripheral bus has the write buffer function, which allows the next round of bus access to start, before the current write access is completed, in write access. However, if the following round of bus access is from the same bus master but to the different internal peripheral bus, it is suspended until the bus operations already in progress are completed. When read access to the internal memory is scheduled after the write access to the internal peripheral bus from the CPU, the following round of bus access can be started before the current bus operation is completed and thus the order of accesses may be changed (see Figure 15.3).

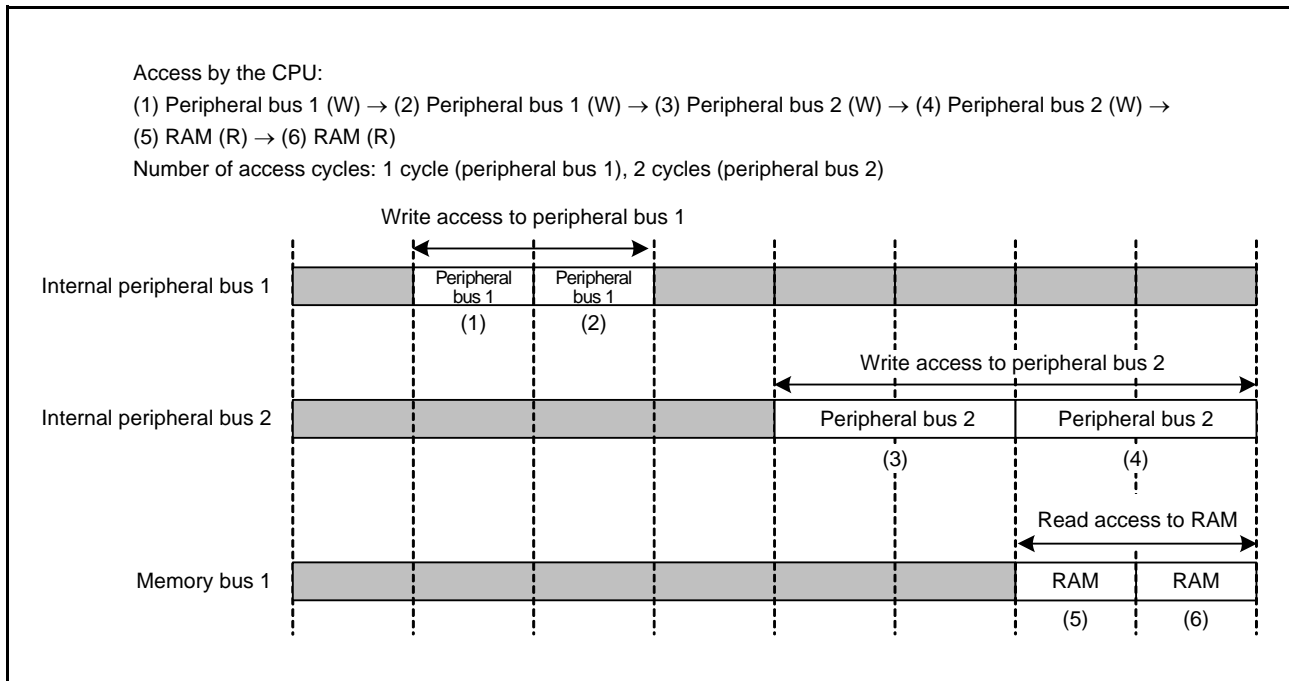


Figure 15.3 Write Buffer Function

### 15.2.6 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave modules. For example, if the CPU is fetching an instruction from ROM and an operand from RAM, the DTC is able to handle transfer between a peripheral bus and peripheral bus at the same time.

An example of parallel operations is shown in Figure 15.4. In this example, the CPU is able to employ the instruction and operand buses for simultaneous access to ROM and RAM, respectively. Furthermore, the DTC simultaneously employs internal main bus 2 for access to a peripheral bus during access to RAM and ROM by the CPU.

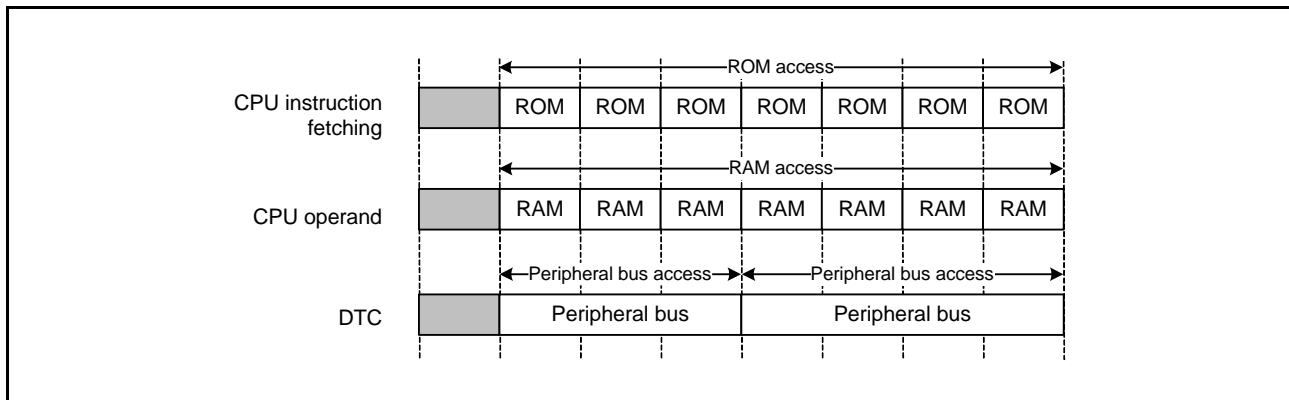


Figure 15.4 Example of Parallel Operations

### 15.2.7 Restrictions

#### (1) Prohibition of Access that Spans Multiple Areas of Address Space

Single access that spans two areas of the address space is prohibited, and operation of such an access is not guaranteed. Ensure that a single word or longword access does not span across two areas by crossing address space area boundaries.

#### (2) Restrictions on RMPA and String-Manipulation Instructions

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

## 15.3 Register Descriptions

### 15.3.1 Bus Error Status Clear Register (BERCLR)

Address(es): 0008 1300h

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

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

Bit	Symbol	Bit Name	Description	R/W
b0	STSCLR	Status Clear	0: Invalid 1: Bus error status register cleared	(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only writing 1 is effective; i.e. writing 0 has no effect.

#### STSCLR Bit (Status Clear)

Writing 1 to this bit clears the bus error status registers 1 and 2 (BERSR1 and BERSR2).

Writing 0 has no effect. It is read as 0.

### 15.3.2 Bus Error Monitoring Enable Register (BEREN)

Address(es): 0008 1304h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	TOEN	IGAEN

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

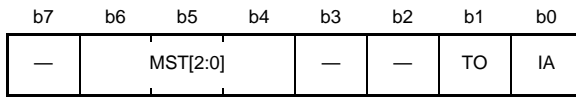
Bit	Symbol	Bit Name	Description	R/W
b0	IGAEN	Illegal Address Access Detection Enable	0: Illegal address access detection is disabled. 1: Illegal address access detection is enabled.	R/W
b1	TOEN	Timeout Detection Enable*1, *2	0: Bus timeout detection is disabled. 1: Bus timeout detection is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When detection is disabled (the TOEN bit is cleared to 0), bus access can cause the bus to freeze.

Note 2. Do not set the TOEN bit to 0 (bus timeout detection disabled) while timeout errors are being detected.

### 15.3.3 Bus Error Status Register 1 (BERSR1)

Address(es): 0008 1308h



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

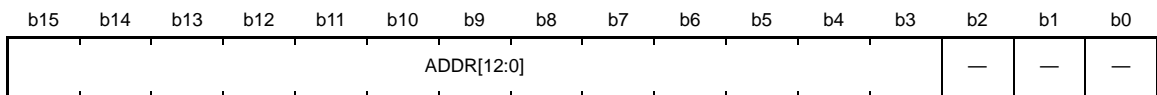
Bit	Symbol	Bit Name	Description	R/W																											
b0	IA	Illegal Address Access	0: Illegal address access not made 1: Illegal address access made	R																											
b1	TO	Timeout	0: Timeout not generated 1: Timeout generated	R																											
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R																											
b6 to b4	MST[2:0]	Bus Master Code	<table border="0"> <tr> <td>b6</td> <td>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</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	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																													
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.

### 15.3.4 Bus Error Status Register 2 (BERSR2)

Address(es): 0008 130Ah



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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15 to b3	ADDR[12:0]	Bus Error Occurrence Address	The upper 13 bits of an address that was accessed when a bus error occurred (in units of 512 Kbytes).	R

### 15.3.5 Bus Priority Control Register (BUSPRI)

Address(es): 0008 1310h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	BPFB[1:0]		—	—	BPGB[1:0]		BPIB[1:0]		BPRO[1:0]		BPRA[1:0]	
Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BPRA[1:0]	Memory Bus 1 (RAM) Priority Control	b1 b0 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b3, b2	BPRO[1:0]	Memory Bus 2 (ROM) Priority Control	b3 b2 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b5, b4	BPIB[1:0]	Internal Peripheral Bus 1 Priority Control	b5 b4 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b7, b6	BPGB[1:0]	Internal Peripheral Buses 2 and 3 Priority Control	b7 b6 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b9, b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11, b10	BPFB[1:0]	Internal Peripheral Bus 6 Priority Control	b11 b10 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be written to only once while the DTC is stopped. When they are written to more than one time, the operation is not guaranteed.

#### BPRA[1:0] Bits (Memory Bus 1 (RAM) Priority Control)

These bits specify the priority order for memory bus 1 (RAM).

When the priority order is fixed, internal main bus 2 has priority over the CPU bus.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

#### BPRO[1:0] Bits (Memory Bus 2 (ROM) Priority Control)

These bits specify the priority order for memory bus 2 (ROM).

When the priority order is fixed, internal main bus 2 has priority over the CPU bus.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

#### BPIB[1:0] Bits (Internal Peripheral Bus 1 Priority Control)

These bits specify the priority order for internal peripheral bus 1.

When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

**BPGB[1:0] Bits (Internal Peripheral Buses 2 and 3 Priority Control)**

These bits specify the priority order for internal peripheral buses 2 and 3.

When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

**BPFB[1:0] Bits (Internal Peripheral Bus 6 Priority Control)**

These bits specify the priority order for internal peripheral bus 6.

When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.



## 15.4 Bus Error Monitoring Section

The bus error monitoring section monitors the individual areas for bus errors, and when a bus error occurs, the error is indicated to the bus master.

### 15.4.1 Type of Bus Error

There is a illegal address access bus error.

Illegal address access is the detection of access to an illegal area.

#### 15.4.1.1 Illegal Address Access

When the illegal address access detection enable bit (IGAEN) in the bus error monitoring enable register (BEREN) is set to 1, access to an illegal address area leads to illegal address access errors.

The address ranges where access will lead to illegal address access errors are listed in Table 15.5.

#### 15.4.1.2 Timeout

When the timeout detection enable bit in the bus error monitoring enable register is enabled (BEREN.TOEN = 1), bus access that is not completed within 768 cycles leads to a timeout error.

- Internal peripheral buses (2 and 3): Bus access is not completed within 768 peripheral module clock (PCLKB) cycles from the start of the access.  
If a timeout error occurs, accesses from the bus master are not accepted for 256 PCLKB cycles.
- Internal peripheral bus (6): Bus access is not completed within 768 FlashIF clock (FCLK) cycles from the start of the access.  
If a timeout error occurs, accesses from the bus master are not accepted for 256 FCLK cycles.

### 15.4.2 Operations When a Bus Error Occurs

When a bus error occurs, the error is indicated to the CPU. Operation is not guaranteed when a bus error occurs.

- Bus error indication to the CPU  
An interrupt is generated. The ICU.IERn register can specify whether to generate an interrupt in the case of a bus error.

### 15.4.3 Conditions Leading to Bus Errors

Table 15.5 lists the type of bus errors for each area in the respective address space.

If an illegal address access error is detected when no bus error has occurred (bus error status register n (BERSRn; n = 1, 2) is cleared), the detected error is reflected in the BERSRn register. Once a bus error occurs, no subsequent bus errors are reflected in the register unless the register is cleared.

If bus errors are simultaneously caused by two or more bus masters, error information of only one bus master is reflected. Once a bus error occurs, the status is retained until the BERSRn register is cleared.

**Table 15.5 Type of Bus Errors**

Address	Type of Area	Type of Error	
		Illegal Address Access	Timeout
0000 0000h to 0007 FFFFh	Memory bus 1	—	—
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	—	—
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	Δ	—
000A 0000h to 000B FFFFh	Internal peripheral bus 3	Δ	Δ
000C 0000h to 000E FFFFh	Reserved area	○	—
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	Δ	—
0100 0000h to 07FF FFFFh	Reserved area	○	—
0800 0000h to 0FFF FFFFh	Reserved area	—	—
1000 0000h to 7FFF FFFFh	Reserved area	○	—
8000 0000h to FFFF FFFFh	Memory bus 2	—	—

—: A bus error does not result.

Δ: A bus error may or may not result.

○: A bus error results.

Note: The capacity of the RAM, data flash, and ROM differs depending on the product. For details, refer to section 40, RAM, and section 41, Flash Memory.

## 16. Data Transfer Controller (DTCa)

This MCU incorporates a data transfer controller (DTC).

The DTC is activated by an interrupt request to perform data transfers.

### 16.1 Overview

Table 16.1 lists the specifications of the DTC, and Figure 16.1 shows a block diagram of the DTC.

**Table 16.1 DTC Specifications**

Item	Description
Transfer modes	<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 number of repeat transfers is 256, and the maximum data transfer size is 256 × 32 bits, 1024 bytes.</li> <li>• Block transfer mode A single activation leads to the transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.</li> </ul>
Transfer channel	<ul style="list-style-type: none"> <li>• Channel transfer corresponding to the interrupt source is possible (transferred by the DTC activation request from the ICU).</li> <li>• Multiple data 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 except reserved areas)</li> <li>• In full-address mode: 4 Gbytes (Area from 0000 0000h to FFFF FFFFh except reserved areas)</li> </ul>
Data transfer units	<ul style="list-style-type: none"> <li>• Single data: 1 byte (8 bits), 1 word (16 bits), 1 longword (32 bits)</li> <li>• Single block size: 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>
Event link function	An event link request is generated after one data transfer (for block, after one block transfer).
Read skip	Transfer information read skip can be executed.
Write-back skip	When "fixed" is selected for transfer source address or transfer destination address, write-back skip is executed.
Low power consumption function	Module stop state can be set.

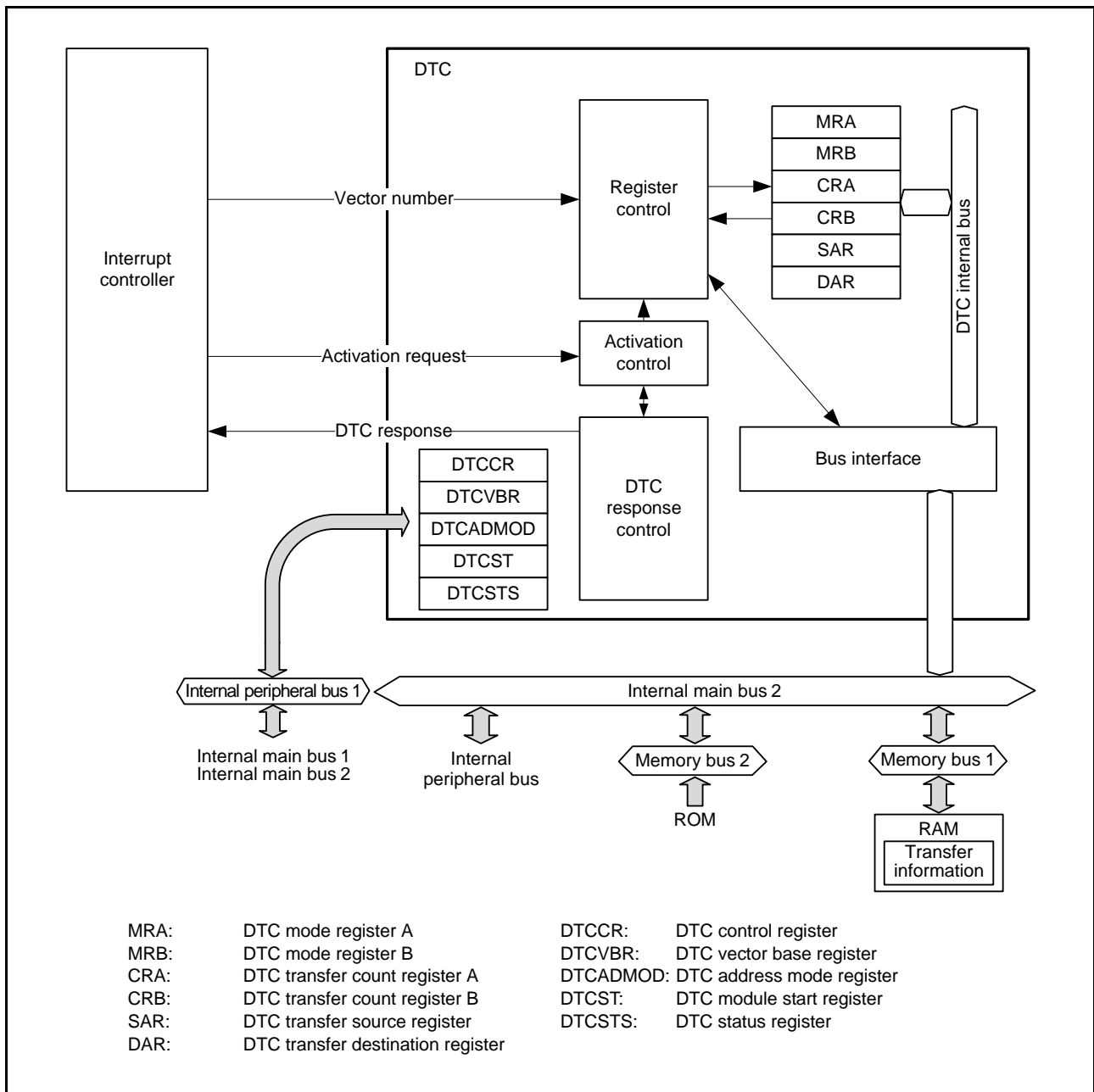


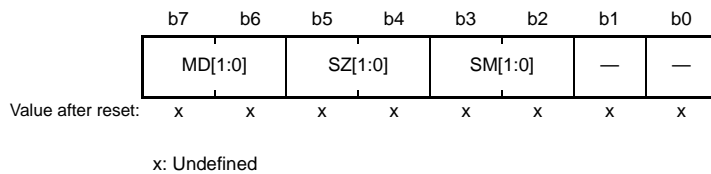
Figure 16.1 DTC Block Diagram

## 16.2 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 these DTC internal registers are placed in the RAM area as transfer information. When an activation request is generated, the DTC reads the transfer information 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.

### 16.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU)

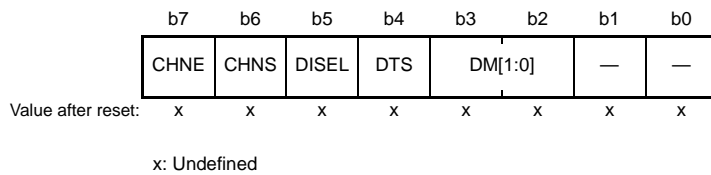


Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as undefined. The write value should be 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 0 0: Address in the SAR register is fixed. (write-back to SAR is skipped.) 0 1: Address in the SAR register is fixed. (write-back 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: Byte (8-bit) transfer 0 1: Word (16-bit) transfer 1 0: Longword (32-bit) transfer 1 1: Setting prohibited	—
b7, b6	MD[1:0]	DTC Transfer Mode Select	b7 b6 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

MRA cannot be accessed directly from the CPU.

## 16.2.2 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	$b^3 b^2$ 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 MRA.SZ[1:0] bits = 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 = 00b, -2 when SZ[1:0] bits = 01b, -4 when MRA.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, refer to Table 16.3, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the activation source flag is not cleared, and an interrupt request to the CPU is not generated.

### CHNE Bit (DTC Chain Transfer Enable)

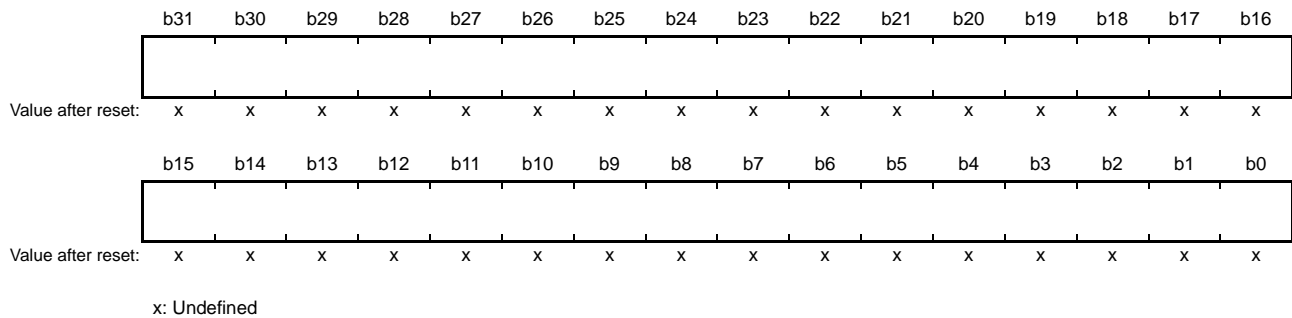
The CHNE bit enables or disables chain transfer.

The chain transfer condition is selected by the CHNS bit.

For details of chain transfer, refer to section 16.4.6, Chain Transfer.

### 16.2.3 DTC Transfer Source Register (SAR)

Address(es): (inaccessible directly from the CPU)



SAR is used to set the transfer source start address.

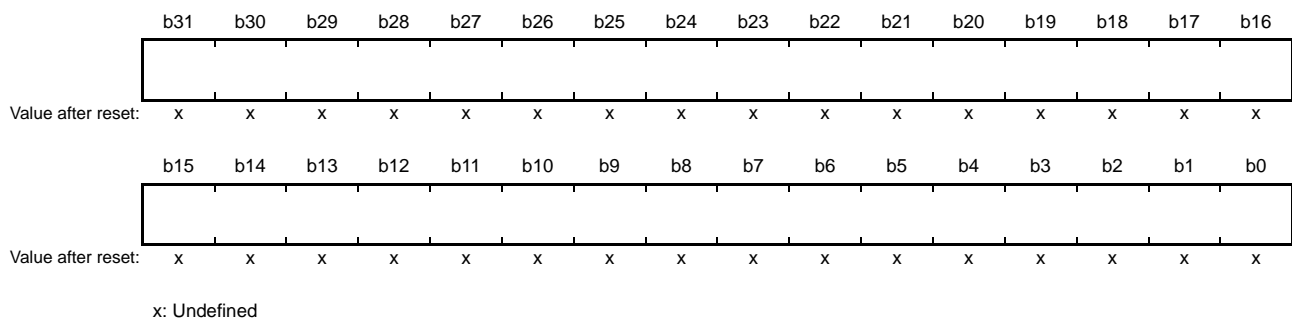
In full-address mode, 32 bits are valid.

In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.

SAR cannot be accessed directly from the CPU.

### 16.2.4 DTC Transfer Destination Register (DAR)

Address(es): (inaccessible directly from the CPU)



DAR is used to set the transfer destination start address.

In full-address mode, 32 bits are valid.

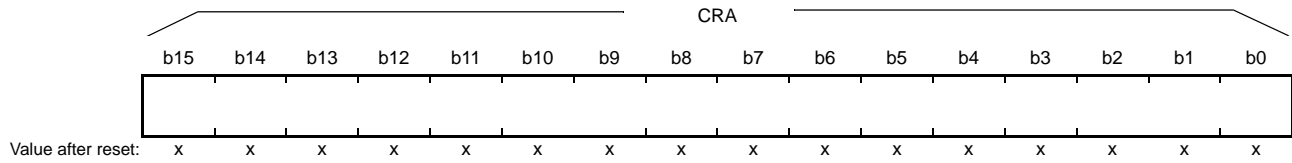
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.

DAR cannot be accessed directly from the CPU.

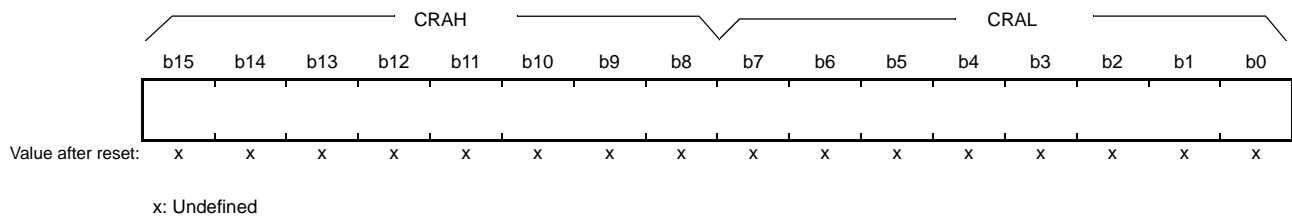
## 16.2.5 DTC Transfer Count Register A (CRA)

Address(es): (inaccessible directly from the CPU)

- Normal transfer mode



- Repeat transfer mode/block transfer mode



Symbol	Register Name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count.	—
CRAH	Transfer Counter A Upper Register		—

Note: The function depends on transfer mode.

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 (MRA.MD[1:0] bits = 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 (MRA.MD[1:0] bits = 01b)

The CRAH register retains the transfer count and the CRAL register functions as an 8-bit transfer counter.

The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.

The CRAL value is decremented (-1) at each data transfer. When it reaches 00h, the CRAH value is transferred to CRAL.

### (3) Block transfer mode (MRA.MD[1:0] bits = 10b)

The CRAH register retains the block size and the CRAL register functions as an 8-bit block size counter.

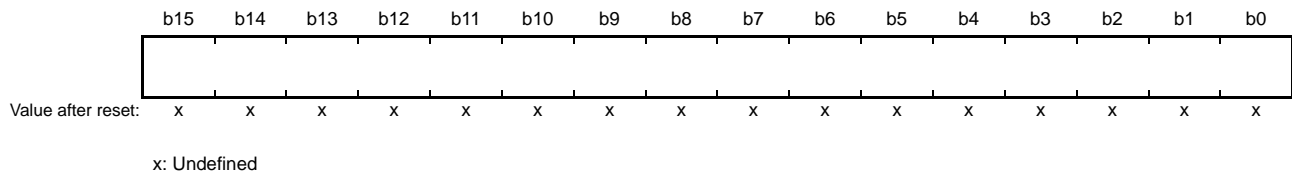
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.

The CRAL value is decremented (-1) at each data transfer. When it reaches 00h, the CRAH value is transferred to CRAL.



## 16.2.6 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.

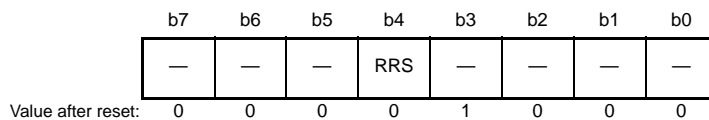
The CRB value is decremented (−1) when the final data of a single block size is transferred.

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.

## 16.2.7 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 Information Read Skip Enable	0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

### RRS Bit (DTC Transfer Information Read Skip Enable)

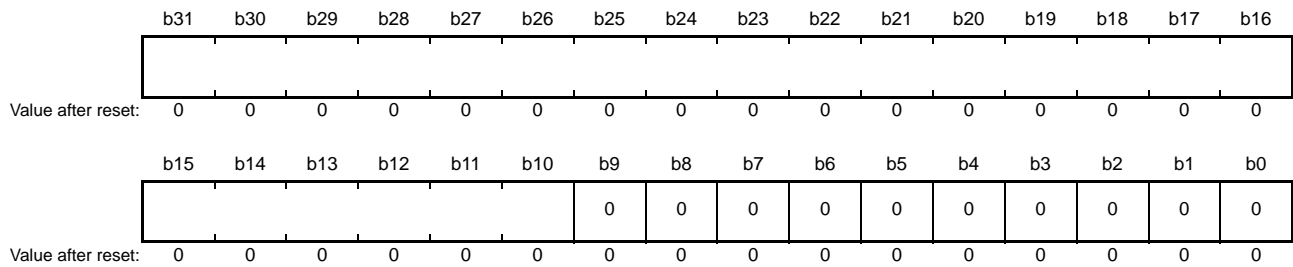
The DTC vector number is compared with the vector number in the previous activation process.

When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred information. However, when the previous transfer was chain transfer, the transferred information is read regardless of the value of the RRS bit.

Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, the transferred information is read regardless of the RRS bit value.

## 16.2.8 DTC Vector Base Register (DTCVBR)

Address(es): 0008 2404h



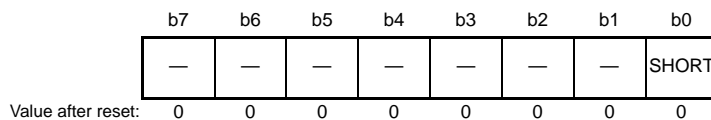
Bit	Bit Name	Description	R/W
b9 to b0	DTC Vector Base Address (Lower 10 bits)	These bits are read as 0. The write value should be 0.	R
b31 to b10	DTC Vector Base Address (Upper 22 bits)	Writing to the upper 4 bits (b31 to b28) is ignored, and the address of this register is extended by the value specified 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 FC00h and F800 0000h to FFFF FC00h in 1-Kbyte units.

## 16.2.9 DTC Address Mode Register (DTCADM0D)

Address(es): 0008 2408h



Bit	Symbol	Bit Name	Description	R/W
b0	SHORT	Short-Address Mode Set	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.

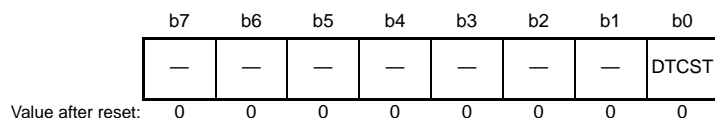
### SHORT Bit (Short-Address Mode Set)

Full-address mode allows the DTC to access to a 4-Gbyte space (0000 0000h to FFFF FFFFh).

Short-address mode allows the DTC to access to a 16-Mbyte space (0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh).

### 16.2.10 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 start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### DTCST Bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted.

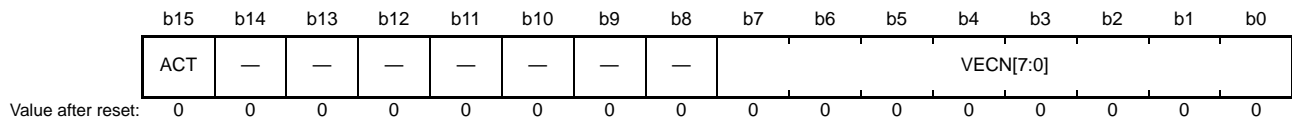
If this bit is set to 0 during data transfer, the accepted transfer request is active until the processing is completed.

Before making a transition to the module stop state or software standby mode, the DTCST bit must be set to 0.

For details on transitions to the module stop state and software standby mode, refer to section 16.9, Low Power Consumption Function, and section 11, Low Power Consumption.

### 16.2.11 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 activation 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 these bits 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 activation 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 activation sources and the vector addresses, refer to Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

#### ACT Flag (DTC Active Flag)

This flag 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.

### 16.3 Activation Sources

The DTC is activated by an interrupt request. Setting the DTCERn.DTCE bit (n = interrupt vector number) of the ICU to 1 selects the corresponding interrupt as an activation source for the DTC.

For the correspondence between the DTC activation sources and the vector addresses, refer to Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller (ICUb). For activation by software, refer to section 14.2.5, Software Interrupt Activation Register (SWINTR) in section 14, Interrupt Controller (ICUb).

Once the DTC has accepted an activation request, it does not accept another activation request until transfer for that single request is completed, regardless of the priority of the requests. When multiple activation requests are generated during DTC transfer, a request with the highest priority on completion of the transfer is accepted. When multiple activation requests are generated while the DTC module start bit (DTCST.DTCST) is 0, a request with the highest priority at the moment when the bit is subsequently set to 1 is accepted.

The DTC performs the following operations at the start of a single data transfer (or the last of the consecutive transfers in the case of a chain transfer).

- On completion of a specified round of data transfer, the DTCERn.DTCE bit is set to 0 and an interrupt is requested to the CPU.
- If the MRB.DISEL bit is 1, an interrupt is requested to the CPU on completion of data transfer.
- For the other transfers, the interrupt status flag of the activation source is set to 0 at the start of data transfer.

#### 16.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information corresponding to each activation source from the vector table and reads the transfer information starting at that address.

The vector table should be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC vector base register (DTCVBR) to set the base address of the DTC vector table.

Transfer information is allocated in the RAM area. In the RAM area, the start address of the transfer information (n) with vector number n should be  $4n$  added to the base address in the vector table.

Transfer information can be allocated in short-address mode (3 longwords) or full-address mode (4 longwords). Use the DTCADM.SHORT bit to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).

Figure 16.2 shows the relationship between the DTC vector table and transfer information.

Figure 16.3 shows the allocation of transfer information in the RAM area. The lower addresses vary according to the endian of the corresponding allocation area. For details, refer to section 16.10.2, Allocating Transfer Information.

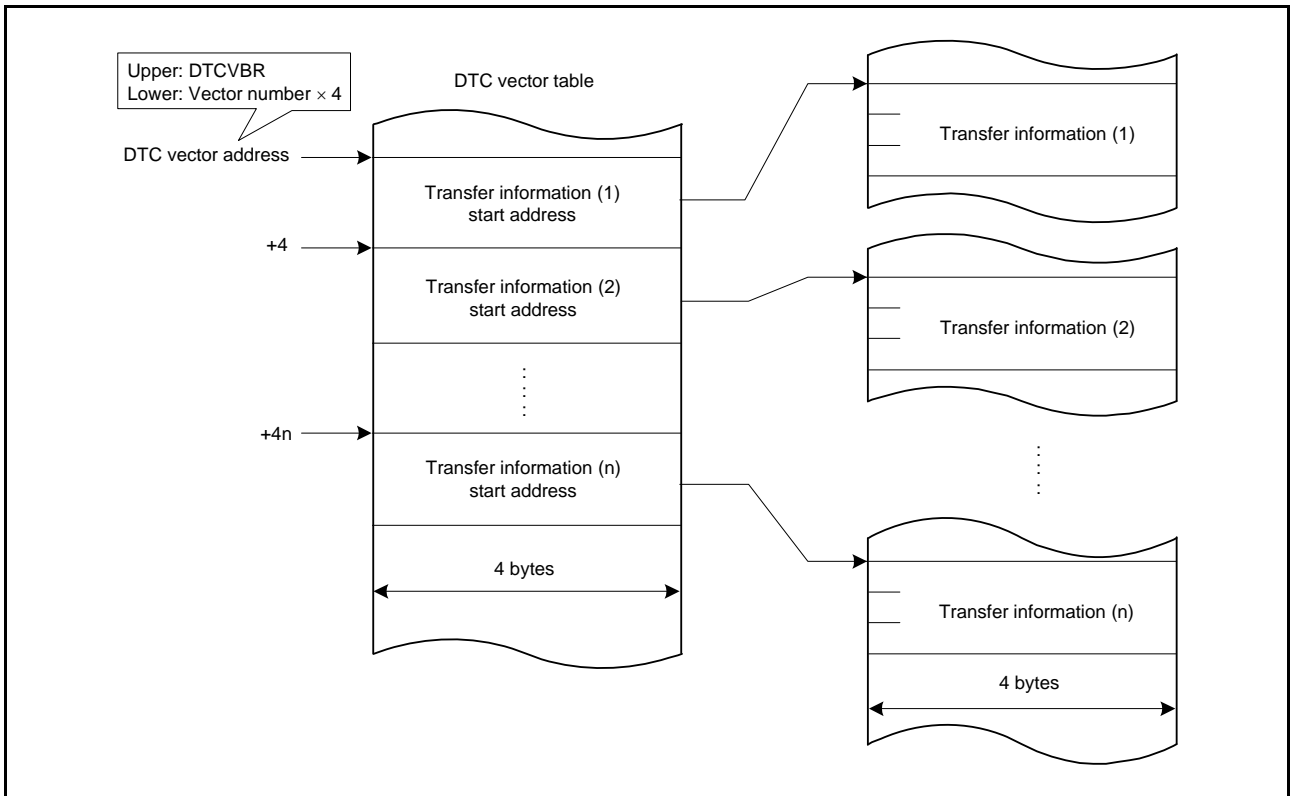


Figure 16.2 DTC Vector Table and Transfer Information

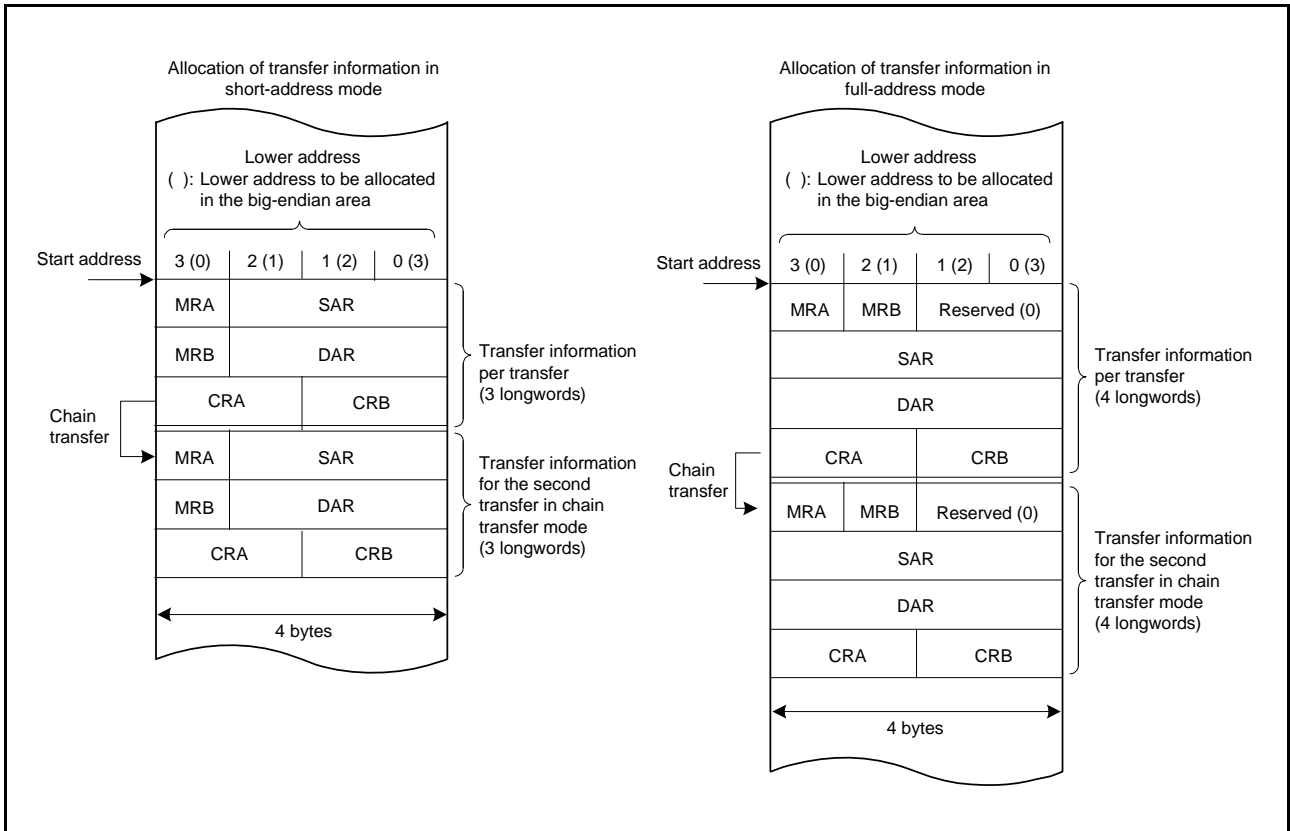


Figure 16.3 Allocation of Transfer Information in the RAM Area

## 16.4 Operation

The DTC transfers data in accordance with the transfer information. Storage of the transfer information in the RAM area is required before DTC operation.

When the DTC is activated, it reads the DTC vector corresponding to the vector number. Next, the DTC reads transfer information from the transfer information store address pointed by the DTC vector, transfers data, and then writes back the transfer information after the data transfer. Storing transfer information in the RAM area allows data transfer of arbitrary number of channels.

There are three transfer modes: normal transfer mode, repeat transfer mode, and block transfer mode.

The DTC specifies a transfer source address in the SAR register and a transfer destination address in the DAR register.

The values of these registers are incremented, decremented, or address-fixed independently after data transfer.

Table 16.2 lists transfer modes of the DTC.

**Table 16.2 Transfer Modes of the DTC**

Transfer Mode	Data Size Transferred on Single Transfer Request	Increment/Decrement of Memory Address	Settable Transfer Count
Normal transfer mode	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536
Repeat transfer mode*1	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes/1 to 256 words/1 to 256 longwords)	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536

Note 1. Set transfer source or transfer destination in the repeat area.

Note 2. Set transfer source or transfer destination in the block area.

Note 3. After data transfer of the specified count, the initial state is restored and the operation is continued (repeated).

Setting the MRB.CHNE bit to 1 allows multiple transfers (chain transfer) on a single activation source. Setting the MRB.CHNS bit also enables chain transfer when specified data transfer is completed.

Figure 16.4 shows the operation flowchart of the DTC. Table 16.3 lists chain transfer conditions.

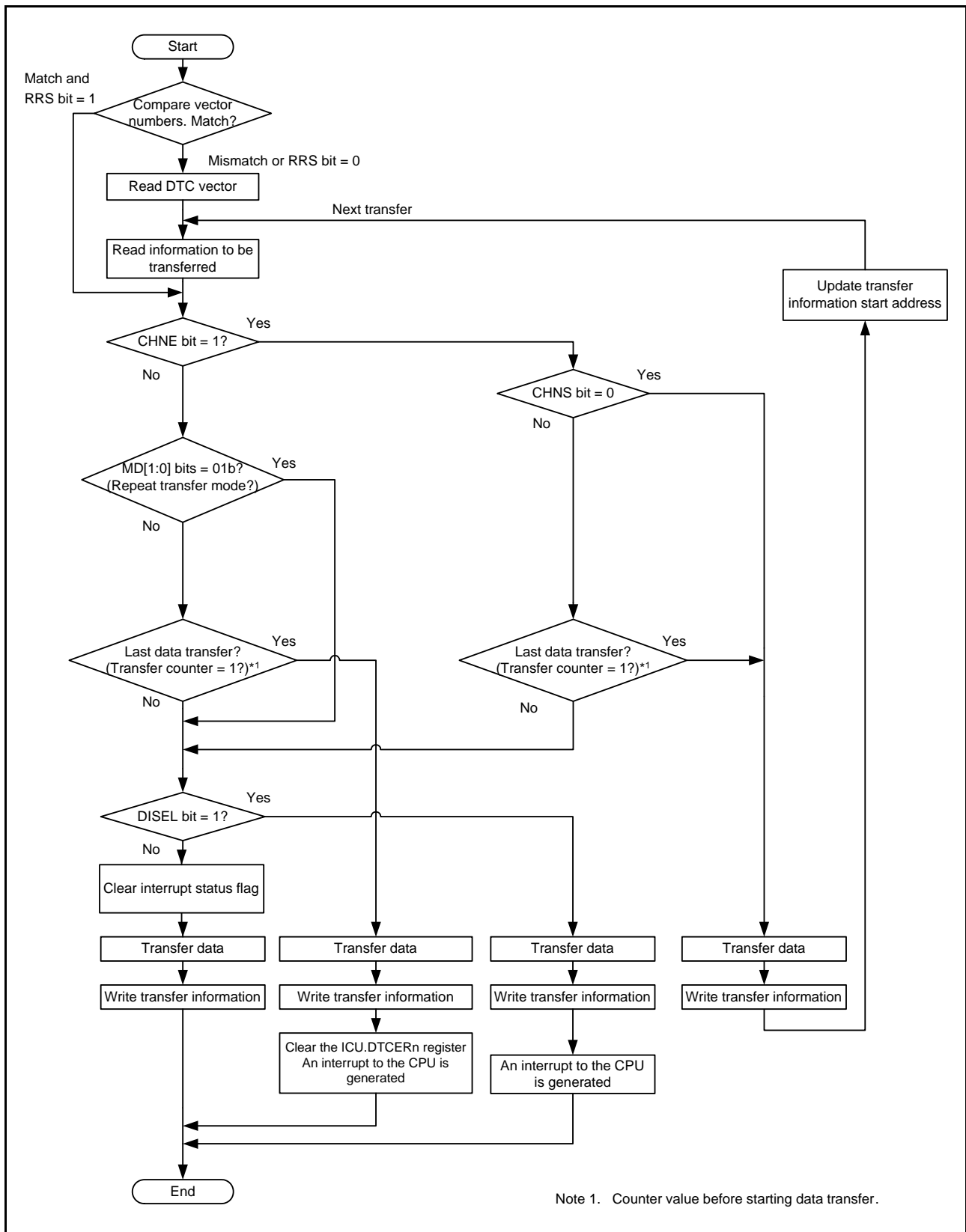


Figure 16.4 Operation Flowchart of the DTC



**Table 16.3 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.

### 16.4.1 Transfer Information Read Skip Function

Reading of vector addresses and transfer information can be skipped by the setting of the DTCCR.RRS bit.

When a DTC activation request is generated, the current DTC vector number is compared with the DTC vector number in the previous activation 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 information. However, when the previous transfer was chain transfer, the vector address and transfer information are read. Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, transfer information is read regardless of the value of the RRS bit. Figure 16.13 shows an example of transfer information read skip.

To update the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, and then set the RRS bit to 1. The retained vector number is discarded by setting the RRS bit to 0. The updated DTC vector table and transfer information are read in the next activation process.

### 16.4.2 Transfer Information Write-Back Skip Function

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to “address fixed”, a part of transfer information is not written back. This function is performed independently of the setting of short-address mode or full-address mode. Table 16.4 lists transfer information write-back skip conditions and applicable registers.

The CRA and CRB registers are written back independently of the setting of short-address mode or full-address mode. Furthermore, in full-address mode, write-back of the MRA and MRB registers are skipped.

**Table 16.4 Transfer Information Write-Back Skip Conditions and Applicable Registers**

MRA.SM[1:0] Bits		MRB.DM[1:0] Bits		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		

### 16.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single activation source. The transfer count can be set to 1 to 65536.

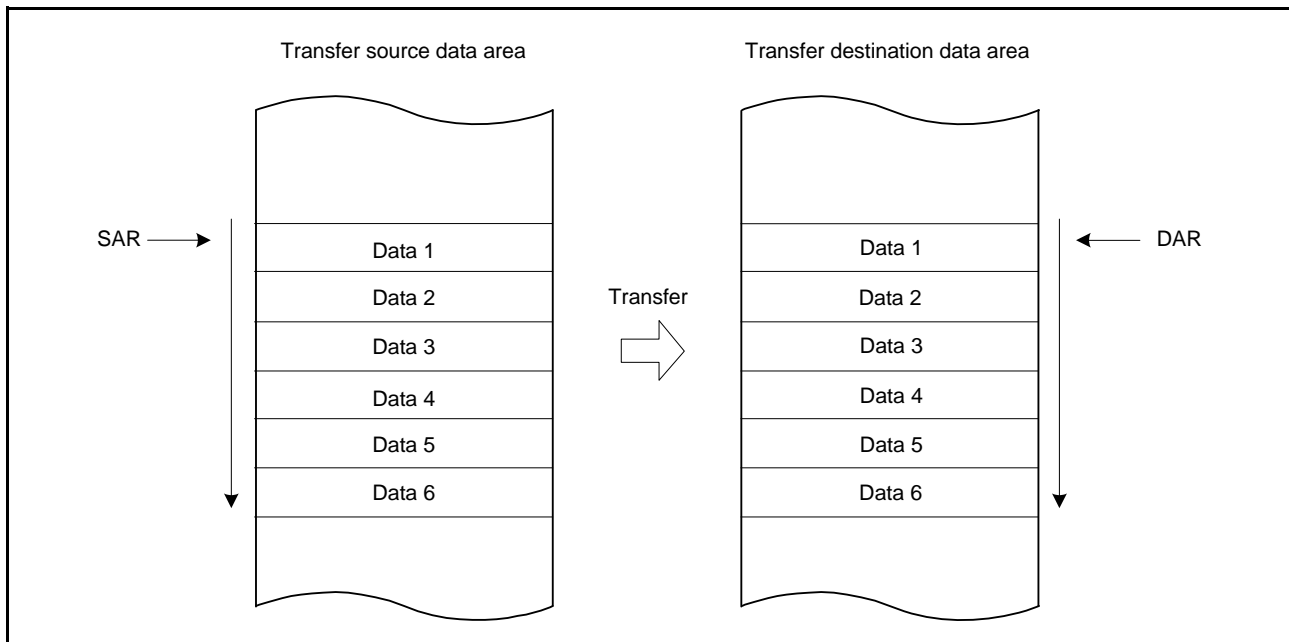
Transfer source addresses and transfer destination addresses can be set to increment, decrement, or fixed independently. This mode enables an interrupt request to the CPU to be generated at the end of specified-count transfer.

Table 16.5 lists register functions in normal transfer mode, and Figure 16.5 shows the memory map of normal transfer mode.

**Table 16.5 Register Functions in Normal Transfer Mode**

Register	Description	Value Written Back by Writing Transfer Information
SAR	Transfer source address	Increment/decrement/fixe* <sup>1</sup>
DAR	Transfer destination address	Increment/decrement/fixe* <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 16.5 Memory Map of Normal Transfer Mode**

### 16.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single activation source.

Specify either transfer source or transfer destination for the repeat area by the MRB.DTS bit. The transfer count can be set to 1 to 256. When the specified-count transfer is completed, the initial value of the address register specified in the transfer counter and the repeat area is restored and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.

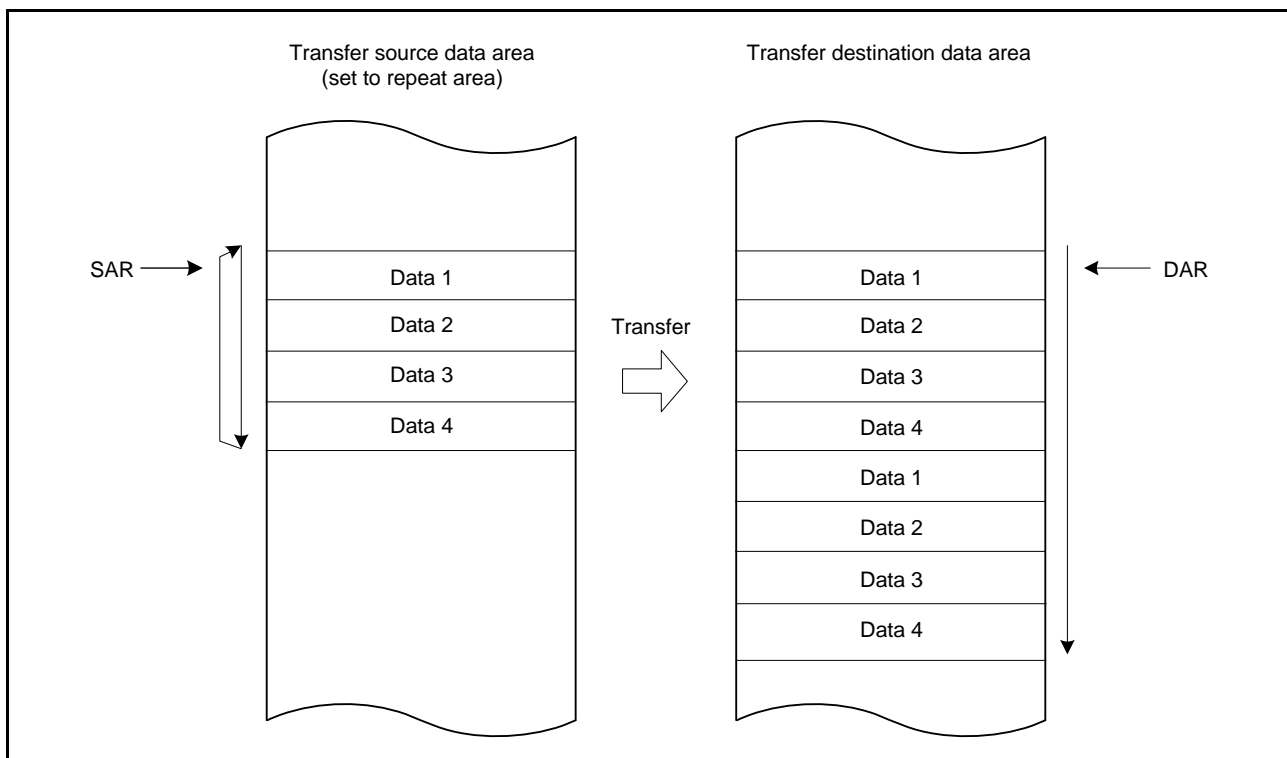
When the transfer counter CRAL is decreased to 00h in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. Thus the transfer counter does not become 00h, which disables an interrupt request to be generated to the CPU when the MRB.DISEL bit is set to 0 (an interrupt request to the CPU is generated when specified data transfer is completed).

Table 16.6 lists the register functions in repeat transfer mode, and Figure 16.6 shows the memory map of repeat transfer mode.

**Table 16.6 Register Functions in Repeat Transfer Mode**

Register	Description	Value Written Back by Writing Transfer Information	
		When CRAL is not 1	When CRAL is 1
SAR	Transfer source address	Increment/decrement/fix <sup>*1</sup>	(When the MRB.DTS bit is 0) Increment/decrement/fix <sup>*1</sup> (When the MRB.DTS bit is 1) SAR register initial value
DAR	Transfer destination address	Increment/decrement/fix <sup>*1</sup>	(When the MRB.DTS bit is 0) DAR register initial value (When the MRB.DTS bit is 1) Increment/decrement/fix <sup>*1</sup>
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 16.6 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)**

### 16.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single activation source.

Specify either transfer source or transfer destination for the block area by the MRB.DTS bit. The block size can be set to 1 to 256 bytes, 1 to 256 words, or 1 to 256 longwords.

When transfer of the specified one block is completed, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS bit = 1 or the DAR register when the DTS bit = 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.

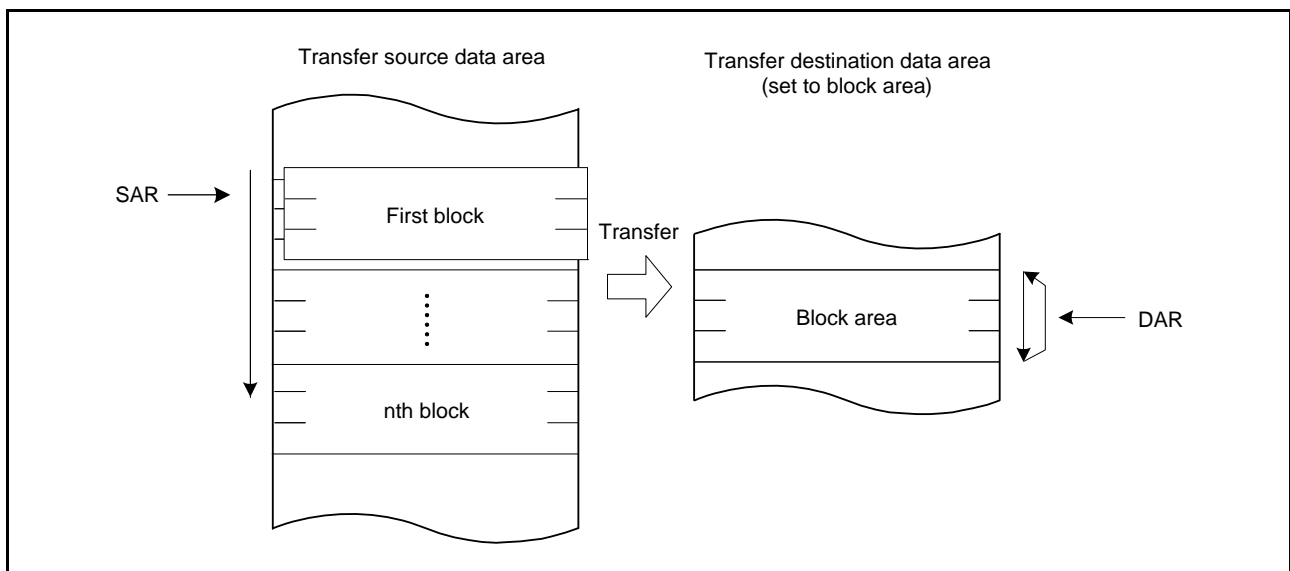
The transfer count (block count) can be set to 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of specified-count block transfer.

Table 16.7 lists register functions in block transfer mode, and Figure 16.7 shows the memory map of block transfer mode.

**Table 16.7 Register Functions in Block Transfer Mode**

Register	Description	Value Written Back by Writing Transfer Information
SAR	Transfer source address	(When MRB.DTS bit is 0) Increment/decrement/fix*1 (When MRB.DTS bit is 1) SAR register initial value
DAR	Transfer destination address	(When MRB.DTS bit is 0) DAR register initial value (When MRB.DTS bit 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.



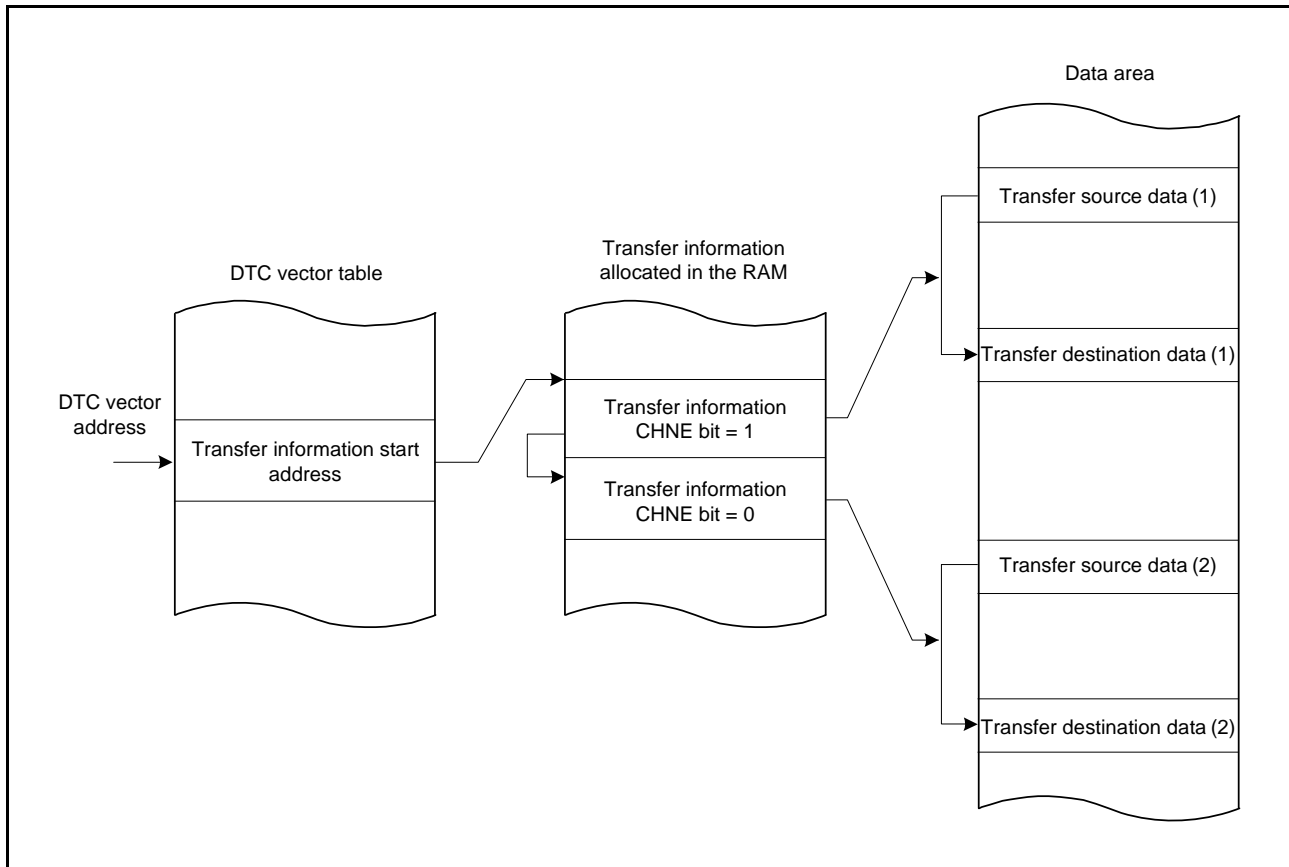
**Figure 16.7 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)**

### 16.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single activation source.

If the MRB.CHNE and CHNS bits are set to 1 and 0, respectively, an interrupt request to the CPU is not generated by completion of specified number of rounds of transfer or by setting the MRB.DISEL bit 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 of the activation source.

The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define data transfer. Figure 16.8 shows chain transfer operation.



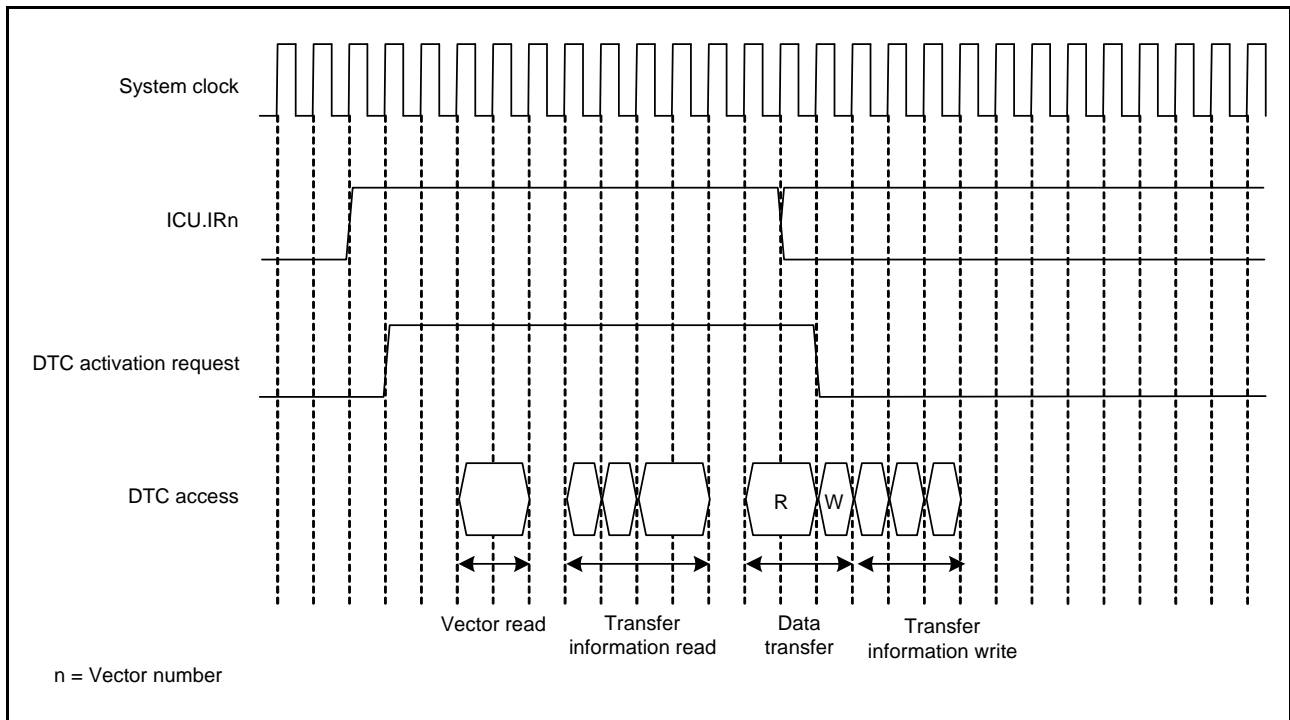
**Figure 16.8 Chain Transfer Operation**

Writing 1 to the MRB.CHNE and CHNS bits 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.

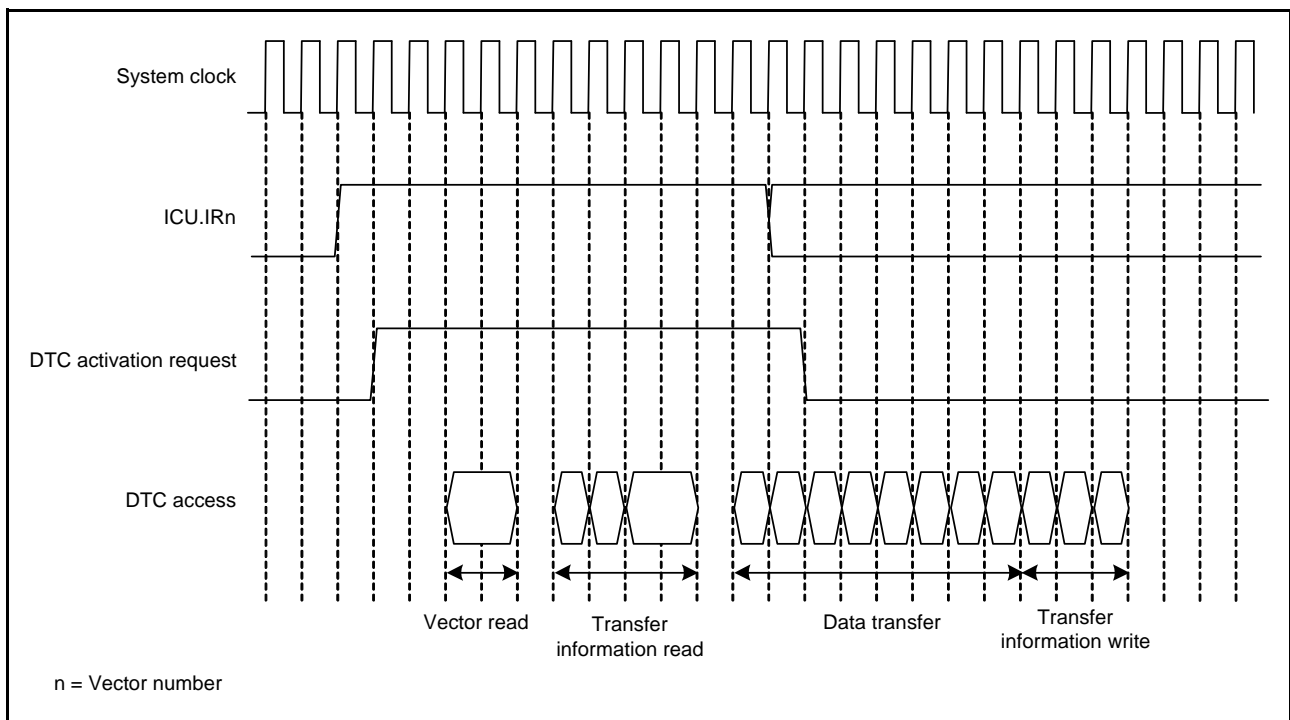
For details on chain transfer conditions, see Table 16.3, Chain Transfer Conditions.

### 16.4.7 Operation Timing

Figure 16.9 to Figure 16.13 show examples of DTC operation timing.



**Figure 16.9 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)**



**Figure 16.10 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)**

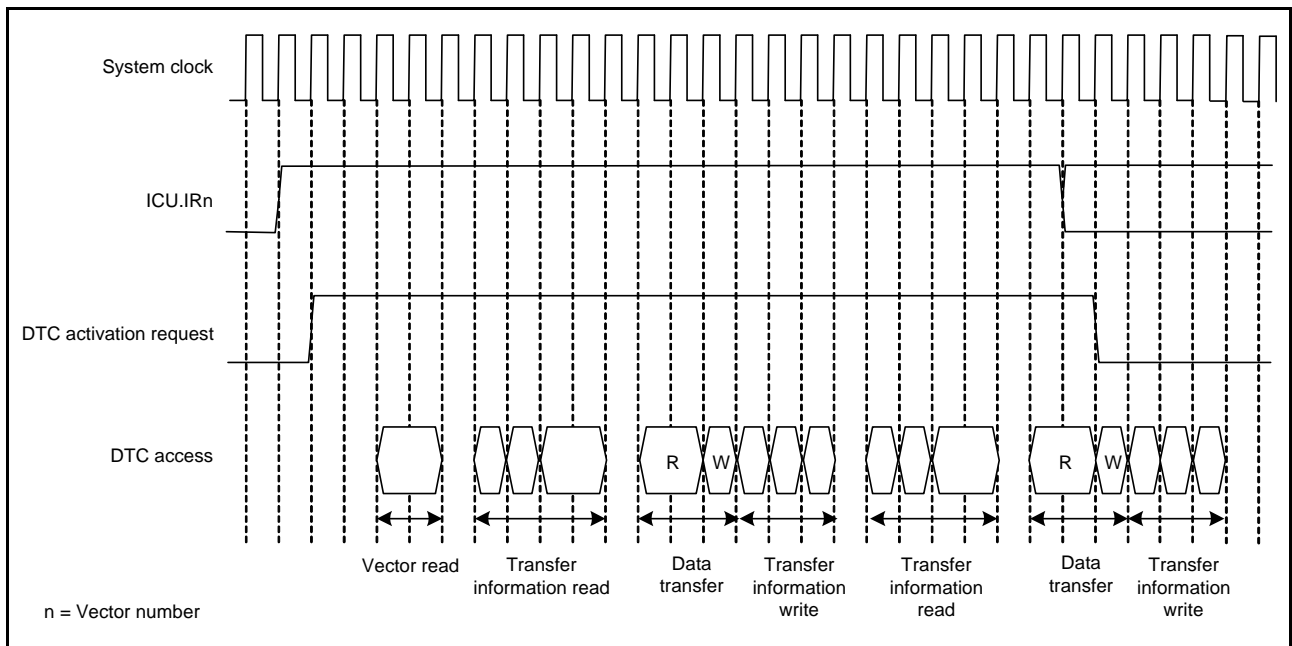


Figure 16.11 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

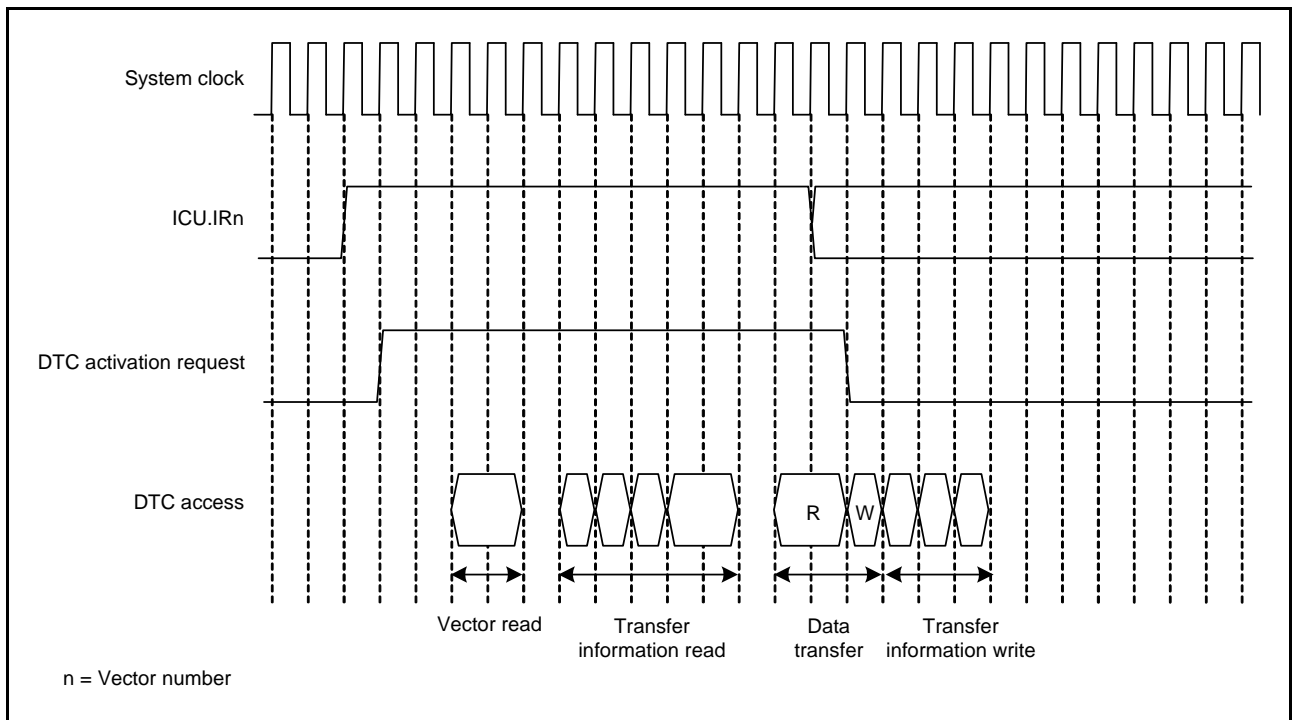
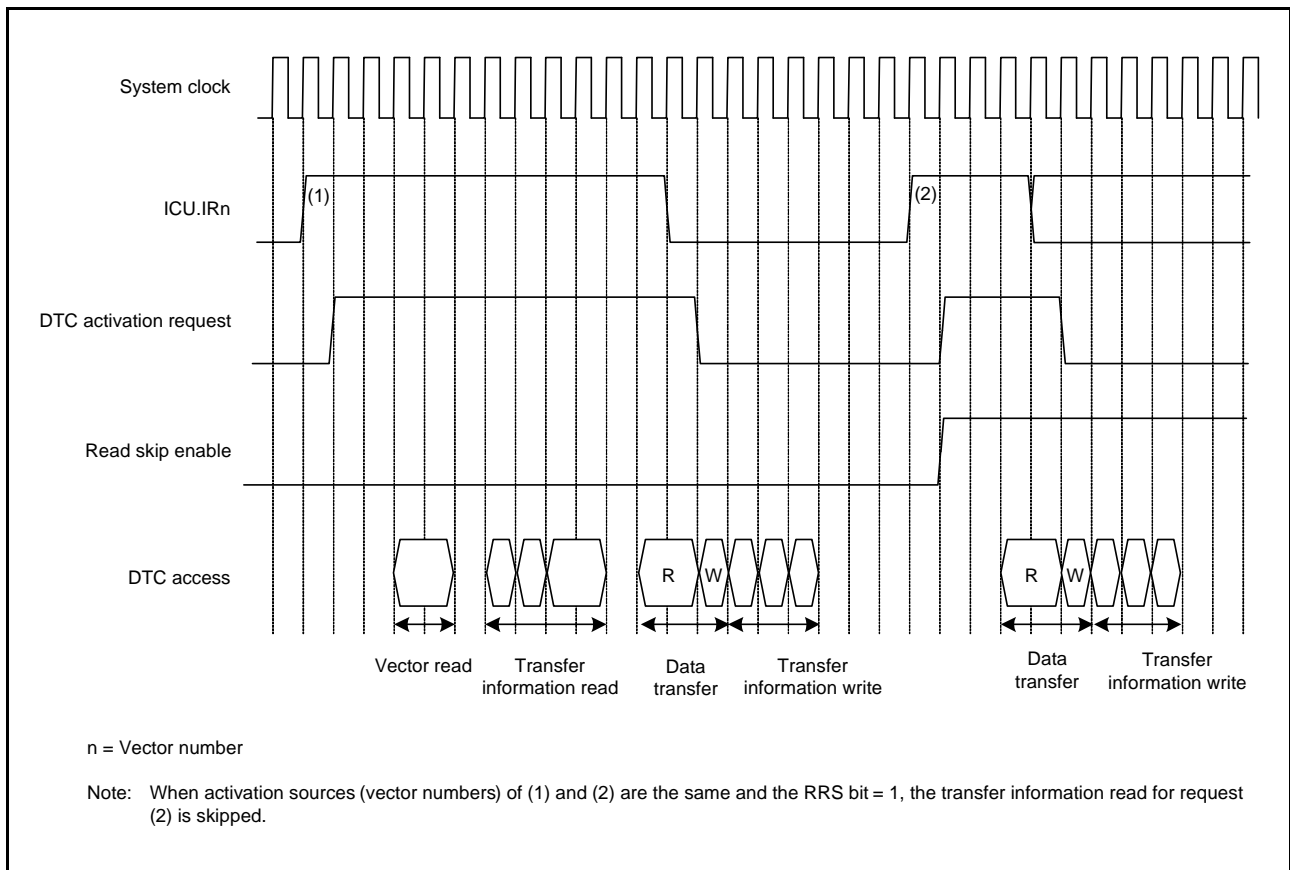


Figure 16.12 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)





**Figure 16.13 Example of Operation When Transfer Information Read Skip is Executed (Vector, Transfer Information, and Transfer Destination Data on the RAM, and Transfer Source Data on the Peripheral Module)**

### 16.4.8 Execution Cycles of the DTC

Table 16.8 lists the execution cycles of single data transfer of the DTC.

For the order of the execution states, refer to section 16.4.7, Operation Timing.

**Table 16.8 Execution Cycles of the DTC**

Transfer Mode	Vector Read		Transfer Information Read			Transfer Information Write			Data Transfer		Internal Operation	
	Read	Write	Read	Write	Write	Read	Write	Read	Write	Read	Write	
Normal	$C_v + 1$	$0^{*1}$	$4 \times C_i + 1^{*2}$	$3 \times C_i + 1^{*3}$	$0^{*1}$	$3 \times C_i^{*4}$	$2 \times C_i^{*5}$	$C_i^{*6}$	$C_r + 1$	$C_w$	2	$0^{*1}$
Repeat									$C_r + 1$	$C_w$		
Block <sup>*7</sup>									$P \times C_r$	$P \times C_w$		

Note 1. When transfer information read is skipped

Note 2. In full-address mode

Note 3. In short-address mode

Note 4. When neither SAR nor DAR is set to address-fixed 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 more. If the block size is 1, the cycle number for normal transfer is applied.

P: Block size (initial settings of CRAH and CRAL)

$C_v$ : Cycles for access to vector transfer information storage destination

$C_i$ : Cycles for access to transfer information storage destination address

$C_r$ : Cycles for access to data read destination

$C_w$ : Cycles for access to data write destination

(The unit is system clocks (ICLK) for "+ 1" in the Vector Read, Transfer Information Read, and Data Transfer Read columns and "2" in the Internal Operation column.)

( $C_v$ ,  $C_i$ ,  $C_r$ , and  $C_w$  vary depending on the corresponding access destination. For the number of cycles for respective access destinations, refer to section 40, RAM, section 41, Flash Memory, and section 5, I/O Registers.)

### 16.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer information read and transfer information write. While transfer information is not read or written, bus arbitration is made according to the priority determined by the bus master arbitrator.

For bus arbitration, refer to section 15, Buses.

## 16.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR).

Figure 16.14 shows the procedure to set the DTC.

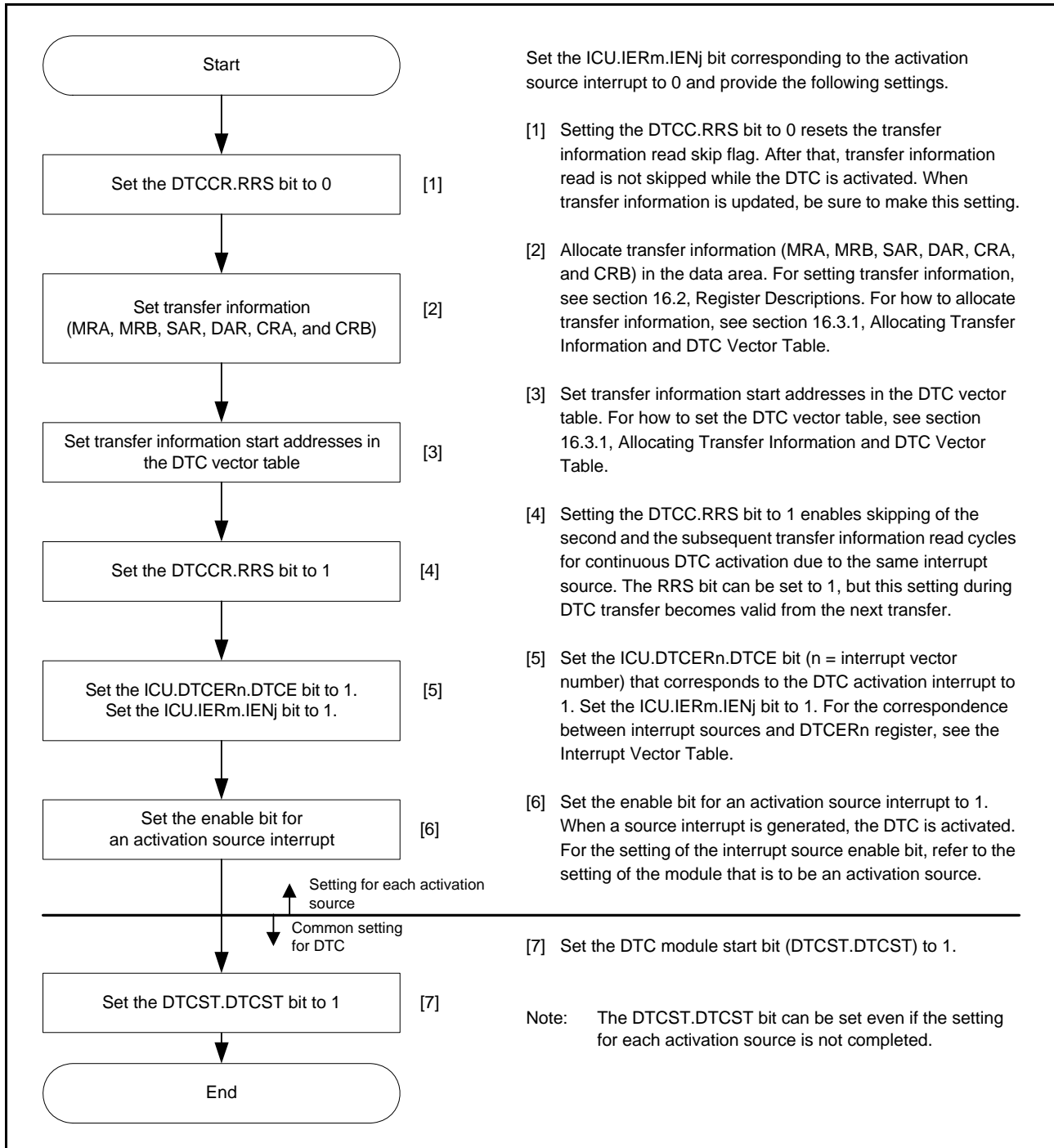


Figure 16.14 Procedure to Set the DTC

## 16.6 Examples of DTC Usage

### 16.6.1 Normal Transfer

As an example of DTC usage, its employment in the reception of 128 bytes of data by an SCI is described below.

#### (1) Transfer Information Setting

In the MRA register, select a fixed source address (MRA.SM[1:0] bits = 00b), normal transfer mode (MRA.MD[1:0] bits = 00b), and byte-sized transfer (MRA.SZ[1:0] bits = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] bits = 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 RDR register address of the SCI in the SAR register, the start address of the RAM area for data storage in the DAR register, and 128 (0080h) in the CRA register. The CRB register can be set to any value.

#### (2) DTC Vector Table Setting

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

#### (3) ICU Setting and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to 1 and the ICU.IERi.IENj bit to 1.  
Set the DTCST.DTCST bit to 1.

#### (4) SCI Setting

Enable the RXI interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, further reception is not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

#### (5) DTC Transfer

Every time the reception of 1 byte by the SCI is completed, an RXI interrupt is generated to activate the DTC. The DTC transfers the received byte from the RDR of the SCI to RAM, after which the DAR register is incremented and the CRA register is decremented.

#### (6) Interrupt Handling

After 128 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. Complete the process in the handling routine for this interrupt.

## 16.6.2 Chain Transfer When the Counter = 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second transfer. Repeating this chain transfer enables transfers to be repeated 256 times or more.

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 16.15 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, the CRA register = 0000h (65,536 times), the MRB.CHNE bit = 1 (chain transfer is enabled), the MRB.CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0), and the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed).
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 20 0000h to 21 FFFFh, 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 the DAR register in the first transfer information area for the transfer destination. At this time, set the MRB.CHNE bit = 0 (chain transfer is disabled) and the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed). When setting the input buffer mentioned above to 20 0000h to 21 FFFFh, 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.

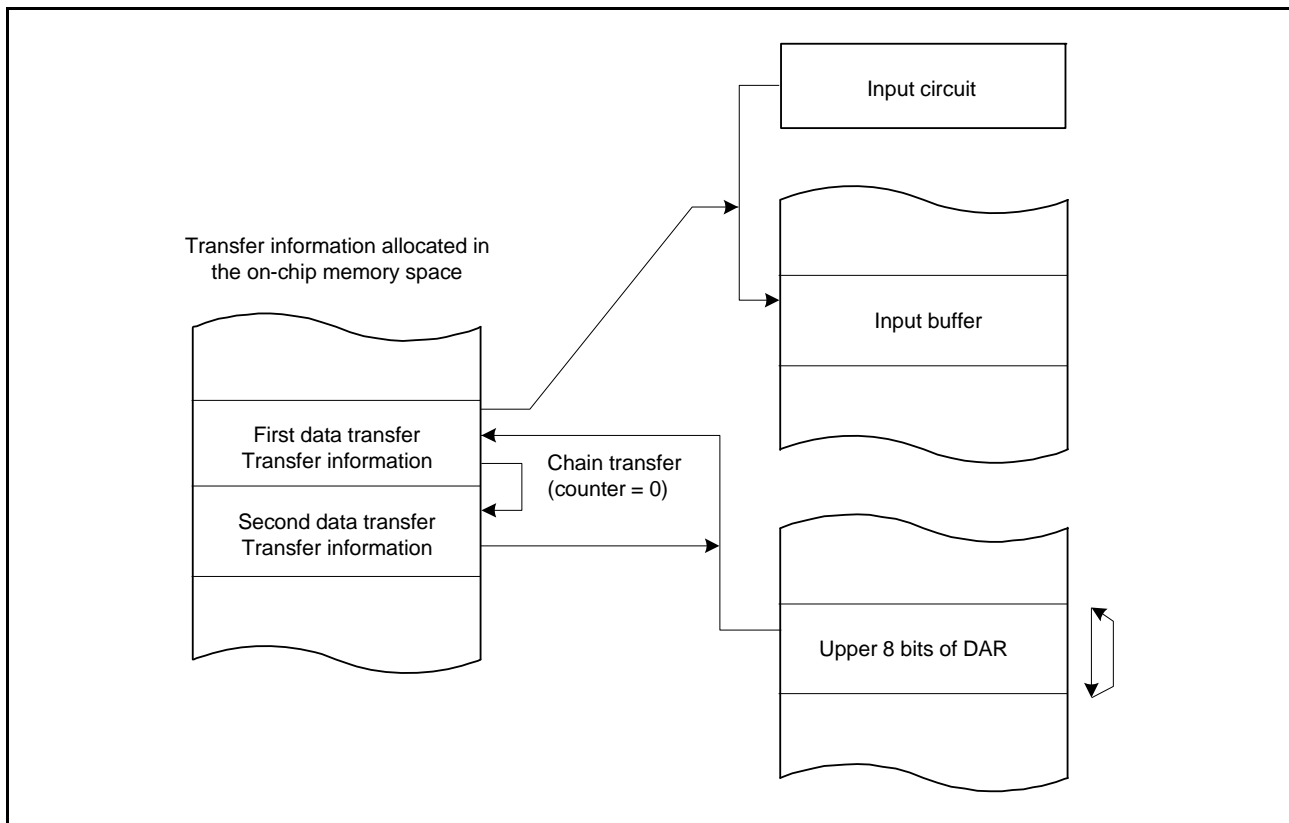


Figure 16.15 Chain Transfer When the Counter = 0

## 16.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the MRB.DISEL 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 activation source. Such interrupts to the CPU are controlled according to the PSW.I bit (interrupt enable) of the CPU, the PSW.IPL[3:0] bits (processor interrupt priority level), and the priority level of the interrupt controller.

## 16.8 Event Link

The DTC is capable of producing an event link request on completing transfer in response to one request.

## 16.9 Low Power Consumption Function

Before making a transition to the module stop state or software standby mode, set the DTCST.DTCST bit to 0 (DTC module stop), and then perform the following.

### (1) Module Stop Function

Writing 1 (this module clock is enabled) to the MSTPCRA.MSTPA28 bit enables the module stop function of the DTC. If DTC transfer is in progress at the time 1 is written to the MSTPCRA.MSTPA28 bit, the transition to the module stop state proceeds after DTC transfer has ended. While the MSTPCRA.MSTPA28 bit is 1, accessing the DTC registers is prohibited.

Writing 0 (this module clock is disabled) to the MSTPCRA.MSTPA28 bit releases the DTC from the module stop state.

### (2) Software Standby Mode

Make settings according to the procedure under section 11.6.3.1, Entry to Software Standby Mode, in section 11, Low Power Consumption.

If DTC transfer operations are in progress at the time the WAIT instruction is executed, the transition to software standby mode follows the completion of DTC transfer.

### (3) Notes on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, refer to section 11.7.5, Timing of WAIT Instructions in section 11, Low Power Consumption.

To perform DTC transfer after returning from a low power consumption mode, set the DTCST.DTCST bit to 1 again.

To use a request that is generated in software standby mode as an interrupt request to the CPU but not as a DTC activation request, specify the CPU as the interrupt request destination according to the description in section 14.4.3, Selecting Interrupt Request Destinations in section 14, Interrupt Controller (ICUb), and then execute the WAIT instruction.

## 16.10 Usage Notes

### 16.10.1 Transfer Information Start Address

Be sure to set multiples of 4 for the transfer information start addresses in the vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

### 16.10.2 Allocating Transfer Information

Allocate transfer data in the memory area according to the endian of the area as shown in Figure 16.16.

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.

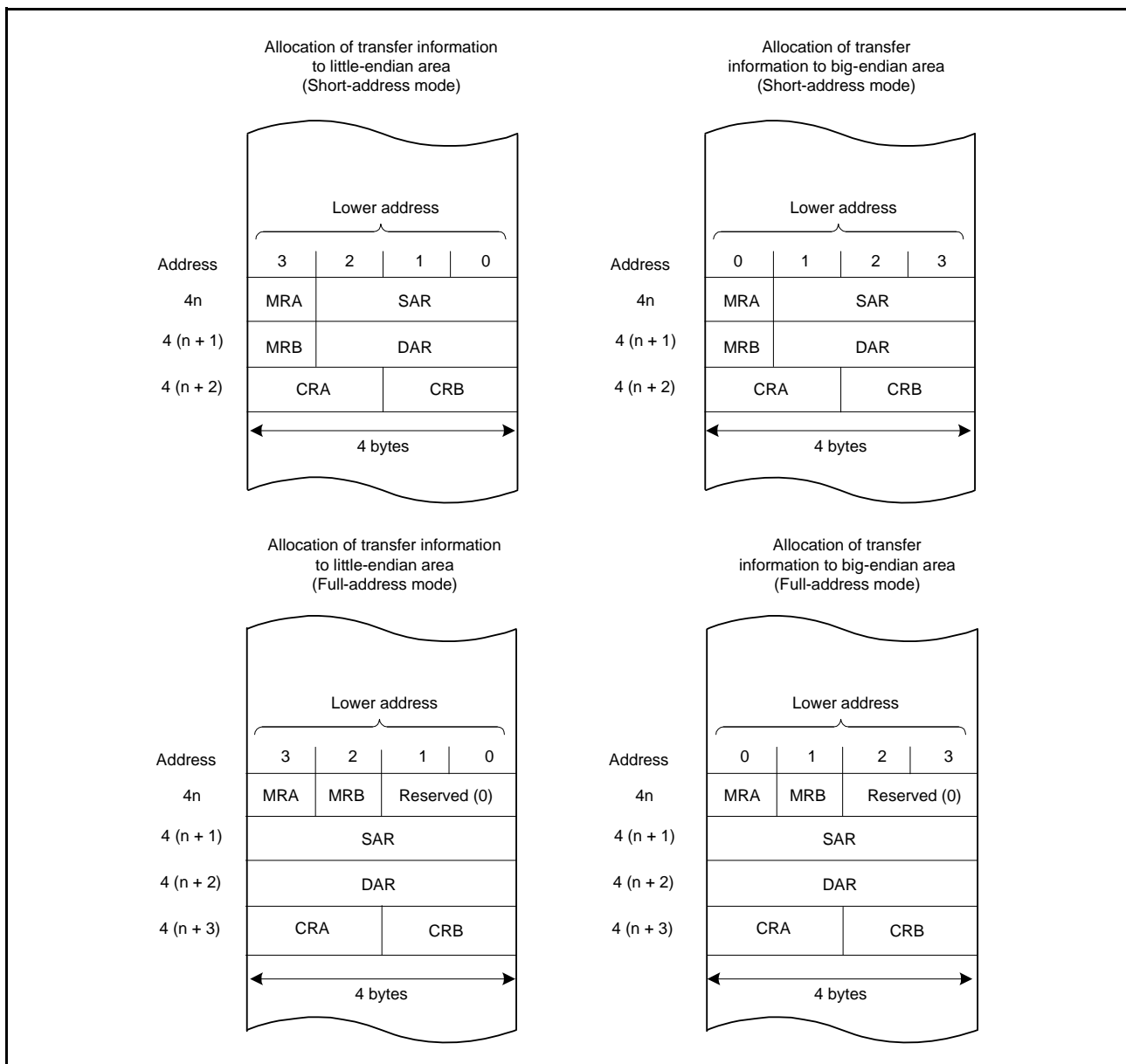


Figure 16.16 Allocation of Transfer Information



## 17. Event Link Controller (ELC)

### 17.1 Overview

The event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals to connect (link) them to different modules, allowing direct cooperation between the modules without CPU intervention. Event signals can be output regardless of the setting of the corresponding interrupt request enable bit.

Table 17.1 lists the specifications of the ELC, and Figure 17.1 shows a block diagram of the ELC.

**Table 17.1 ELC Specifications**

Item	Description
Event link function	<ul style="list-style-type: none"> <li>• 44 types of event signals can be directly connected to modules.</li> <li>• The operation of timer modules can be selected when an event is input to the timer module.</li> <li>• Event link operation is possible for port B.               <ul style="list-style-type: none"> <li>Single port*1: An event link can be set for a single bit specified in a port.</li> <li>Port group*1: An event link can be set for a group of single bits specified within eight I/O ports.</li> </ul> </li> </ul>
Low power consumption function	Module stop state can be set.

Note 1. The single port and port group specified as the input generate an event according to the change in the connected signal value. In products with 64-pin packages, when ports PC0 and PC1 are selected in port switching register A (PSRA), ports PB6 and PB7 of the ELC cannot be used as input and output events.

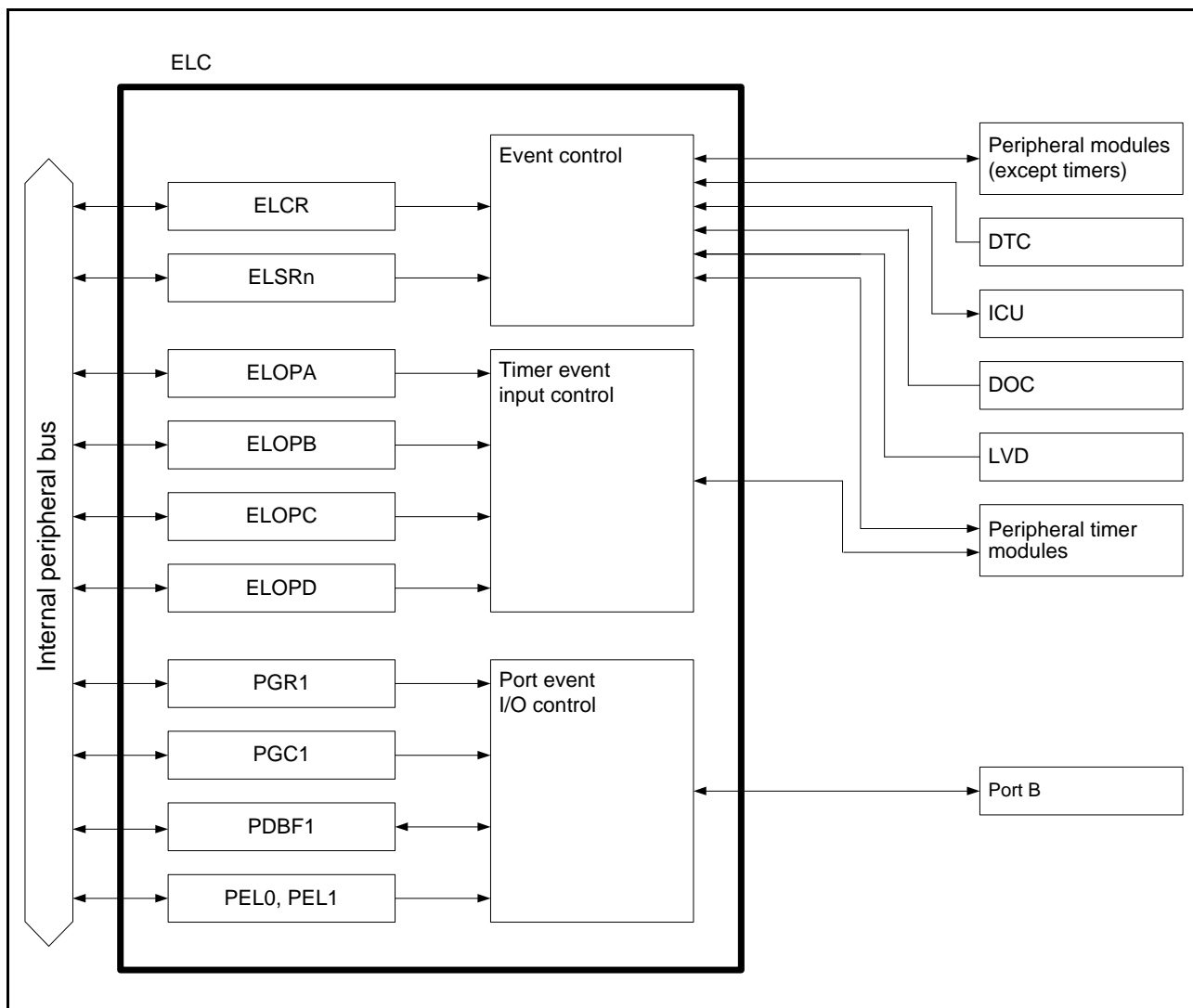
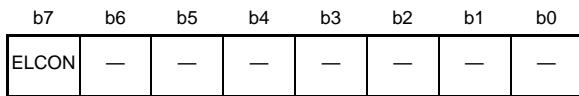


Figure 17.1 ELC Block Diagram (n = 1 to 4, 7, 10, 12, 14, 15, 17, 18, 20, 22, 24, 25)

## 17.2 Register Descriptions

### 17.2.1 Event Link Control Register (ELCR)

Address(es): 0008 B100h

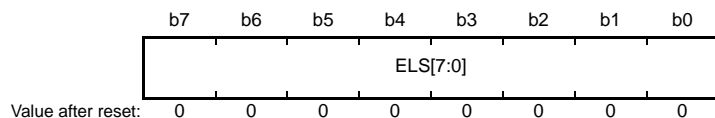


Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	ELCON	All Event Link Enable	0: ELC function is disabled. 1: ELC function is enabled	R/W

The ELCR register controls operation of the ELC.

### 17.2.2 Event Link Setting Register n (ELSRn) (n = 1 to 4, 7, 10, 12, 14, 15, 17, 18, 20, 22, 24, 25)

Address(es): ELSR1 0008 B102h, ELSR2 0008 B103h, ELSR3 0008 B104h, ELSR4 0008 B105h,  
 ELSR7 0008 B108h, ELSR10 0008 B10Bh, ELSR12 0008 B10Dh, ELSR14 0008 B10Fh,  
 ELSR15 0008 B110h, ELSR17 0008 B112h, ELSR18 0008 B113h, ELSR20 0008 B115h,  
 ELSR22 0008 B117h, ELSR24 0008 B119h, ELSR25 0008 B11Ah



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ELS[7:0]	Event Link Select	b7 b0 00000000: Event output to the corresponding peripheral module is disabled. 00001000 to 01101010: Set the number for the event signal to be linked. Settings other than above are prohibited.	R/W

The ELSRn register specifies an event signal to be linked to for each peripheral module. Table 17.2 shows the correspondence between the ELSRn register and the peripheral modules. Table 17.3 shows the correspondence between the event signal names set in the ELSRn register and the signal numbers.

**Table 17.2 Correspondence between the ELSRn Register and the Peripheral Functions**

Register Name	Peripheral Function (Module)
ELSR1	MTU1
ELSR2	MTU2
ELSR3	MTU3
ELSR4	MTU4
ELSR7	CMT1
ELSR10	TMR0
ELSR12	TMR2
ELSR14	CTSU
ELSR15	12-bit A/D converter
ELSR17	DA0
ELSR18	Interrupt 1
ELSR20	Output port group 1
ELSR22	Input port group 1
ELSR24	Single port 0*1
ELSR25	Single port 1*1

Note 1. Do not set the DOC data operation condition met signal (ELS[7:0] bits = 6Ah) in the ELSR24 or ELSR25 register.

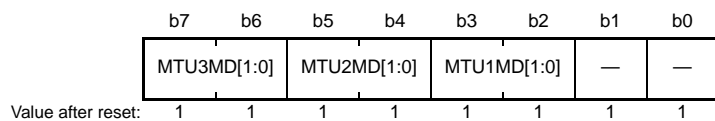
**Table 17.3 Correspondence between Event Signal Names Set in ELSRn.ELS[7:0] Bits and Signal Numbers**

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
08h	Multifunction timer pulse unit 2	MTU1 compare match 1A
09h		MTU1 compare match 1B
0Ah		MTU1 overflow
0Bh		MTU1 underflow
0Ch		MTU2 compare match 2A
0Dh		MTU2 compare match 2B
0Eh		MTU2 overflow
0Fh		MTU2 underflow
10h		MTU3 compare match 3A
11h		MTU3 compare match 3B
12h		MTU3 compare match 3C
13h		MTU3 compare match 3D
14h		MTU3 overflow
15h		MTU4 compare match 4A
16h		MTU4 compare match 4B
17h		MTU4 compare match 4C
18h		MTU4 compare match 4D
19h		MTU4 overflow
1Ah		MTU4 underflow
1Fh	Compare match timer	CMT1 compare match 1
22h	8-bit timers	TMR0 compare match A0
23h		TMR0 compare match B0
24h		TMR0 overflow
28h		TMR2 compare match A2
29h		TMR2 compare match B2
2Ah		TMR2 overflow
3Ah	Serial communications interfaces	SCI5 error (receive error or error signal detection)
3Bh		SCI5 receive data full
3Ch		SCI5 transmit data empty
3Dh		SCI5 transmit end
4Eh	I <sup>2</sup> C bus interface	RIIC0 communication error or event generation
4Fh		RIIC0 receive data full
50h		RIIC0 transmit data empty
51h		RIIC0 transmit end
58h	12-bit A/D converter	A/D conversion end of 12-bit A/D converter
59h	Comparator B0	Comparison result change of comparator B0
5Ah	Comparator B0/B1	Comparison result change of comparator B0/B1
5Bh	Voltage detection circuit	LVD1 voltage detection
61h	Data transfer controller	DTC transfer end
63h	I/O ports	Input edge detection of input port group 1
65h		Input edge detection of single input port 0
66h		Input edge detection of single input port 1
69h	Event link controller	Software event
6Ah	Data operation circuit	DOC data operation condition met signal

Settings other than above are prohibited.

### 17.2.3 Event Link Option Setting Register A (ELOPA)

Address(es): 0008 B11Fh



Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b3, b2	MTU1MD[1:0]	MTU1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture <sup>*1</sup> 1 1: Event is disabled.	R/W
b5, b4	MTU2MD[1:0]	MTU2 Operation Select	b5 b4 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture <sup>*2</sup> 1 1: Event is disabled.	R/W
b7, b6	MTU3MD[1:0]	MTU3 Operation Select	b7 b6 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture <sup>*3</sup> 1 1: Event is disabled.	R/W

Note 1. The MTU1.TCNT value is captured into MTU1.TGRA.

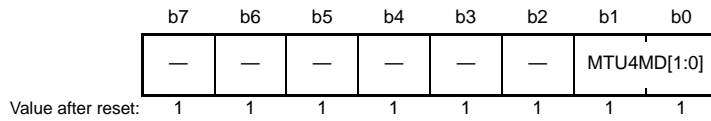
Note 2. The MTU2.TCNT value is captured into MTU2.TGRA.

Note 3. The MTU3.TCNT value is captured into MTU3.TGRA.

ELOPA determines the operation of MTU1 to MTU3 when an event is input. The event setting should be disabled when the ELC function is not to be used.

### 17.2.4 Event Link Option Setting Register B (ELOPB)

Address(es): 0008 B120h



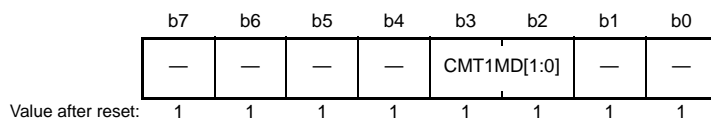
Bit	Symbol	Bit Name	Description	R/W
b1, b0	MTU4MD[1:0]	MTU4 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event is disabled.	R/W
b7 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MTU4.TCNT value is captured into MTU4.TGRA.

ELOPB determines the operation of MTU4 when an event is input. The event setting should be disabled when the ELC function is not to be used.

### 17.2.5 Event Link Option Setting Register C (ELOPC)

Address(es): 0008 B121h

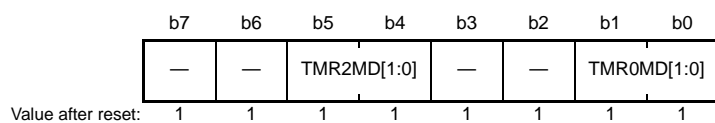


Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b3, b2	CMT1MD[1:0]	CMT1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event is disabled.	R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ELOPC determines the operation of CMT1 when an event is input. The event setting should be disabled when the ELC function is not to be used.

## 17.2.6 Event Link Option Setting Register D (ELOPD)

Address(es): 0008 B122h



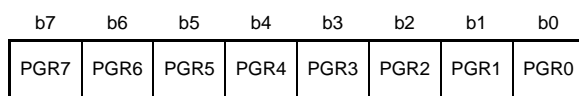
Bit	Symbol	Bit Name	Description	R/W
b1, b0	TMR0MD[1:0]	TMR0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event is disabled.	R/W
b3, b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5, b4	TMR2MD[1:0]	TMR2 Operation Select	b5 b4 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event is disabled.	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ELOPD determines the operation of TMR0 and TMR2 when an event is input. The event setting should be disabled when the ELC function is not to be used.



## 17.2.7 Port Group Setting Register 1 (PGR1)

Address(es): PGR1 0008 B123h



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

Bit	Symbol	Bit Name	Description	R/W
b0	PGR0	Port Group Setting 0	0: The port bit is not specified as a member of the same group. 1: The port bit is specified as a member of the same group.	R/W
b1	PGR1	Port Group Setting 1		R/W
b2	PGR2	Port Group Setting 2		R/W
b3	PGR3	Port Group Setting 3		R/W
b4	PGR4	Port Group Setting 4		R/W
b5	PGR5	Port Group Setting 5		R/W
b6	PGR6	Port Group Setting 6		R/W
b7	PGR7	Port Group Setting 7		R/W

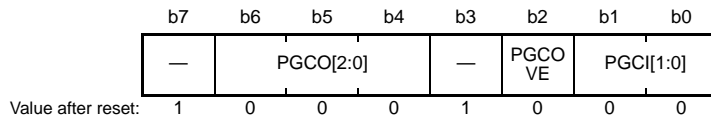
PGR1 specifies a group for I/O port bits. PGR1 specifies each port bit in the same eight I/O ports as the member of a group. One to eight port bits can be specified as the members of the same group as required. Table 17.4 shows the PGR1 register and corresponding port.

**Table 17.4 Registers Related to Port Groups and Corresponding Port Numbers**

Port Number	Port Group Setting Register (PGR)	Port Group Control Register (PGC)	Port Buffer Register (PDBF)
Port B	PGR1 register	PGC1 register	PDBF1 register

## 17.2.8 Port Group Control Register 1 (PGC1)

Address(es): PGC1 0008 B125h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	PGCI[1:0]	Event Output Edge Select	b1 b0 0 0: Event is generated upon detection of the rising edge of the external input signal. 0 1: Event is generated upon detection of the falling edge of the external input signal. 1 x: Event is generated upon detection of both the rising and falling edges of the external input signal.	R/W
b2	PGCOVE	PDBF Overwrite	0: Overwriting PDBF1 register is disabled. 1: Overwriting PDBF1 register is enabled.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6 to b4	PGCO[2:0]	Port Group Operation Select	b6 b4 0 0 0: 0 is output when the event is input. 0 0 1: 1 is output when the event is input. 0 1 0: The toggled (inverted) value is output when the event is input. 0 1 1: The buffer value is output when the event is input. 1 x x: The bit value is rotated out in the group (from MSB to LSB) when the event is input.	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

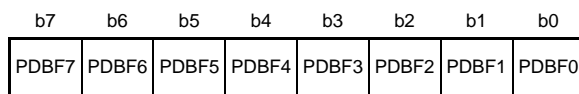
x: Don't care

For the output port group, PGC1 specifies the form of outputting the signal externally via the port when the event signal is input. For the input port group, PGC1 enables/disables overwriting of PDBF and specifies the conditions of event generation (edge of the externally input signal).

Refer to Table 17.4 for the PGC1 register and corresponding port.

## 17.2.9 Port Buffer Register 1 (PDBF1)

Address(es): PDBF1 0008 B127h



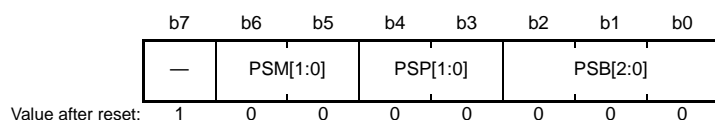
Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	PDBF0	Port Buffer 0	Data is transferred between PODR and PDBF when an event is input.	R/W
b1	PDBF1	Port Buffer 1	Write access to the bit specified as a member of the input port group is disabled. For details, refer to section 17.3, Operation.	R/W
b2	PDBF2	Port Buffer 2		R/W
b3	PDBF3	Port Buffer 3		R/W
b4	PDBF4	Port Buffer 4		R/W
b5	PDBF5	Port Buffer 5		R/W
b6	PDBF6	Port Buffer 6		R/W
b7	PDBF7	Port Buffer 7		R/W

PDBF1 is an 8-bit readable/writable register used in combination with PGR1. Refer to section 17.3.5, I/O Port Operation upon Event Input and Event Generation for PDBF1 operations. Refer to Table 17.4 for the PDBF1 register and corresponding port.

### 17.2.10 Event Link Port Setting Register n (PEL<sub>n</sub>) (n = 0, 1)

Address(es): PEL0 0008 B129h, PEL1 0008 B12Ah



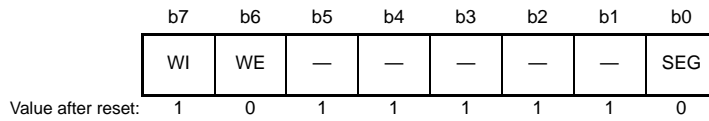
Bit	Symbol	Bit Name	Description	R/W
b2 to b0	PSB[2:0]	Bit Number Specification	A bit number in eight I/O ports is specified.	R/W
b4, b3	PSP[1:0]	Port Number Specification	<div style="margin-left: 20px;">                     b4 b3                      0 0: Setting disabled                      0 1: Port B (corresponding to PGR1)                      1 x: Setting prohibited                 </div>	R/W
b6, b5	PSM[1:0]	Event Link Specification	<ul style="list-style-type: none"> <li>• For the output port, data to be output from the port is specified.                             <div style="margin-left: 20px;">                                 b6 b5                                  0 0: 0 is output when the event is input.                                  0 1: 1 is output when the event is input.                                  1 x: The toggled (inverted) value is output when the event is input.                             </div> </li> <li>• For the input port, the edge on which the event is to be output is specified.                             <div style="margin-left: 20px;">                                 b6 b5                                  0 0: Event is output upon detection of the rising edge.                                  0 1: Event is output upon detection of the falling edge.                                  1 x: Event is output upon detection of both the rising and falling edges.                             </div> </li> </ul>	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

x: Don't care

PEL<sub>n</sub> specifies the single port to which an event is to be linked, the port operation upon the event signal input, and the conditions of event generation. In this MCU, a total of 2 bits in port B can be specified as single ports.

### 17.2.11 Event Link Software Event Generation Register (ELSEGR)

Address(es): 0008 B12Dh



Bit	Symbol	Bit Name	Description	R/W
b0	SEG	Software Event Generation	0: Normal operation 1: Software event is generated.	W
b5 to b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b6	WE	SEG Bit Write Enable	0: Write to SEG bit is disabled. 1: Write to SEG bit is enabled.	R/W
b7	WI	ELSEGR Register Write Disable	0: Write to ELSEGR register is enabled. 1: Write to ELSEGR register is disabled.	W

The MOV instruction must be used to write to this register.

#### SEG Bit (Software Event Generation)

When 1 is written to this bit while the WE bit is 1, a software event is generated.

This bit is read as 0. Even if 1 is written to this bit, the data will not be stored.

#### WE Bit (SEG Bit Write Enable)

The SEG bit can be written to only when the WE bit is 1.

[Setting condition]

If 1 is written to this bit while the WI bit is 0, this bit becomes 1.

[Clearing condition]

If 0 is written to this bit while the WI bit is 0, this bit becomes 0.

#### WI Bit (ELSEGR Register Write Disable)

The ELSEGR register can be written to only when the value to be written to the WI bit is 0.

This bit is read as 1.

### 17.3 Operation

#### 17.3.1 Relation between Interrupt Handling and Event Linking

The peripheral modules incorporated in the MCU are provided with the interrupt request status flags and the bits to enable/disable these interrupt requests. When an interrupt request is generated in a module, the corresponding interrupt request status flag is set. If the corresponding interrupt request is enabled then, the interrupt requested is issued to the CPU.

In contrast, the event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals to connect (link) them to different modules, allowing direct cooperation between the modules without CPU intervention. Event signals can be output regardless of the setting of the corresponding interrupt request enable bit. Figure 17.2 shows the relation between the interrupt handling and ELC.

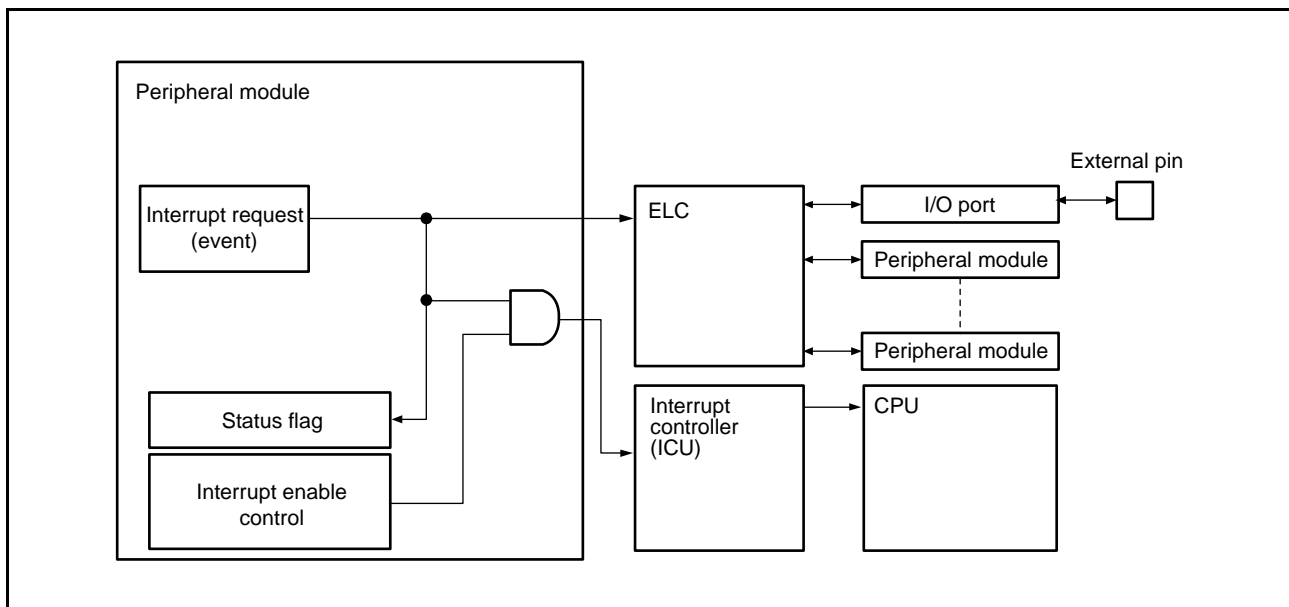


Figure 17.2 Relation between Interrupt Handling and ELC

### 17.3.2 Event Linkage

When an event has been set as a trigger in an event link setting register (ELSRn) and then occurs, the corresponding module is activated. Only one type of event can be connected with one module. When a module is to be activated by the ELC, the operation of the module must be set up in advance. Table 17.5 lists the operations of modules when an event is input.

**Table 17.5 Operations of Modules When Event is Input**

Module	Operations When Event is Input		
MTU CMT TMR	Each timer operates differently depending on the ELOPA to ELOPD registers as below. <ul style="list-style-type: none"> <li>• Starts counting when an event signal is input.</li> <li>• Restarts counting when an event signal is input.</li> <li>• Counts the input events (CMT, TMR).</li> <li>• Performs input-capture operation when an event is input (MTU).</li> </ul>		
POE	Places the MTU complementary PWM output pins and MTU0 output pins in the high-impedance state when an event signal is input.		
A/D converter	Starts A/D conversion when an event signal is input.		
D/A converter	Starts D/A conversion when an event signal is input		
I/O ports (output)	The value of PODR (port output data register) changes when an event signal is input. (The value output from the relevant external pin changes.)	Port group	<ul style="list-style-type: none"> <li>• Changes the PODR value to the specified value.</li> <li>• Transfers the PDBF1 value to the PODR register.</li> <li>• Rotates out the bit value.</li> </ul>
		Single port	Changes the PODR value to the specified value.
I/O ports (input)	When the signal value of the input pin changes	Port group	Generates an event.
		Single port	
	When an event is input	Port group	Transfers the signal value of the external pin to the PDBF1 register.
		Single port	Event connection is not possible.
Interrupt controller	Issues an event to the CPU and starts DTC data transfer when an event signal is input.		

### 17.3.3 Operation of Peripheral Timer Modules When Event is Input

The operations are performed depending on the ELOPA to ELOPD registers when an event is input.

#### (1) Count Start Operation

When an event is input, the timer starts counting, which sets the count start bit\*1 in each timer control register to 1. An event that is input while the count start bit is 1 is invalid.

#### (2) Count Restart Operation

When an event is input, the timer counter\*1 is initialized. Since the count start bit\*1 in each timer control register is retained, counting is restarted when an event is input while the count start bit is set to 1.

#### (3) Event Counter Operation

Event input is selected as the timer clock source and the timer counts events.

#### (4) Input Capture Operation

When an event is input, the timer performs input-capture operation.

Note 1. Refer to the register descriptions on starting the timer in the relevant peripheral timer module section.

### 17.3.4 Operation of A/D and D/A Converters When Event is Input

The A/D and D/A converter start A/D and D/A conversion, respectively, when the ADCSR.ADST bit and the DACR.DAOE0 bit\*1 are set to 1.

Note 1. Refer to the bit descriptions in the A/D converter and D/A converter sections.

### 17.3.5 I/O Port Operation upon Event Input and Event Generation

The I/O port operation to be performed upon event input and the operation to generate an event can be set.

#### (1) Single Ports and Port Groups

There are two event link modes: event link to single ports and event link to port groups. In the former mode, events can be connected to eight I/O ports. In the latter mode, events can be connected to port groups consisting of any two or more bits in the same eight I/O ports.

A single port can be set by specifying any bit in the I/O port\*1 to which an event can be connected using the PEL0 and PEL1 registers. A port group can be set by specifying any 2 or more bits in the I/O port\*1 to which an event can be connected using the PGC1 register. One input port group and one output port group can be set in the same I/O port.

If the I/O port bit is specified as both a single port and a member of a port group, both functions are enabled when the relevant port is input, whereas only the port group function is enabled when the relevant port is output.

Set the PDR register to select the direction of the I/O ports.

Note 1. Port B



(2) Event Generation by Single Input Ports

An single input port generates an event when the signal value of the external pin connected to the relevant port changes. The event generation condition is specified using the PEL0 and PEL1 registers. An example of operation is shown in Figure 17.3.

(3) Single Output Ports Operation upon Event Input

When an event is input to a single output port, the signal of the external pin connected to the relevant port changes according to the settings of the PEL0 and PEL1 registers. This changes the signal value of the external pin connected to the relevant port. An example of operation is shown in Figure 17.3.

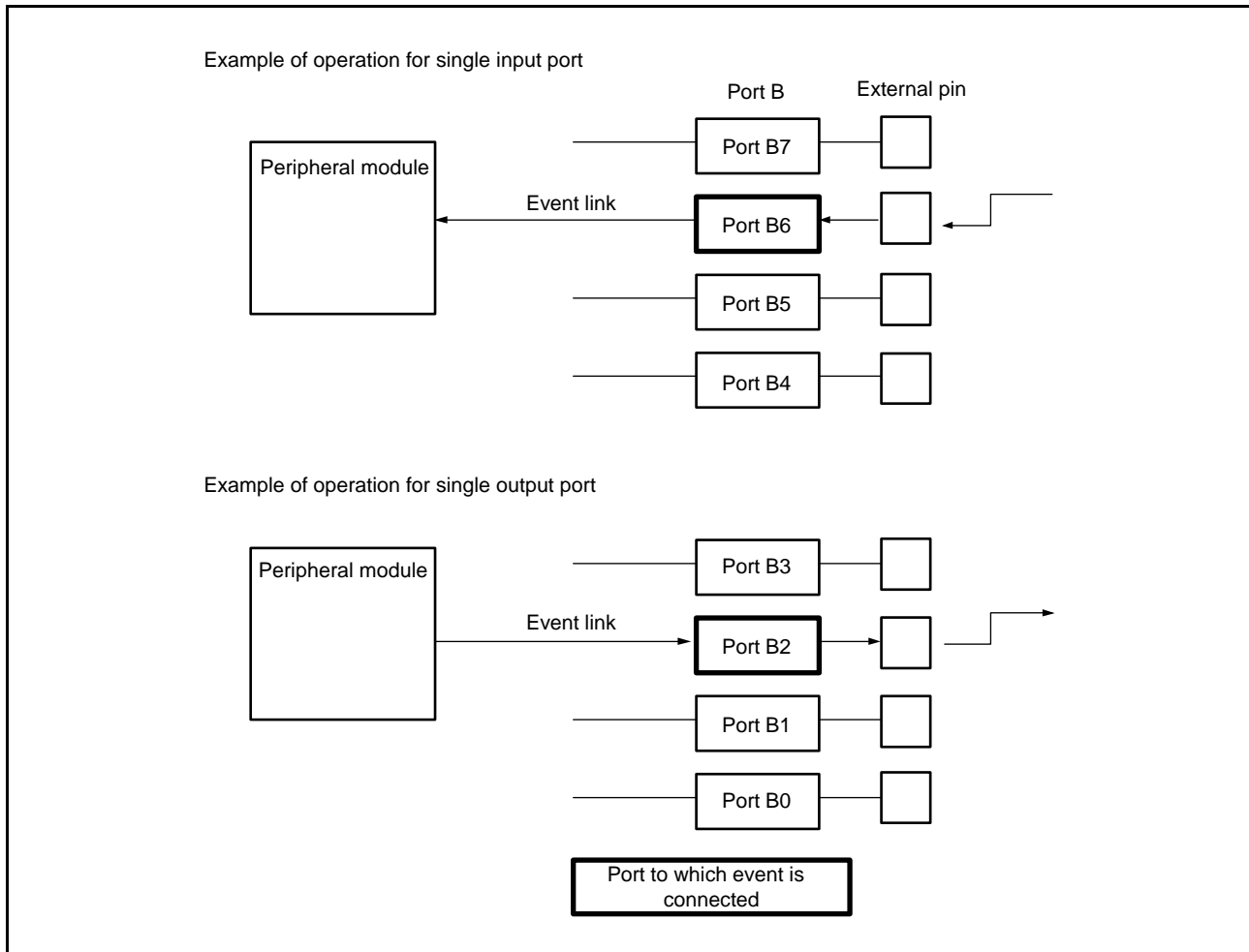


Figure 17.3 Event Linkage Related to Single Ports (Port B)

(4) Input Port Group Operation upon Event Input and Event Generation

An input port group generates an event when the signal value of any one of the external pins connected to the relevant port group changes. The event generation condition is specified using the PGC1 register. When an event is input to an input port group, the signal value of the external pin upon event input is transferred to the PDBF1 register. In this case, only the values of the bits specified as members of the input port group are transferred. An example of operation is shown in Figure 17.4.

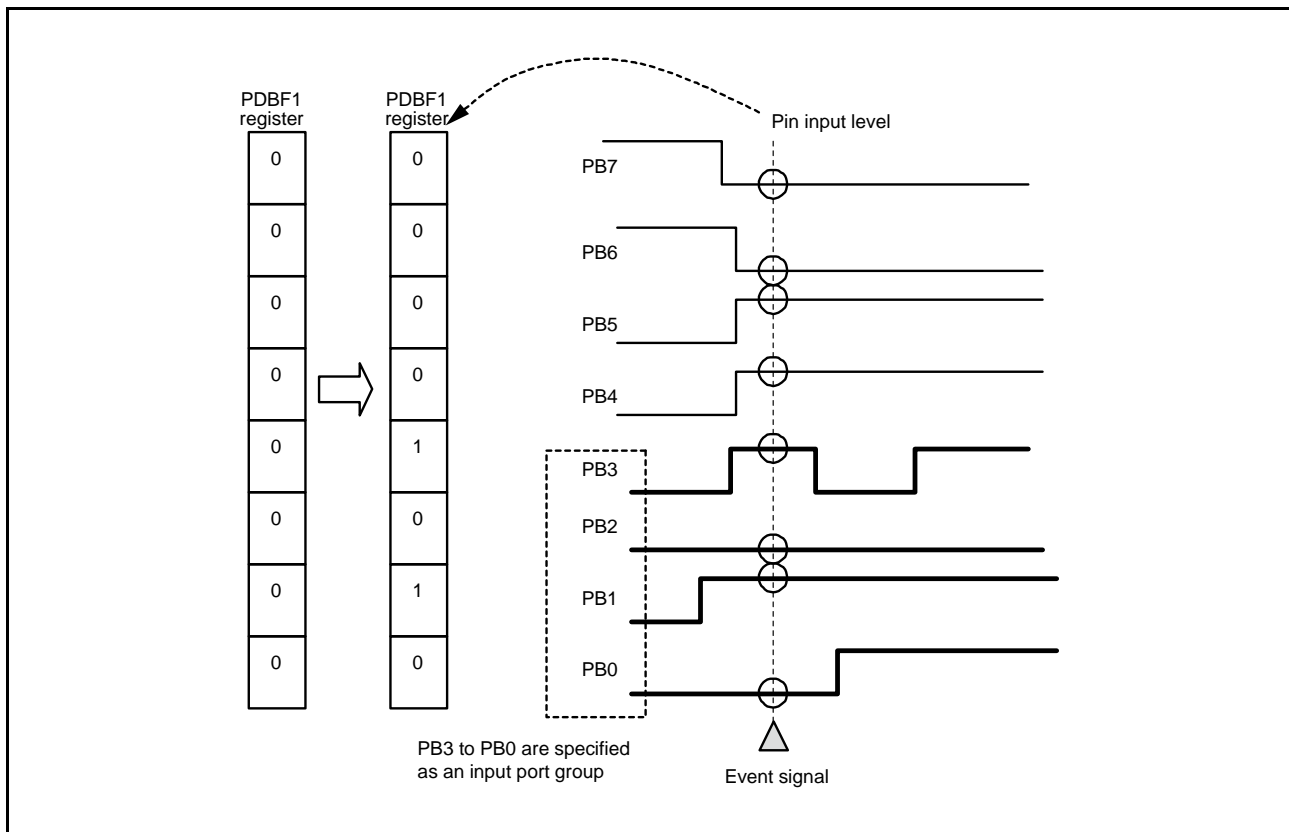


Figure 17.4 Event Linkage Related to Input Port Groups (Port B)

### (5) Output Port Group Operation upon Event Input

When an event is input to an output port group, the PODR values change to the values according to the PGC1 settings. An example of operation is shown in Figure 17.5.

### (6) Operation of Port Buffer Registers

#### (a) Input Port Groups

When an event is input to an input port group, the signal value of the external pin of the bit specified as a member of the input port group is transferred to the PDBF1 register. If another event is input to the input port group in this state, operations are performed depending on the PGC1.PGCOVE bit setting as described below.

- PGC1.PGCOVE = 0 (overwriting is disabled)  
If the PDBF1 value that has been transferred upon the latest event input has already been read by the CPU (or transferred by the DTC), the signal value of the external pin is transferred to the PDBF1 register. If not read, the signal value of the external pin is not transferred and the input event is invalid.
- PGC1.PGCOVE = 1 (overwriting is enabled)  
When another event is input to an input port group, the signal value of the external pin is transferred to the PDBF1 register.

#### (b) Output Port Groups

If an output port group is specified so that it should output the PDBF1 value, the PDBF1 value is transferred to the PODR register when an event is input to the output port group. In this case, only the values of the bits specified as members of the output port group are transferred.

If an output port group is specified so that it should rotate out the bit values in the group (PGC1.PGCO[2:0] bits = 1xxb), the PDBF1 data is transferred to the PODR register, and then the PODR value is rotated bit by bit from MSB to LSB. The

initial value to be output to the port group should be provided in the PDBF1 register. Examples of operation are shown in Figure 17.5 and Figure 17.6.

(7) Restrictions on Writing to PODR and PDBF Registers

When the ELCR.ELCON bit is set to 1, write access to the following registers is disabled.

- If bits are specified as members of the input port group and the event linkage is set for the port group, write access to the relevant bits in the PDBF1 register is disabled. However, when the DOC is selected for event input, write access is enabled.
- If port bits are specified as members of the output port group, write access to the relevant bits in the PODR register is disabled.
- If a port bit is specified as a single output port and the event linkage is set (by the ELSRn register) for the port, write access to the relevant bit in the PODR register is disabled. However, when the DOC is selected for event input, write access is enabled.

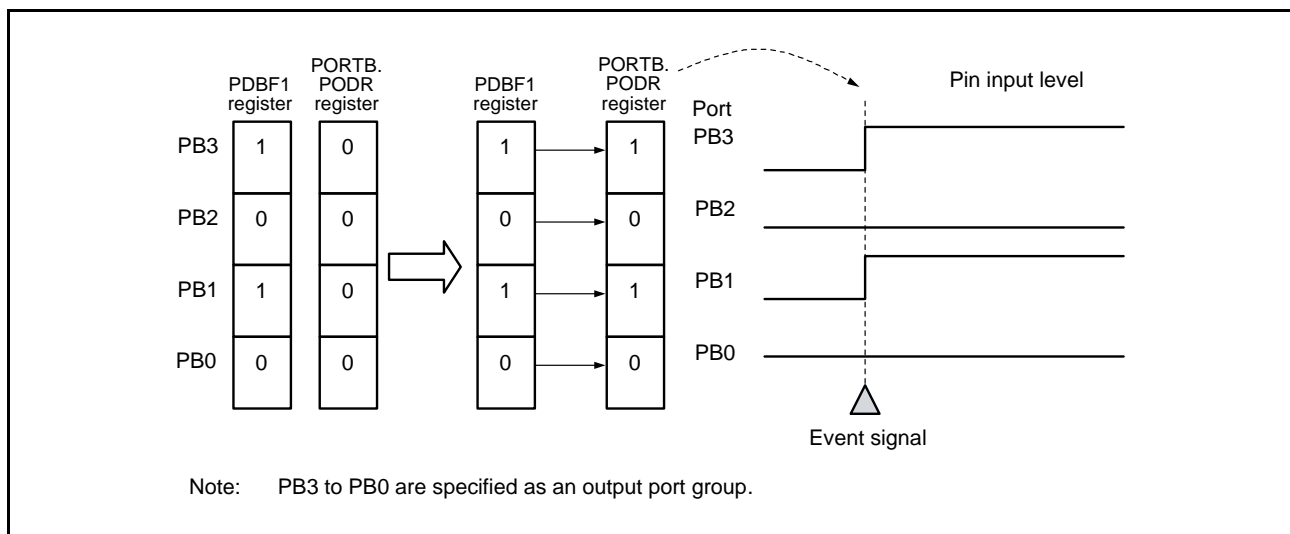


Figure 17.5 Event Linkage Related to Output Port Groups (Port B)

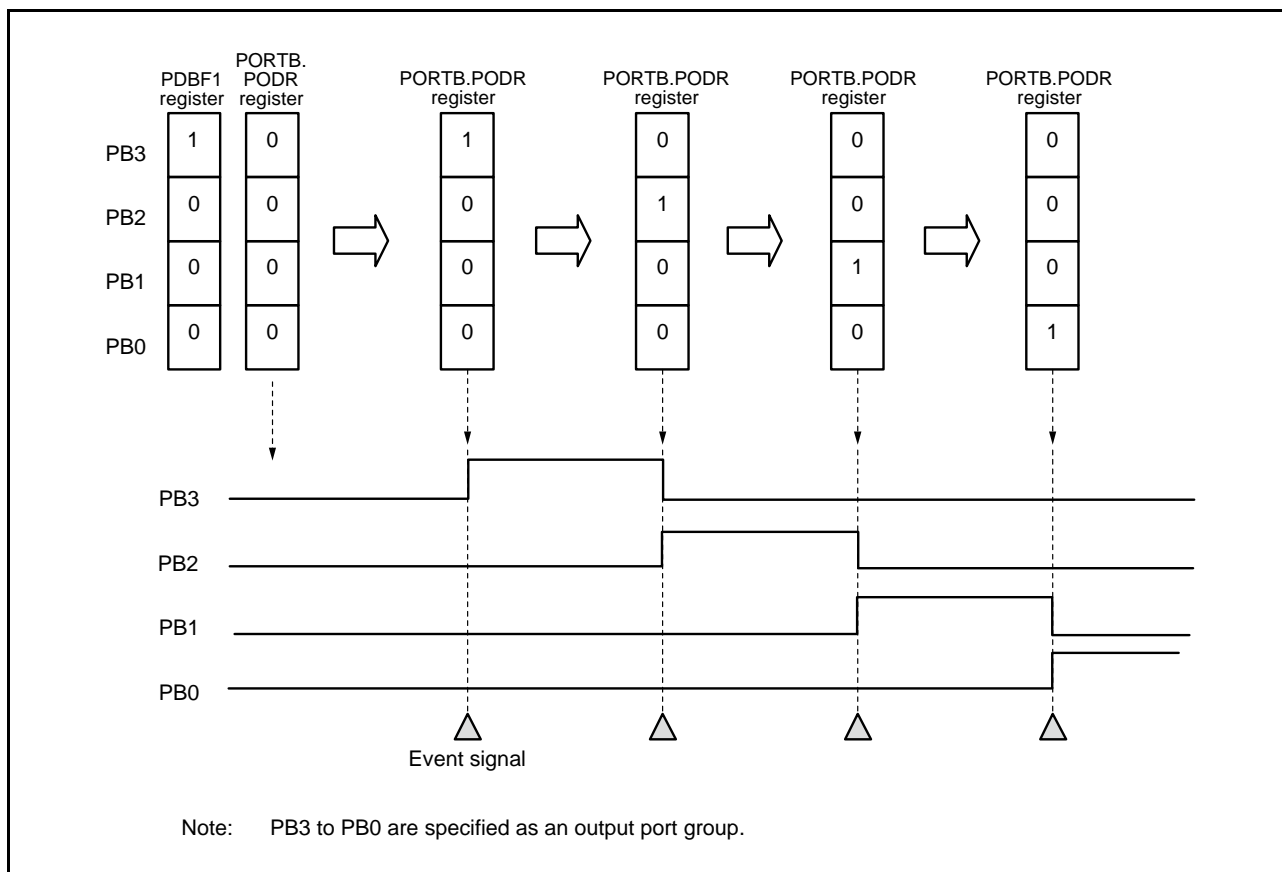


Figure 17.6 Bit-Rotating Operation of Output Port Groups (Port B)

### 17.3.6 Example of Procedure for Linking Events

The following describes the procedure for linking events.

1. Set the operation of the module to which an event is to be linked.
2. If events are linked to ports, set the registers corresponding to the ports as below.  
PODR: Set the initial values of the output ports.  
PDR: Set the I/O direction of the ports.  
PGR1: If ports are used as a port group, set the ports (in bit units) to be grouped.  
PGC1: Set the operation of the port group.  
PEL0, PEL1: If ports are used as single ports, set the ports, the operation of the ports when an event is input, and the condition when an event is generated.
3. To the ELSRn register corresponding to the module to which an event signal is to be linked, set the number of the event signal.
4. If events are to be linked to timer modules, set the ELOPA to ELOPD registers corresponding to the timers as required.
5. Set the ELCR.ELCON bit to 1, which enables linkage of all the events.
6. Set the operation of the module from which an event is output, and activate the module. This allows the event output from the module to start the module to which an event is linked as preset.
7. To stop event linkage of independent modules, set 00000000b to the ELSRn.ELS[7:0] bits corresponding to the modules. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

## 17.4 Usage Notes

### 17.4.1 Setting ELSRn Register

#### (1) Setting ELSR18 Register

Specify an event number from among 01100011b (63h) to 01101010b (6Ah); other settings are prohibited.

#### (2) Setting ELSR24 and ELSR25 Registers

Setting the DOC data operation condition met signal (01101010b (6Ah)) is prohibited.

### 17.4.2 Setting Bit-Rotating Operation of Output Port Groups

When the values of the PDBF1 register are changed in the bit-rotating operation mode of the output port group, set the ELSRn register again. When events are used during bit-rotating operation, generate an event after an interval of one PCLKB cycle. If not, the normal operation cannot be provided.

### 17.4.3 Linking DTC Transfer End Signals as Events

When linking the DTC transfer end signals as events, do not set the same peripheral module as the DTC transfer destination and event link destination. If set, the peripheral module might be started before DTC transfer to the peripheral module is completed.

### 17.4.4 Setting Clocks

To link events, it is necessary for the ELC and the related modules to be enabled. The modules cannot operate if the related modules are in the module stop state or in the specific low power consumption mode in which the module is stopped (software standby mode).

### 17.4.5 Module Stop Function Setting

ELC operation can be disabled or enabled using module stop control register B (MSTPCRB). The initial setting allows the ELC to be stopped. Register access is enabled by releasing the module stop state. For details, refer to [section 11, Low Power Consumption](#).

## 18. I/O Ports

### 18.1 Overview

The I/O ports can function as general I/O ports, I/O pins of a peripheral module, or as input pins for an interrupt. Each pin can also be configured as an I/O pin of a peripheral module or as an input pin for an interrupt. Immediately after a reset, all pins function as input pins, 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 registers for the on-chip peripheral modules.

Each port has the port direction register (PDR) that selects input or output direction, the port output data register (PODR) that holds data for output, the port input data register (PIDR) that indicates the pin states, the open drain control register  $y$  (ODR $_y$ ,  $y = 0, 1$ ) that selects the output type of each pin, the pull-up control register (PCR) that controls on/off of the input pull-up MOS, and the port mode register (PMR) that specifies the pin function of each port.

For details on the PMR register, refer to **section 19, Multi-Function Pin Controller (MPC)**.

Products with 64-pin packages include port switching register A (PSRA). However, when ports PC0 and PC1 are selected in port switching register A (PSRA), input to ports PB6 and PB7 and the output port event function of the ELC cannot be used.

The configuration of the I/O ports differs depending on the package. **Table 18.1** lists the specifications of I/O ports, and **Table 18.2** lists the port functions.

**Table 18.1 I/O Port Specifications**

Port Symbol	Package		Package	
	100 Pins	Number of Pins	64 Pins	Number of Pins
PORT0	P02, P04, P07	3	—	—
PORT1	P10 to P17	8	P14 to P17	4
PORT2	P20 to P27	8	P26, P27	2
PORT3	P30 to P32, P35	4	P30 to P32, P35	4
PORT4	P40 to P44, P46	6	P40 to P42	3
PORT5	P50 to P56	7	P54, P55	2
PORT9	P90 to P92	3	—	—
PORTA	PA0 to PA7	8	PA0, PA1, PA3, PA4, PA6	5
PORTB	PB0 to PB7	8	PB0, PB1, PB3, PB5 to PB7	6
PORTC	PC0 to PC7	8	PC2 to PC7	6
PORTD	PD0 to PD4	5	PD0 to PD2	3
PORTE	PE0 to PE7	8	PE0 to PE7	8
PORTF	PF6, PF7	2	—	—
PORTH	PH7	1	PH7	1
PORTJ	PJ0, PJ2, PJ3, PJ6, PJ7	5	PJ0, PJ2, PJ6, PJ7	4
Total number of pins		84	Total number of pins	48

Note: PB6 and PC0, and PB7 and PC1 can be switched by port switching register A (PSRA).

**Table 18.2 Port Functions**

Port	Port Register	Input Pull-up	Open Drain Output	5-V Tolerant
PORT0	P02, P04, P07	○	○	—
PORT1	P10 to P15	○	○	—
	P16, P17	○	○	○
PORT2	P20 to P27	○	○	—
PORT3	P30 to P32	○	○	—
	P35	—	—	—
PORT4	P40 to P44, P46	—	—	—
PORT5	P50 to P53, P56	○	○	—
	P54, P55	○	—	—
PORT9	P90 to P92	—	—	—
PORTA	PA0 to PA5, PA7	○	○	—
	PA6	○	○	○
PORTB	PB1 to PB7	○	○	—
	PB0	○	○	○
PORTC	PC2 to PC7	○	○	—
PORTD	PD0 to PD4	○	—	—
PORTE	PE0 to PE7	○	○	—
PORTF	PF6, PF7	○	—	—
PORTH	PH7	—	—	—
PORTJ	PJ0, PJ2	○	—	—
	PJ3	○	○	—
	PJ6, PJ7	—	—	—

Specifying input pull-up, open-drain output, or 5-V tolerance is available for other signals on pins that also function as general I/O pins.



18.2 I/O Port Configuration

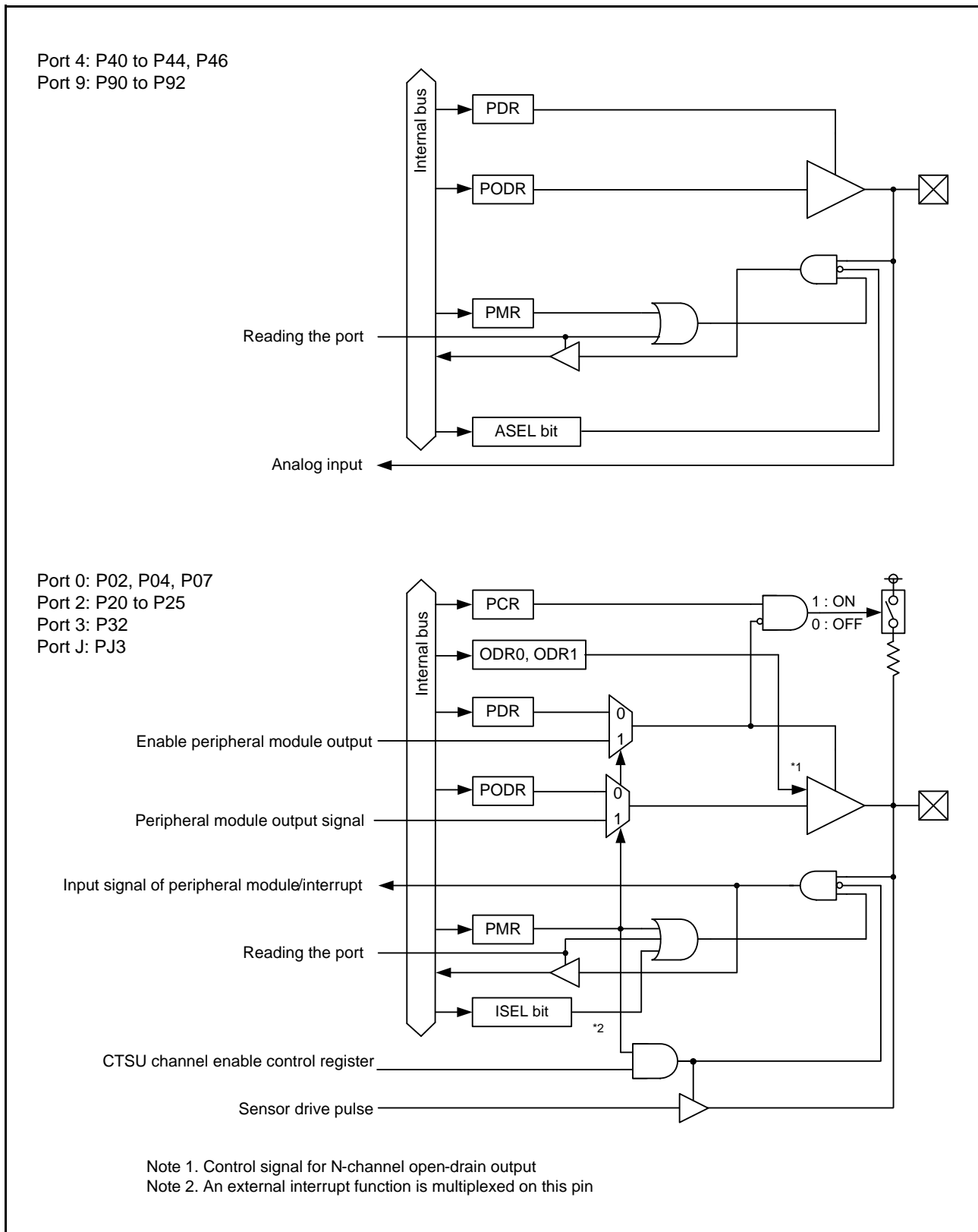


Figure 18.1 I/O Port Configuration (1)

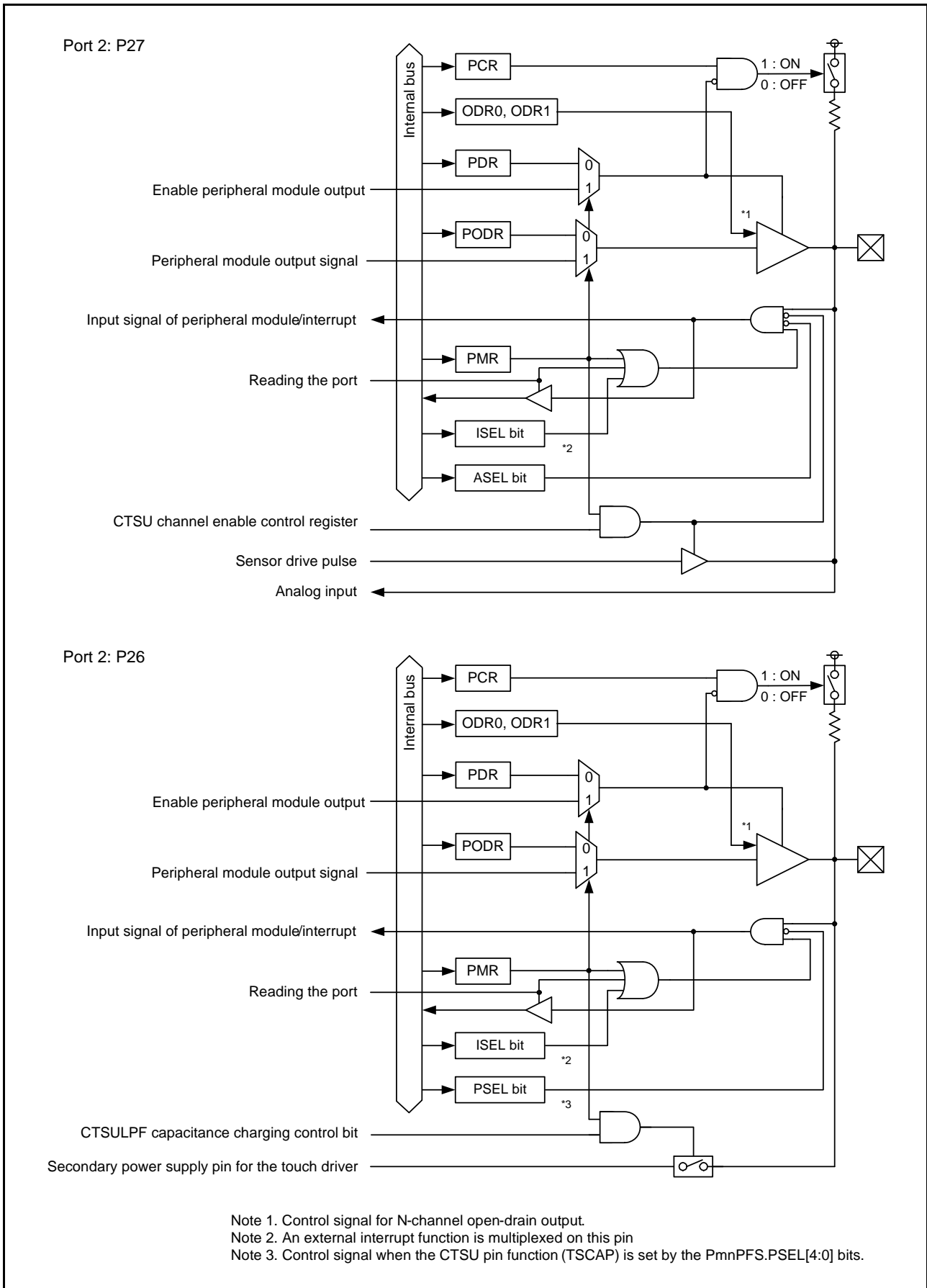


Figure 18.2 I/O Port Configuration (2)

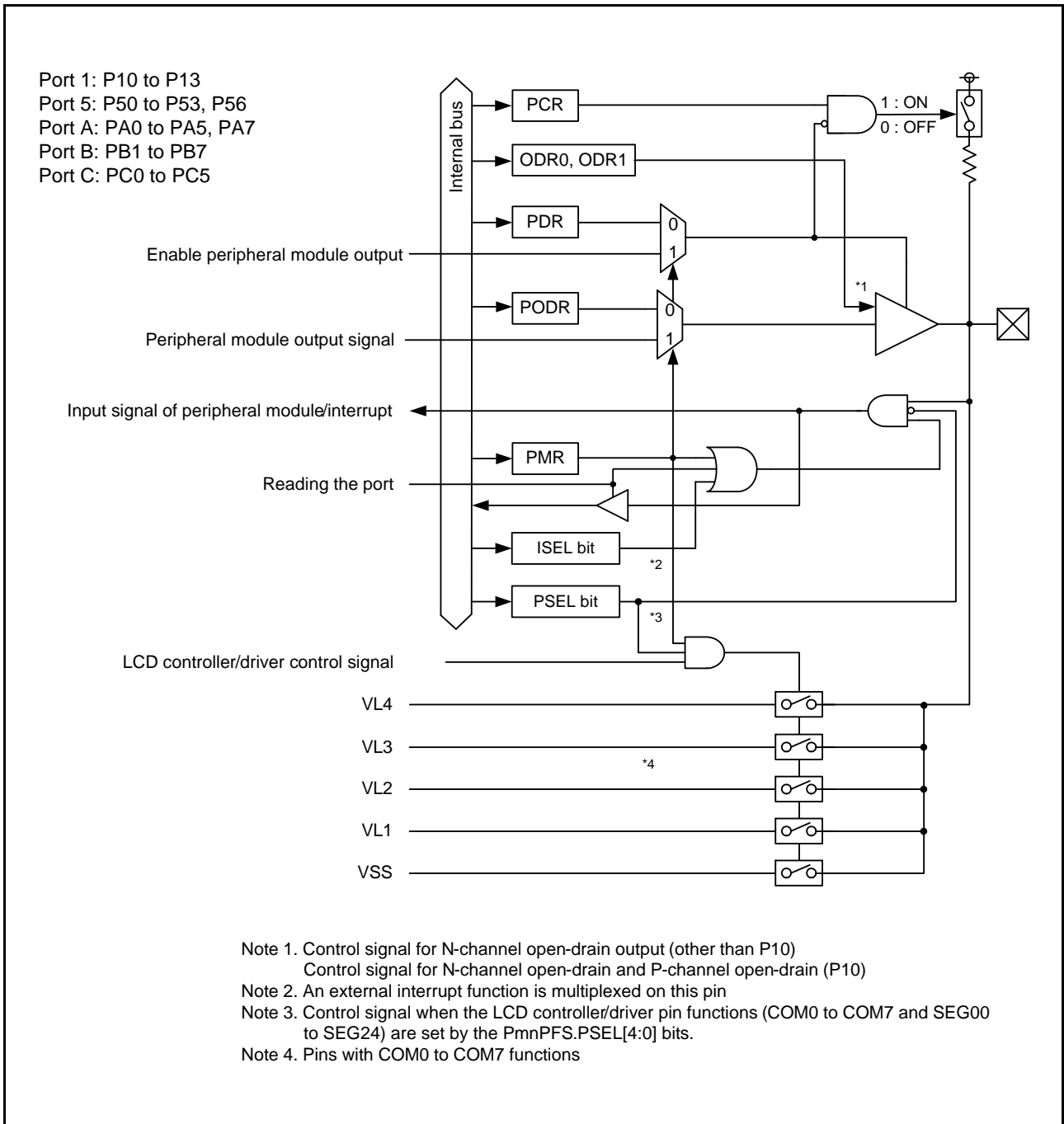


Figure 18.3 I/O Port Configuration (3)

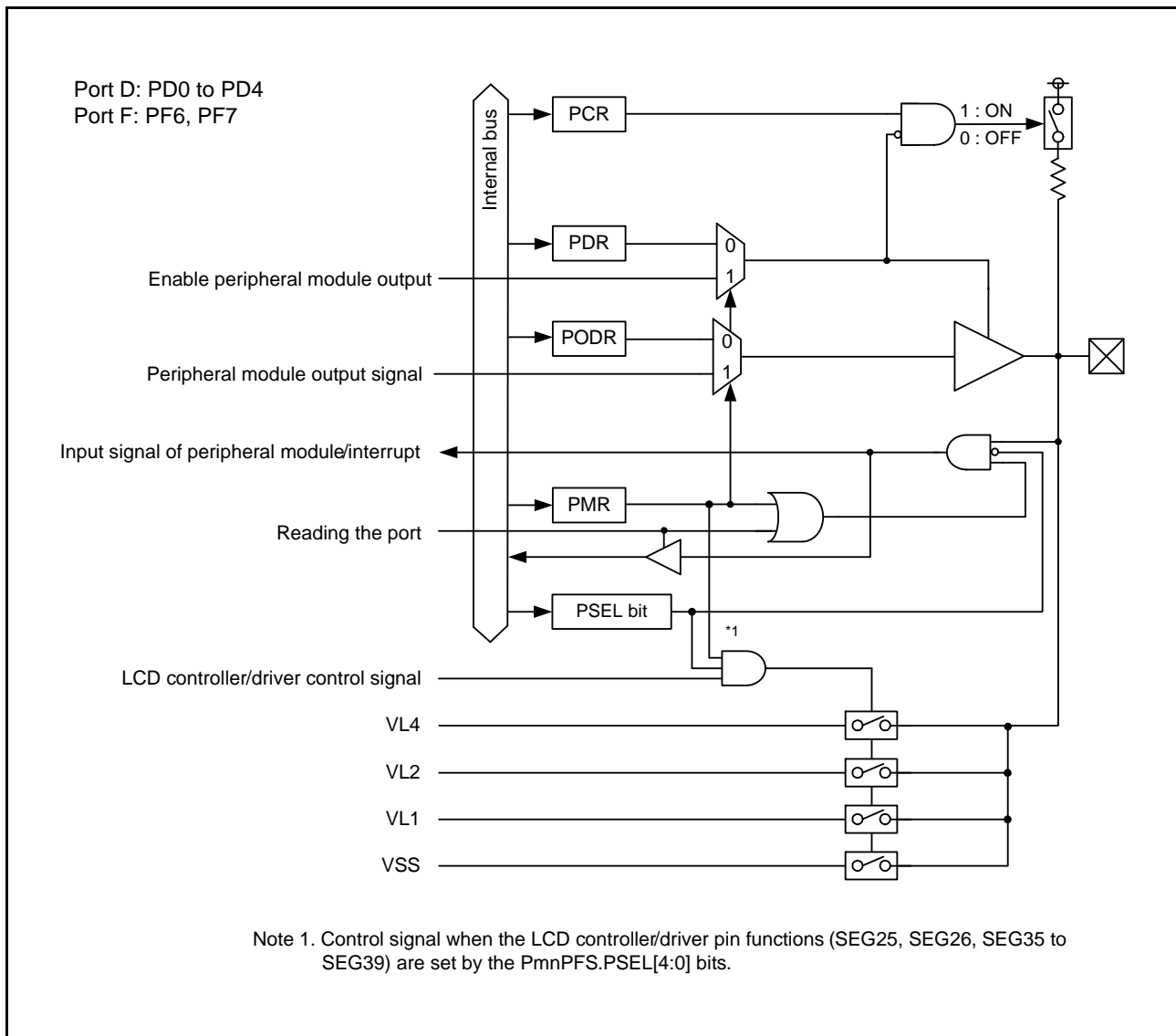


Figure 18.4 I/O Port Configuration (4)

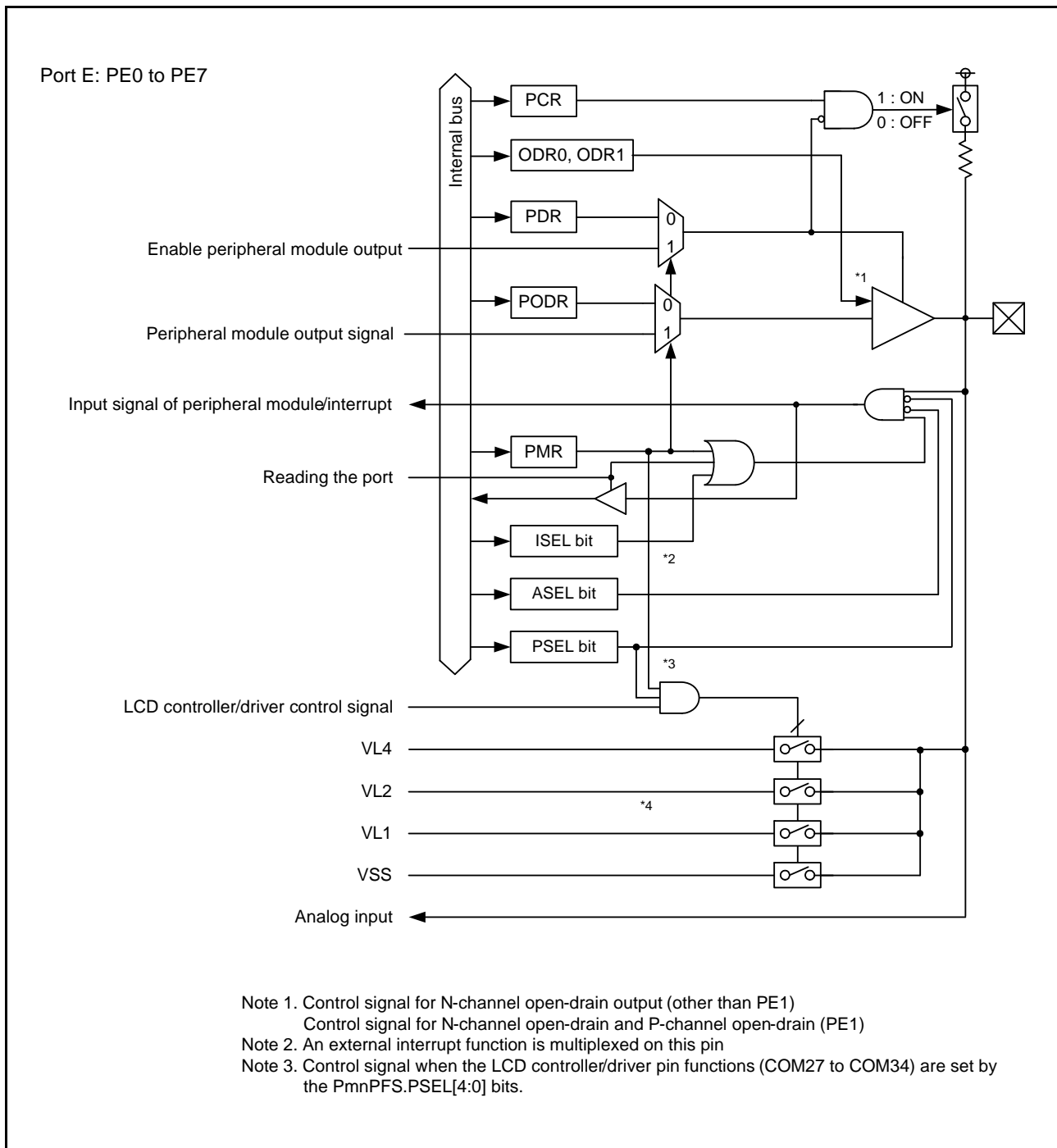


Figure 18.5 I/O Port Configuration (5)

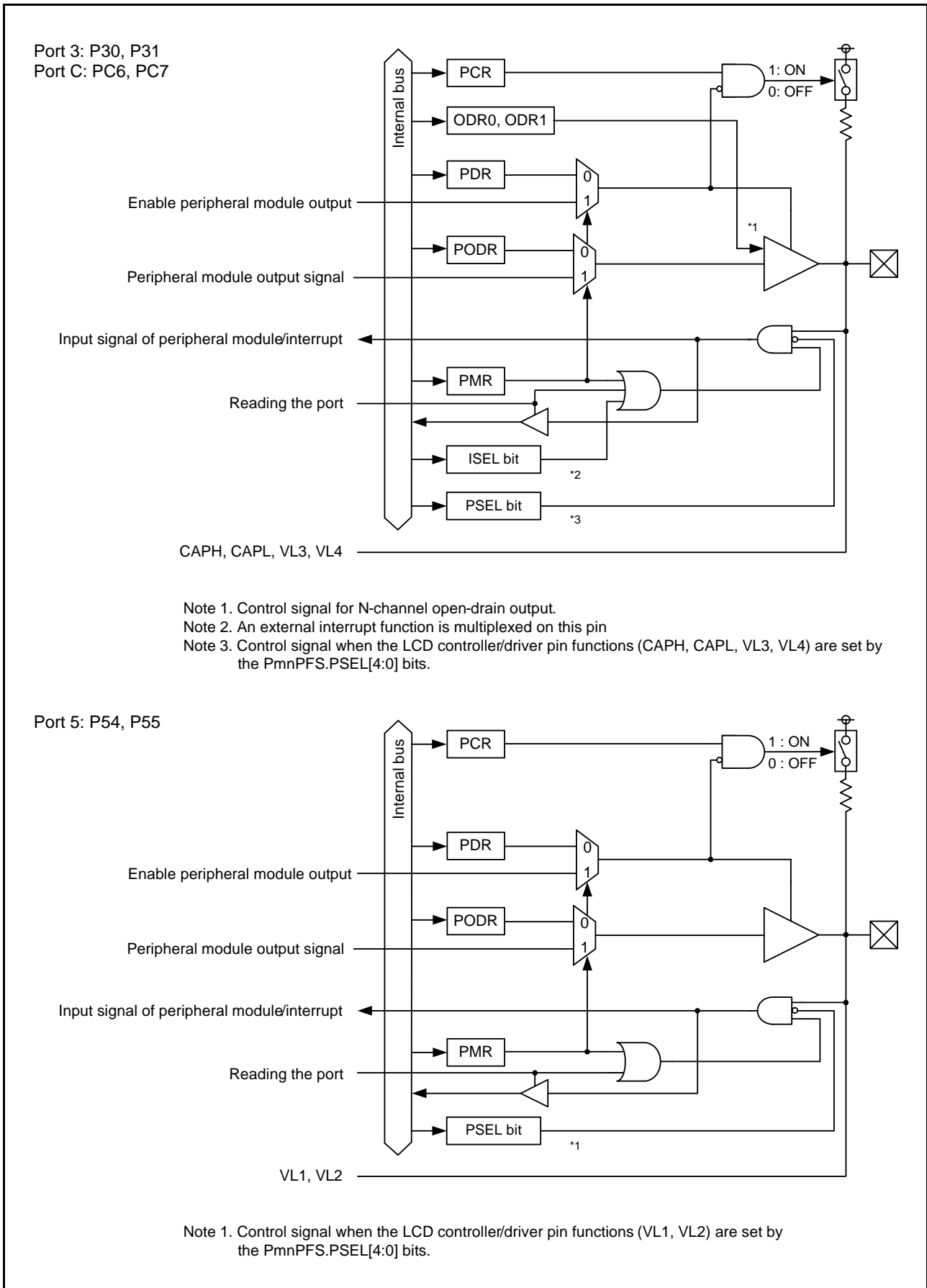


Figure 18.6 I/O Port Configuration (6)

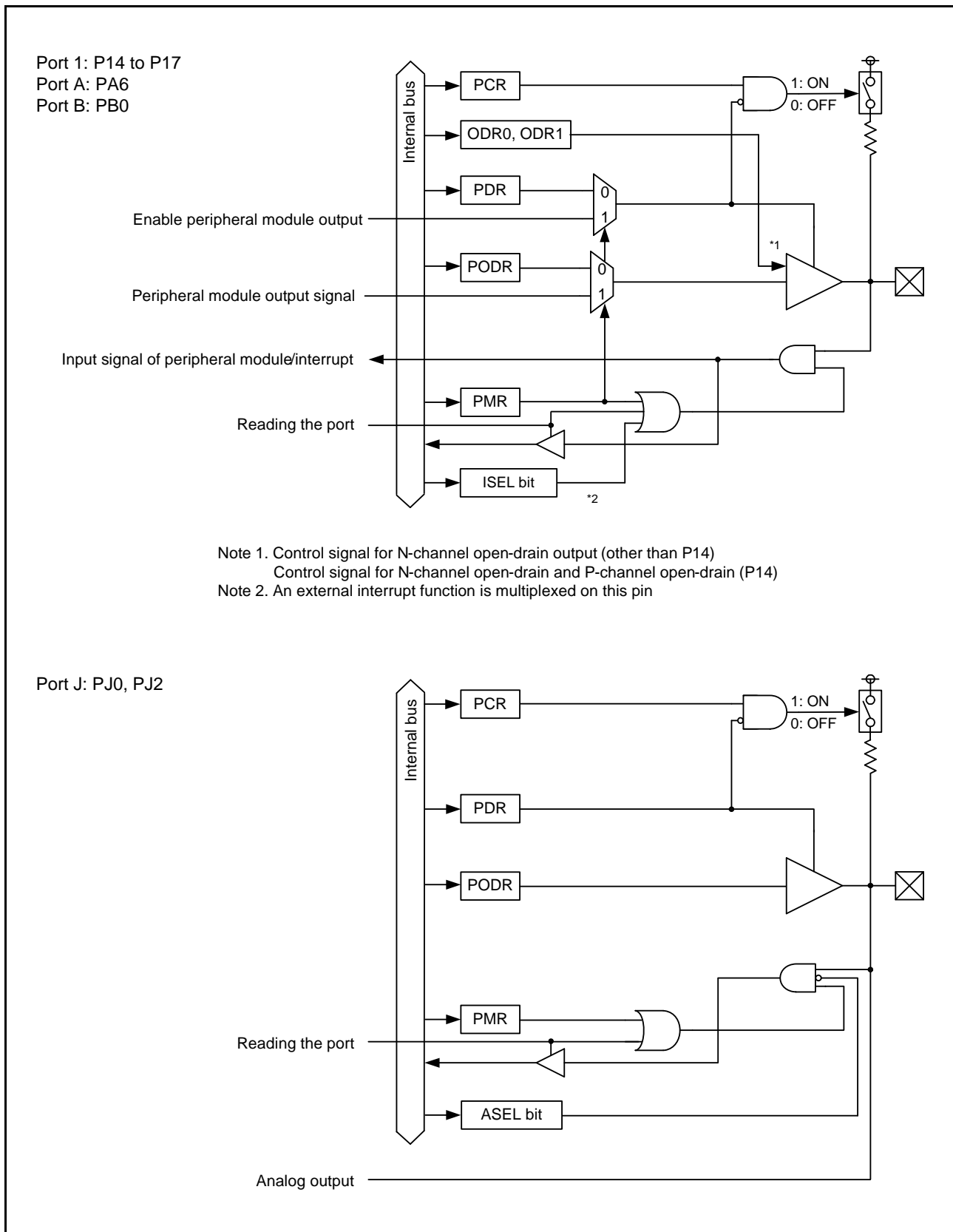


Figure 18.7 I/O Port Configuration (7)

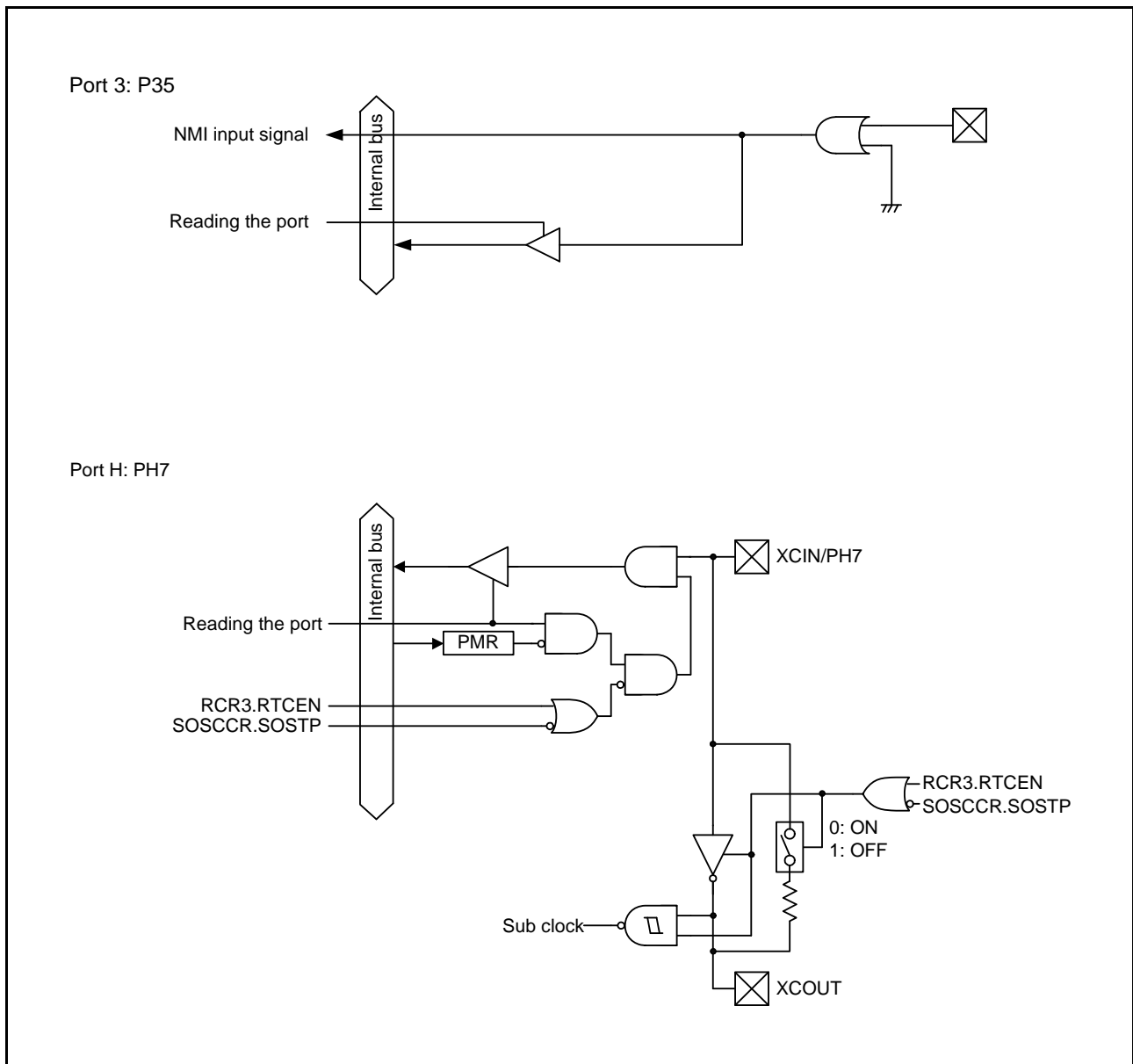


Figure 18.8 I/O Port Configuration (8)



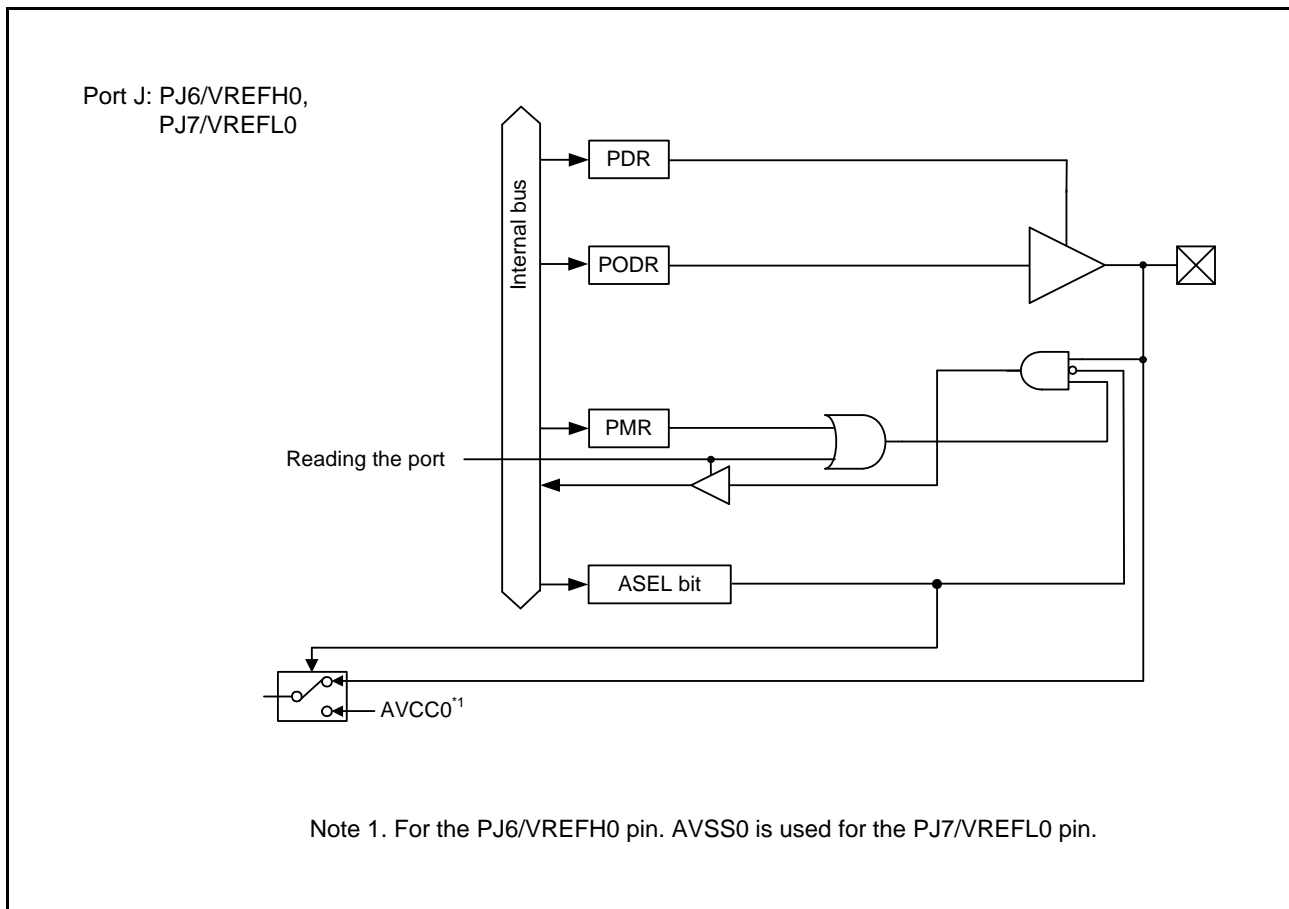


Figure 18.9 I/O Port Configuration (9)

## 18.3 Register Descriptions

### 18.3.1 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, 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, PORTJ.PDR 0008 C012h

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 Direction Control	0: Input (Functions as an input pin.)	R/W
b1	B1	Pm1 Direction Control	1: Output (Functions as an output pin.)	R/W
b2	B2	Pm2 Direction Control		R/W
b3	B3	Pm3 Direction Control		R/W
b4	B4	Pm4 Direction Control		R/W
b5	B5	Pm5 Direction Control		R/W
b6	B6	Pm6 Direction Control		R/W
b7	B7	Pm7 Direction Control		R/W

m = 0 to 5, 9, A to F, J

PDR is used to select the input or output direction for individual pins of the corresponding port m when the pins are configured as the general I/O pins.

PORTm.PDR is a direction register of port m. Each bit in this register corresponds to each pin of port m and I/O direction can be specified in 1-bit units. The bits corresponding to pins that are not listed in Table 18.1 and the PORT3.PDR.B5 bit of the input-only P35 pin are reserved. A reserved bit should be set to 0 or 1 according to Table 18.5. When setting a value to a reserved bit, access in byte units.

### 18.3.2 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, 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, PORTJ.PODR 0008 C032h

b7	b6	b5	b4	b3	b2	b1	b0
B7	B6	B5	B4	B3	B2	B1	B0

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

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Data Store	Holds output data.	R/W
b1	B1	Pm1 Output Data Store		R/W
b2	B2	Pm2 Output Data Store		R/W
b3	B3	Pm3 Output Data Store		R/W
b4	B4	Pm4 Output Data Store		R/W
b5	B5	Pm5 Output Data Store		R/W
b6	B6	Pm6 Output Data Store		R/W
b7	B7	Pm7 Output Data Store		R/W

m = 0 to 5, 9, A to F, J

PODR holds the data to be output from the pins used for general output ports.

The bits corresponding to pins that are not listed in Table 18.1 and the PORT3.PODR.B5 bit of the input-only P35 pin are reserved. A reserved bit is read as 0. The write value should be 0. When setting a value to a reserved bit, access in byte units.

### 18.3.3 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, 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, PORTH.PIDR 0008 C051h, PORTJ.PIDR 0008 C052h

b7	b6	b5	b4	b3	b2	b1	b0
B7	B6	B5	B4	B3	B2	B1	B0

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0	Indicates individual pin states of the corresponding port.	R/W
b1	B1	Pm1		R/W
b2	B2	Pm2		R/W
b3	B3	Pm3		R/W
b4	B4	Pm4		R/W
b5	B5	Pm5		R/W
b6	B6	Pm6		R/W
b7	B7	Pm7		R/W

m = 0 to 5, 9, A to F, H, J

PIDR indicates individual pin states of port m.

The pin states of port m can be read with the PORTm.PIDR, regardless of the values of PORTm.PDR and PORTm.PMR.

The NMI pin state is reflected in the P35 bit.

The bits corresponding to pins that are not listed in Table 18.1 are reserved. The read value of a reserved bit is undefined. Writing has no effect.

#### Notes when using PH7 as a general input port and PJ6 and PJ7 as general I/O ports

When using PH7 as a general input port, follow the procedure below to make settings.

1. Set the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) to 0. For details on this register, refer to section 24.2.19, RTC Control Register 3 (RCR3).
2. Set the sub-clock oscillator stop bit in the sub-clock oscillator control register (SOSCCR.SOSTP) to 1. For details on the functions and rewriting of this register, refer to section 9.2.8, Sub-Clock Oscillator Control Register (SOSCCR).

When using PJ6 and PJ7 as a general input port, follow the procedure below to make settings.

1. Set the PJ6PFS.ASEL bit to 0. (When using the PJ6 port)  
Set the PJ7PFS.ASEL bit to 0. (When using the PJ7 port)
2. Set the PORTJ.PMR.B6 to 0. (When using the PJ6 port)  
Set the PORTJ.PMR.B7 to 0. (When using the PJ7 port)

### 18.3.4 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, 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, PORTH.PMR 0008 C071h, PORTJ.PMR 0008 C072h

b7	b6	b5	b4	b3	b2	b1	b0
B7	B6	B5	B4	B3	B2	B1	B0

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

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Pin Mode Control	0: Use pin as general I/O port.	R/W
b1	B1	Pm1 Pin Mode Control	1: Use pin as I/O port for peripheral functions.	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 5, 9, A to F, H, J

Each bit of PORTm.PMR corresponds to each pin of port m; pin function can be specified in 1-bit units.

The bits corresponding to pins that are not listed in Table 18.1 and the PORT3.PMR.B5 bit of the input-only P35 pin are reserved. A reserved bit is read as 0. The write value should be 0. When setting a value to a reserved bit, access in byte units.

### 18.3.5 Open Drain Control Register 0 (ODR0)

Address(es): PORT0.ODR0 0008 C080h, PORT1.ODR0 0008 C082h, PORT2.ODR0 0008 C084H, PORT3.ODR0 0008 C086h, PORT5.ODR0 0008 C08Ah, PORTA.ODR0 0008 C094h, PORTB.ODR0 0008 C096h, PORTC.ODR0 0008 C098h, PORTE.ODR0 0008 C09Ch, PORTJ.ODR0 0008 C0A4h

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

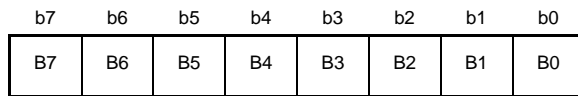
Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Type Select	<ul style="list-style-type: none"> <li>P20, P30, P50, PA0, PB0, PC0, PE0</li> </ul>	R/W
b1	B1		b0 0: CMOS output 1: N-channel open-drain output b1 These bits are read as 0. The write value should be 0. <ul style="list-style-type: none"> <li>P10</li> </ul> b1 b0 0 0: CMOS output 0 1: N-channel open-drain output 1 0: P-channel open-drain output 1 1: Setting prohibited.	R/W
b2	B2	Pm1 Output Type Select	<ul style="list-style-type: none"> <li>P11, P21, P31, P51, PA1, PB1, PC1</li> </ul>	R/W
b3	B3		b2 0: CMOS output 1: N-channel open-drain output b3 These bits are read as 0. The write value should be 0. <ul style="list-style-type: none"> <li>PE1</li> </ul> b3 b2 0 0: CMOS output 0 1: N-channel open-drain output 1 0: P-channel open-drain output 1 1: Setting prohibited.	R/W
b4	B4	Pm2 Output Type Select	0: CMOS output 1: N-channel open-drain output	R/W
b5	B5	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	B6	Pm3 Output Type Select	0: CMOS output 1: N-channel open-drain output	R/W
b7	B7	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 0 to 3, 5, A to C, E, J

The bits corresponding to pins that are not listed in Table 18.1 are reserved. A reserved bit is read as 0. The write value should be 0. When setting a value to a reserved bit, access in byte units.

### 18.3.6 Open Drain Control Register 1 (ODR1)

Address(es): PORT0.ODR1 0008 C081h, PORT1.ODR1 0008 C083h, PORT2.ODR1 0008 C085h, PORT5.ODR1 0008 C08Bh, PORTA.ODR1 0008 C095h, PORTB.ODR1 0008 C097h, PORTC.ODR1 0008 C099h, PORTE.ODR1 0008 C09Dh



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

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm4 Output Type Select	<ul style="list-style-type: none"> <li>P04, P24, PA4, PB4, PC4, PE4</li> </ul>	R/W
b1	B1		b0 0: CMOS output 1: N-channel open-drain output b1 These bits are read as 0. The write value should be 0. <ul style="list-style-type: none"> <li>P14</li> </ul> b1 b0 0 0: CMOS output 0 1: N-channel open-drain output 1 0: P-channel open-drain output 1 1: Setting prohibited.	R/W
b2	B2	Pm5 Output Type Select	0: CMOS output 1: N-channel open-drain output	R/W
b3	B3	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	B4	Pm6 Output Type Select	0: CMOS output 1: N-channel open-drain output	R/W
b5	B5	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	B6	Pm7 Output Type Select	0: CMOS output 1: N-channel open-drain output	R/W
b7	B7	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 0 to 2, 5, A to C, E

The bits corresponding to pins that are not listed in Table 18.1 and the PORT3.ODR1.B2 bit of the input-only P35 pin are reserved. A reserved bit is read as 0. The write value should be 0. When setting a value to a reserved bit, access in byte units.

### 18.3.7 Pull-Up Control Register (PCR)

Address(es): PORT0.PCR 0008 C0C0h, PORT1.PCR 0008 C0C1h, PORT2.PCR 0008 C0C2h, PORT3.PCR 0008 C0C3h, PORT5.PCR 0008 C0C5h, PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTC.PCR 0008 C0CCh, PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh, PORTF.PCR 0008 C0CFh, PORTJ.PCR 0008 C0D2h

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 3, 5, A to E, J

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 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 bits corresponding to pins that are not listed in Table 18.1 and the PORT3.PCR.B5 bit are reserved. A reserved bit is read as 0. The write value should be 0. When setting a value to a reserved bit, access in byte units.



### 18.3.8 Port Switching Register A (PSRA)

Address(es): PORT.PSRA 0008 C121h

b7	b6	b5	b4	b3	b2	b1	b0
PSEL7	PSEL6	—	—	—	—	—	—

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

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	PSEL6	PB6/PC0 Switching	0: PB6 general I/O port function is selected 1: PC0 general I/O port function is selected	R/W
b7	PSEL7	PB7/PC1 Switching	0: PB7 general I/O port function is selected 1: PC1 general I/O port function is selected	R/W

The PSRA register is for 64-pin packages. The PSRA register is used to select either the general I/O functions of PB6 and PB7 or those of PC0 and PC1. When 1 is written to the PSEL6 and PSEL7 bits, port C can be used as an 8-bit port. As for the I/O functions of the peripheral functions, functions multiplexed with PB6 and PB7 are enabled. To enable the peripheral functions, write 1 to the corresponding pin mode control bit of the PORTB.PMR register. Rewriting to this register must be performed when the PMR, PDR, and PCR registers for the corresponding pins are 0.

## 18.4 Initialization of the Port Direction Register (PDR)

Initialize reserved bits in the PDR register according to Table 18.3 and Table 18.4.

- The blank columns in Table 18.3 and Table 18.4 indicate the bits corresponding to the pins listed in Table 18.1, I/O Port Specifications.

The corresponding bits should be set to 1 (output) or 0 (input) depending on the user system.

However, the PORT3.PDR.B5 bit of the input-only P35 pin is reserved.

This bit should be set to 0 (input).

- The columns other than the blank columns in Table 18.3 and Table 18.4 indicate reserved bits. A reserved bit should be set to 0 (input) or 1 (output) according to Table 18.3 and Table 18.4. When setting a value to a reserved bit, access in byte units.

**Table 18.3 PDR Register Settings in 100-Pin Packages**

Port Symbol	PDR Register							
	b7	b6	b5	b4	b3	b2	b1	b0
PORT0		1	1		1		1	1
PORT1								
PORT2								
PORT3	1	1	0	1	1			
PORT4	1		1					
PORT5	1							
PORT9	1	1	1	1	1			
PORTA								
PORTB								
PORTC								
PORTD	1	1	1					
PORTE								
PORTF			1	1	1	1	1	1
PORTJ			1	1			1	

**Table 18.4 PDR Register Settings in 64-Pin Packages**

Port Symbol	PDR Register							
	b7	b6	b5	b4	b3	b2	b1	b0
PORT0	1	1	1	1	1	1	1	1
PORT1					1	1	1	1
PORT2			1	1	1	1	1	1
PORT3	1	1	0	1	1			
PORT4	1	1	1	1	1			
PORT5	1	1			1	1	1	1
PORT9	1	1	1	1	1	1	1	1
PORTA	1		1			1		
PORTB				1		1		
PORTC								
PORTD	1	1	1	1	1			
PORTE								
PORTF	1	1	1	1	1	1	1	1
PORTJ			1	1	1		1	

## 18.5 Handling of Unused Pins

The handling of unused pins is listed in Table 18.5.

**Table 18.5 Handling of Unused Pins**

Pin Name	Description
MD	(Always used as mode pins)
RES#	Connect this pin to VCC via a pull-up resistor.
P35/NMI	Connect this pin to VCC via a pull-up resistor.
USB0_DM, USB0_DP	Leave this pin open.
EXTAL	Connect this pin to VSS via a pull-down resistor.
XTAL	Leave this pin open.
PH7/XCIN	When the sub-clock is not used, set the RCR3.RTCEN bit to 0 and the SOSCCR.SOSTP bit to 1 (general port PH7). When this pin is not also used as port PH7, handle it in the same way as the input setting of ports 0 to 5, 9, A to F, H, and J.
XCOUT	Leave this pin open.
Port 0 to 3, 5, 9, A to F, H, J (for pins that exist on products with fewer than 100 pins)	<ul style="list-style-type: none"> <li>Set these pins to input (PORTn.PDR bit = 0) and connect each of them to VCC via a pull-up resistor or to VSS via a pull-down resistor. *1</li> <li>Set these pins to output (PORTn.PDR bit = 1), set the output data to 0 (PORTn.PODR bit = 0), and leave them open. *1, *2</li> </ul>
Port 4 (for pins that exist on products with fewer than 100 pins)	<ul style="list-style-type: none"> <li>Set these pins to input (PORTn.PDR bit = 0) and connect each of them to AVCC0 via a pull-up resistor or to AVSS0 via a pull-down resistor. *1</li> </ul>
Port 0 to 5, 9, A to F, H, J (for pins that do not exist on products with fewer than 100 pins)	Set these pins to output (PORTn.PDR bit = 1), set the output data to 0 (PORTn.PODR bit = 0), and leave them open. *1, *2
VREFH0	When this pin is not used as VREFH0, set the PJ6PFS.ASEL bit to 0 (general port PJ6). When this pin is not also used as port PJ6, handle it in the same way as the handling of port 4.
VREFL0	When this pin is not used as VREFL0, set the PJ7PFS.ASEL bit to 0 (general port PJ7). When this pin is not also used as port PJ7, handle it in the same way as the handling of port 4.
VREFH	When this pin is not used as VREFH, set the P41PFS.ASEL bit to 0 (general port P41). When this pin is not also used as port P41, handle it in the same way as the handling of port 4.
VREFL	When this pin is not used as VREFL, set the P42PFS.ASEL bit to 0 (general port P42). When this pin is not also used as port P42, handle it in the same way as the handling of port 4.
VL1	When this pin is not used as VL1, set the P55PFS.PSEL bit to a value other than 11000b. When this pin is not also used as port P55, handle it in the same way as the handling of ports 0 to 5, 9, A to F, H, and J.
VL2	When this pin is not used as VL2, set the P54PFS.PSEL bit to a value other than 11000b. When this pin is not also used as port P54, handle it in the same way as the handling of ports 0 to 5, 9, A to F, H, and J.
VL3	When this pin is not used as VL3, set the PC7PFS.PSEL bit to a value other than 11000b. When this pin is not also used as port PC7, handle it in the same way as the handling of ports 0 to 5, 9, A to F, H, and J.
VL4	When this pin is not used as VL4, set the PC6PFS.PSEL bit to a value other than 11000b. When this pin is not also used as port PC6, handle it in the same way as the handling of ports 0 to 5, 9, A to F, H, and J.

Note 1. Set the PORTn.PMR bit to 0 and the PmnPFS.ISEL and ASEL bits to 0.

Note 2. If these ports are set to output mode and left open, they remain in input mode after the reset is released and before they are switched to output mode. The voltage level of these pins may be unstable and the power current may increase while the ports remain in input mode.

## 19. Multi-Function Pin Controller (MPC)

### 19.1 Overview

The multi-function pin controller (MPC) is used to allocate input and output signals for peripheral modules and input interrupt signals to pins from among multiple ports.

Table 19.1 shows the allocation of pin functions to multiple pins. A and N/A in the table indicate whether the pins are available or not available on the given package. Allocating the same function to more than one pin is prohibited.

**Table 19.1 Allocation of Pin Functions to Multiple Pins (1/10)**

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				100-pin	64-pin
Interrupt		NMI (input)	P35	A	A
Interrupt	IRQ0	IRQ0 (input)	P30	A	A
			PE0	A	A
			PD0	A	A
	IRQ1	IRQ1 (input)	P31	A	A
			PE1	A	A
			PD1	A	A
	IRQ2	IRQ2 (input)	P32	A	A
			PB0	A	A
			PC4	A	A
			P12	A	N/A
			PD2	A	A
	IRQ3	IRQ3 (input)	P27	A	A
			PE3	A	A
			PA6	A	A
			P13	A	N/A
	IRQ4	IRQ4 (input)	PD3	A	N/A
			P14	A	A
			PB1	A	A
			PE4	A	A
	IRQ5	IRQ5 (input)	PD4	A	N/A
			P15	A	A
			PA4	A	A
			PE5	A	A
	IRQ6	IRQ6 (input)	P56	A	N/A
P16			A	A	
PA3			A	A	
PE6			A	A	
IRQ7	IRQ7 (input)	P10	A	N/A	
		P17	A	A	
		PE2	A	A	
		PE7	A	A	
			P11	A	N/A

Table 19.1 Allocation of Pin Functions to Multiple Pins (2/10)

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				100-pin	64-pin
Multi-function timer unit 2	MTU0	MTIOC0A (I/O)	P14	A	A
			PB3	A	A
			PE3	A	A
			P04	A	N/A
		MTIOC0B (I/O)	P15	A	A
			PA1	A	A
			P13	A	N/A
		MTIOC0C (I/O)	P17	A	A
			P32	A	A
			PB0	A	A
			PB1	A	A
		MTIOC0D (I/O)	PA3	A	A
	P02		A	N/A	
	MTU1	MTIOC1A (I/O)	PE4	A	A
			P20	A	N/A
			P56	A	N/A
		MTIOC1B (I/O)	PA3	A	A
			PB5	A	A
			PE3	A	A
	MTU2	MTIOC2A (I/O)	P26	A	A
			PA6	A	A
			PB5	A	A
			PE0	A	A
			P50	A	N/A
MTIOC2B (I/O)		P27	A	A	
		PA4	A	A	
		PE5	A	A	
		P53	A	N/A	
MTU3	MTIOC3A (I/O)	P14	A	A	
		P17	A	A	
		PC7	A	A	
		PE4	A	A	
		PC1	A	N/A	
		PF7	A	N/A	
	MTIOC3B (I/O)	P17	A	A	
		PB3	A	A	
		PB7	A	A	
		PC5	A	A	
			P22	A	N/A

Table 19.1 Allocation of Pin Functions to Multiple Pins (3/10)

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				100-pin	64-pin
Multi-function timer unit 2	MTU3	MTIOC3C (I/O)	P16	A	A
			PC6	A	A
			PJ3	A	N/A
			PC0	A	N/A
			PF6	A	N/A
		MTIOC3D (I/O)	P16	A	A
			PB6	A	A
			PC4	A	A
			P23	A	N/A
	MTU4	MTIOC4A (I/O)	PA0	A	A
			PB3	A	A
			PE2	A	A
			P24	A	N/A
		MTIOC4B (I/O)	P30	A	A
			P54	A	A
			PC2	A	A
			PE3	A	A
			PD1	A	A
		MTIOC4C (I/O)	PB1	A	A
			PE1	A	A
			PE5	A	A
			P25	A	N/A
			P51	A	N/A
MTIOC4D (I/O)	P31	A	A		
	P55	A	A		
	PC3	A	A		
	PE4	A	A		
	PD2	A	A		
MTU5	MTIC5U (input)	PA4	A	A	
		P11	A	N/A	
	MTIC5V (input)	PA6	A	A	
		P10	A	N/A	
	MTIC5W (input)	PB0	A	A	
		P56	A	N/A	
MTU	MTCLKA (input)	P14	A	A	
		PA4	A	A	
		PC6	A	A	
		P24	A	N/A	
	MTCLKB (input)	P15	A	A	
		PA6	A	A	
		PC7	A	A	
		P25	A	N/A	

**Table 19.1 Allocation of Pin Functions to Multiple Pins (4/10)**

Module/Function	Channel	Pin Functions	Allocation Port	Package		
				100-pin	64-pin	
Multi-function timer unit 2	MTU	MTCLKC (input)	PA1	A	A	
			PC4	A	A	
			P22	A	N/A	
	MTCLKD (input)	PA3	A	A		
		PC5	A	A		
		P23	A	N/A		
Port output enable 2	POE0	POE0# (input)	PC4	A	A	
			PA3	A	A	
			P11	A	N/A	
	POE1	POE1# (input)	PB5	A	A	
			P10	A	N/A	
	POE2	POE2# (input)	PA6	A	A	
			P04	A	N/A	
			P56	A	N/A	
	POE3	POE3# (input)	PB3	A	A	
			PE0	A	A	
			P02	A	N/A	
			PD4	A	N/A	
	POE8	POE8# (input)	P17	A	A	
			P30	A	A	
			PE3	A	A	
			PD3	A	N/A	
	8-bit timer	TMR0	TMO0 (output)	P22	A	N/A
				PB3	A	A
TMC10 (input)			P21	A	N/A	
			PB1	A	A	
TMR10 (input)			P20	A	N/A	
			PA4	A	A	
TMR1		TMO1 (output)	P17	A	A	
			P26	A	A	
		TMC11 (input)	P12	A	N/A	
			P54	A	A	
		TMR11 (input)	P24	A	N/A	
			PB5	A	A	
TMR2		TMO2 (output)	P16	A	A	
			PC7	A	A	
		TMC12 (input)	P15	A	A	
			P31	A	A	
		TMR12 (input)	PC6	A	A	
			P14	A	A	
			PC5	A	A	

**Table 19.1 Allocation of Pin Functions to Multiple Pins (5/10)**

Module/Function	Channel	Pin Functions	Allocation Port	Package		
				100-pin	64-pin	
8-bit timer	TMR3	TMO3 (output)	P13	A	N/A	
			P32	A	A	
			P55	A	A	
		TMC13 (input)	P04	A	N/A	
			P27	A	A	
			PA6	A	A	
		TMR13 (input)	P02	A	N/A	
			P30	A	A	
		Serial communications interface	SCI0	RXD0 (input)/ SMISO0 (I/O)/ SSCL0 (I/O)	P21	A
P11	A				N/A	
TXD0 (output)/ SMOS10 (I/O)/ SSDA0 (I/O)	P20			A	N/A	
	P10			A	N/A	
SCK0 (I/O)	P22			A	N/A	
	P12			A	N/A	
CTS0# (input)/ RTS0# (output)/ SS0# (input)	P23			A	N/A	
	P13			A	N/A	
SCI1	RXD1 (input)/ SMISO1 (I/O)/ SSCL1 (I/O)			P15	A	A
				P30	A	A
				PC6	A	A
	TXD1 (output)/ SMOS11 (I/O)/ SSDA1 (I/O)			P16	A	A
			P26	A	A	
			PC7	A	A	
	SCK1 (I/O)		P56	A	N/A	
			P17	A	A	
			P27	A	A	
CTS1# (input)/ RTS1# (output)/ SS1# (input)	PC5		A	A		
	P14		A	A		
SCI2	RXD2 (input)/ SMISO2 (I/O)/ SSCL2 (I/O)		P52	A	N/A	
			TXD2 (output)/ SMOS12 (I/O)/ SSDA2 (I/O)	P50	A	N/A
		SCK2 (I/O)		P51	A	N/A
		CTS2# (input)/ RTS2# (output)/ SS2# (input)	P53	A	N/A	



**Table 19.1 Allocation of Pin Functions to Multiple Pins (6/10)**

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				100-pin	64-pin
Serial communications interface	SCI5/IRDA	RXD5 (input)/ SMISO5 (I/O)/ SSCL5 (I/O)/ IRRXD5 (input)	PA3	A	A
			PC2	A	A
			PA2	A	N/A
		TXD5 (output)/ SMOSI5 (I/O)/ SSDA5 (I/O)/ IRTXD5 (output)	PA4	A	A
			PC3	A	A
			PC1	A	N/A
		SCK5 (I/O)	PA1	A	A
			PC4	A	A
		CTS5# (input)/ RTS5# (output)/ SS5# (input)	PA6	A	A
			PC0	A	N/A
	SCI6	RXD6 (input)/ SMISO6 (I/O)/ SSCL6 (I/O)	P02	A	N/A
			P27	A	A
			PB0	A	A
		TXD6 (output)/ SMOSI6 (I/O)/ SSDA6 (I/O)	P07	A	N/A
			P26	A	A
			PB1	A	A
		SCK6 (I/O)	P32	A	A
			P04	A	N/A
		CTS6# (input)/ RTS6# (output)/ SS6# (input)	PB3	A	A
			P32	A	A
PJ3	A		N/A		
SCI8	RXD8 (input)/ SMISO8 (I/O)/ SSCL8 (I/O)	PB2	A	N/A	
		PC6	A	A	
		PA6	A	N/A	
	TXD8 (output)/ SMOSI8 (I/O)/ SSDA8 (I/O)	PC7	A	A	
		PA7	A	N/A	
	SCK8 (I/O)	PC5	A	A	
SCI9	RXD9 (input)/ SMISO9 (I/O)/ SSCL9 (I/O)	PA5	A	N/A	
		PC4	A	A	
		PA4	A	N/A	
	CTS8# (input)/ RTS8# (output)/ SS8# (input)	PC4	A	A	
		PA4	A	N/A	
	SCI9	RXD9 (input)/ SMISO9 (I/O)/ SSCL9 (I/O)	PB6	A	A
			PE4	A	A
		TXD9 (output)/ SMOSI9 (I/O)/ SSDA9 (I/O)	PB7	A	A
			PE5	A	A
		SCK9 (I/O)	PB5	A	A
CTS9# (input)/ RTS9# (output)/ SS9# (input)	PE3	A	A		
	PB4	A	N/A		
		PE0	A	A	

**Table 19.1 Allocation of Pin Functions to Multiple Pins (7/10)**

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				100-pin	64-pin
Serial communications interface	SCI12	RXD12 (input)/ SMISO12 (I/O)/ SSCL12 (I/O)/ RXDX12 (input)	PE2	A	A
			P17	A	A
			P11	A	N/A
		TXD12 (output)/ SMOS12 (I/O)/ SSDA12 (I/O)/ TXDX12 (output)/ SIOX12 (I/O)	PE1	A	A
			P14	A	A
			P10	A	N/A
		SCK12 (I/O)	PE0	A	A
			P27	A	A
			P12	A	N/A
		CTS12# (input)/ RTS12# (output)/ SS12# (input)	PE3	A	A
P13	A		N/A		
I <sup>2</sup> C bus interface	RIIC0	SCL0 (I/O)	P16	A	A
			PB0	A	A
		SDA0 (I/O)	P17	A	A
			PA6	A	A
Serial peripheral interface	RSPI0	RSPCKA (I/O)	P15	A	A
			PB0	A	A
			PC5	A	A
			PE3	A	A
			P51	A	N/A
		MOSIA (I/O)	P16	A	A
			PA6	A	A
			PE4	A	A
			PC6	A	A
			P50	A	N/A
		MISOA (I/O)	P17	A	A
			PC7	A	A
			PA3	A	A
			P52	A	N/A
			PE5	A	A
		SSLA0 (I/O)	P14	A	A
			PA4	A	A
			PC4	A	A
			P53	A	N/A
			SSLA1 (output)	PA0	A
		PC0		A	N/A
		SSLA2 (output)	PA1	A	A
			PC1	A	N/A
		SSLA3 (output)	PC2	A	A
PA2	A		N/A		

**Table 19.1 Allocation of Pin Functions to Multiple Pins (8/10)**

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				100-pin	64-pin
USB 2.0 host/function module	USB0	USB0_EXICEN (output)	PC6	A	A
		USB0_VBUSEN (output)	P16	A	A
			PC4	A	A
			P26	A	A
		USB0_OVRCURA (input)	P14	A	A
			PB3	A	A
		USB0_OVRCURB (input)	P16	A	A
			PC7	A	A
		USB0_ID (input)	PC5	A	A
		USB0_VBUS (input)* <sup>2</sup>	P16	A	A
USB0_VBUS (input)* <sup>3</sup>	PC4	A	A		
Realtime clock	RTCOUT (output)	P16	A	A	
		P32	A	A	
		PB0	A	A	
		PA1	A	A	
12-bit A/D converter		AN000 (input)* <sup>1</sup>	P40	A	A
		AN001 (input)* <sup>1</sup>	P41	A	A
		AN002 (input)* <sup>1</sup>	P42	A	A
		AN003 (input)* <sup>1</sup>	P43	A	N/A
		AN004 (input)* <sup>1</sup>	P44	A	N/A
		AN005 (input)* <sup>1</sup>	P90	A	N/A
		AN006 (input)* <sup>1</sup>	P46	A	N/A
		AN007 (input)* <sup>1</sup>	P91	A	N/A
		AN008 (input)* <sup>1</sup>	PE0	A	A
		AN009 (input)* <sup>1</sup>	PE1	A	A
		AN010 (input)* <sup>1</sup>	PE2	A	A
		AN011 (input)* <sup>1</sup>	PE3	A	A
		AN012 (input)* <sup>1</sup>	PE4	A	A
		AN013 (input)* <sup>1</sup>	PE5	A	A
		AN014 (input)* <sup>1</sup>	PE6	A	A
		AN015 (input)* <sup>1</sup>	PE7	A	A
		AN0021 (input)* <sup>1</sup>	P92	A	N/A
		VREFH0 (input)	PJ6	A	A
		VREFL0 (input)	PJ7	A	A
		ADTRG0# (input)	P16	A	A
			P27	A	A
PB0	A		A		
P25	A		N/A		
P07	A		N/A		
12-bit D/A converter		DA0 (output)* <sup>1</sup>	PJ0	A	A
		DA1 (output)* <sup>1</sup>	PJ2	A	A
		VREFH	P41	A	A
		VREFL	P42	A	A

**Table 19.1 Allocation of Pin Functions to Multiple Pins (9/10)**

Module/Function	Channel	Pin Functions	Allocation Port	Package		
				100-pin	64-pin	
Clock		CLKOUT (output)	P15	A	A	
			PC4	A	A	
Clock frequency accuracy measurement circuit		CACREF (input)	P27	A	A	
			PA0	A	A	
			PC7	A	A	
			P15	A	A	
Voltage detection circuit		CMPA2 (input)*1	P27	A	A	
Comparator B		CMPB0 (input)*1	PE1	A	A	
		CVREFB0 (input)*1	PE2	A	A	
		CMPOB0 (output)	PE7	A	A	
		CMPB1 (input)*1	PA3	A	A	
		CVREFB1 (input)*1	PA4	A	A	
		CMPOB1 (output)	PE5	A	A	
Serial sound interface	SSIO	SSISCK0 (I/O)	PB5	A	A	
			PE0	A	A	
		SSIWS0 (I/O)	PB1	A	A	
			PE4	A	A	
		SSIRXD0 (input)	PB6	A	A	
			PE2	A	A	
	SSITXD0 (input)	PB7	A	A		
		PE1	A	A		
	SSI	AUDIO_MCLK (I/O)	PB3	A	A	
			PE3	A	A	
	LCD Controller/Driver		COM0 (output)	PC5	A	A
			COM1 (output)	PC4	A	A
COM2 (output)			PC3	A	A	
COM3 (output)			PC2	A	A	
SEG00 (output)			P13	A	N/A	
SEG01 (output)			P12	A	N/A	
SEG02 (output)			P11	A	N/A	
SEG03 (output)			P10	A	N/A	
SEG04 (output)			P56	A	N/A	
SEG05 (output)			P53	A	N/A	
SEG06 (output)			P52	A	N/A	
SEG07 (output)			P51	A	N/A	
SEG08 (output)			P50	A	N/A	
SEG09 (output)			PC1	A	N/A	
SEG10 (output)			PC0	A	N/A	
SEG11/COM4 (output)			PB7	A	A	
SEG12/COM5 (output)			PB6	A	A	
SEG13/COM6 (output)	PB5	A	A			
SEG14 (output)	PB4	A	N/A			
SEG15/COM7 (output)	PB3	A	A			
SEG16 (output)	PB2	A	N/A			

**Table 19.1 Allocation of Pin Functions to Multiple Pins (10/10)**

Module/Function	Channel	Pin Functions	Allocation Port	Package		
				100-pin	64-pin	
LCD Controller/Driver		SEG17 (output)	PB1	A	A	
		SEG18 (output)	PA7	A	N/A	
		SEG19 (output)	PA5	A	N/A	
		SEG20 (output)	PA4	A	A	
		SEG21 (output)	PA3	A	A	
		SEG22 (output)	PA2	A	N/A	
		SEG23 (output)	PA1	A	A	
		SEG24 (output)	PA0	A	A	
		SEG25 (output)	PF7	A	N/A	
		SEG26 (output)	PF6	A	N/A	
		SEG27 (output)	PE5	A	A	
		SEG28 (output)	PE4	A	A	
		SEG29 (output)	PE3	A	A	
		SEG30 (output)	PE2	A	A	
		SEG31 (output)	PE1	A	A	
		SEG32 (output)	PE0	A	A	
		SEG33 (output)	PE7	A	A	
		SEG34 (output)	PE6	A	A	
		SEG35 (output)	PD4	A	N/A	
		SEG36 (output)	PD3	A	N/A	
		SEG37 (output)	PD2	A	A	
		SEG38 (output)	PD1	A	A	
		SEG39 (output)	PD0	A	A	
		CAPH (output)	P30	A	A	
		CAPL (output)	P31	A	A	
		VL1 (I/O)	P55	A	A	
		VL2 (I/O)	P54	A	A	
		VL3 (I/O)	PC7	A	A	
		VL4 (I/O)	PC6	A	A	
	Capacitive touch sensing unit		TS0 (output)	P07	A	N/A
			TS1 (output)	P04	A	N/A
			TS2 (output)	P02	A	N/A
			TS3 (output)	PJ3	A	N/A
			TS4 (output)	P25	A	N/A
		TS5 (output)	P24	A	N/A	
		TS6 (output)	P23	A	N/A	
		TS7 (output)	P22	A	N/A	
		TS8 (output)	P21	A	N/A	
		TS9 (output)	P20	A	N/A	
		TS10 (output)	P27	A	N/A	
		TS11 (output)	P32	A	N/A	
	TSCAP (I/O)	P26	A	N/A		

Note 1. Select general input (by setting the Bm bits for the given pin in the PDR and PMR for the given port to 0) for the pin if this pin function is to be used.

Note 2. 5 V tolerant.

Note 3. Not 5 V tolerant.

## 19.2 Register Descriptions

Registers and bits for pins that are not present due to differences according to the package are reserved. Write the value after a reset when writing to such bits.

### 19.2.1 Write-Protect Register (PWPR)

Address(es): 0008 C11Fh

b7	b6	b5	b4	b3	b2	b1	b0
B0WI	PFSWE	—	—	—	—	—	—

Value after reset: 1 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PFSWE	PFS Register Write Enable	0: Writing to the PFS register is disabled 1: Writing to the PFS register is enabled	R/W
b7	B0WI	PFSWE Bit Write Disable	0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled	R/W

#### PFSWE Bit (PFS Register Write Enable)

Writing to the PmnPFS register is enabled only when the PFSWE bit is set to 1.

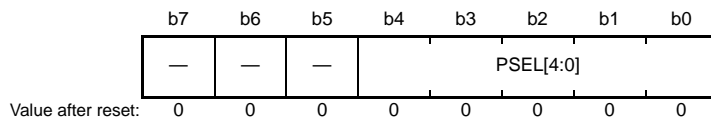
To set the PFSWE bit to 1, write 1 to the PFSWE bit after writing 0 to the B0WI bit.

#### B0WI Bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

### 19.2.2 P0n Pin Function Control Register (P0nPFS) (n = 2, 4, 7)

Address(es): P02PFS 0008 C142h, P04PFS 0008 C144h, 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, refer to the table below.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The Pmn pin function control register (PmnPFS) selects the pin function.

The PSEL[4:0] bits select the peripheral function assigned to each pin.

The ISEL bit is set when a pin is used as an IRQ input pin. This setting can be used with the combination of the peripheral function, though IRQn (external pin interrupt) of the same number should not be enabled by two or more pins.

The ASEL bit is set when a pin is used as an analog pin. When switching a pin to analog using the ASEL bit, set the corresponding port mode register bit (PORTm.PMR) to “general I/O port” and the port direction register bit (PORTm.PDR) to “input”. The pin state cannot be read at this point. 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 19.2 Register Settings for I/O Pin Functions in 100-Pin**

PSEL[4:0] Settings	Pin		
	P02	P04	P07
00000b (initial value)	Hi-Z		
00001b	MTIOC0D	MTIOC0A	—
00101b	TMRI3	TMCI3	—
00111b	POE3#	POE2#	—
01001b	—	—	ADTRG0#
01011b	RXD6 SMISO6 SSCL6	SCK6	TXD6 SMOSI6 SSDA6
11001b	TS2	TS1	TS0

### 19.2.3 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, refer to the table below.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P10: IRQ6 (100 pins) P11: IRQ7 (100 pins) P12: IRQ2 (100 pins) P13: IRQ3 (100 pins) P14: IRQ4 (100 pins, 64 pins) P15: IRQ5 (100 pins, 64 pins) P16: IRQ6 (100 pins, 64 pins) P17: IRQ7 (100 pins, 64 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 19.3 Register Settings for I/O Pin Functions in 100-Pin**

PSEL[4:0] Settings	Pin							
	P10	P11	P12	P13	P14	P15	P16	P17
00000b (initial value)	Hi-Z							
00001b	MTIC5V	MTIC5U	—	MTIOC0B	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
00010b	—	—	—	—	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
00011b	—	—	—	—	MTIOC0A	—	—	MTIOC0C
00101b	—	—	TMC11	TMO3	TMRI2	TMC12	TMO2	TMO1
00111b	POE1#	POE0#	—	—	—	CACREF	RTCOUT	POE8#
01001b	—	—	—	—	—	CLKOUT	ADTRG0#	—
01010b	TXD0 SMOS10 SSDA0	RXD0 SMIS00 SSCL0	—	—	—	RXD1 SMIS01 SSCL1	TXD1 SMOS11 SSDA1	SCK1
01011b	—	—	SCK0	CTS0# RTS0# SS0#	CTS1# RTS1# SS1#	—	—	—
01100b	TXD12 SMOS12 SSDA12 TXDX12 SIOX12	RXD12 SMIS012 SSCL12 RXDX12	SCK12	CTS12# RTS12# SS12#	TXD12 SMOS12 SSDA12 TXDX12 SIOX12	—	—	RXD12 SMIS012 SSCL12 RXDX12
01101b	—	—	—	—	SSLA0	RSPCKA	MOSIA	MISOA
01111b	—	—	—	—	—	—	SCL0	SDA0
10001b	—	—	—	—	—	—	USB0_VBUSEN	—
10010b	—	—	—	—	—	—	USB0_VBUS*1	—
10011b	—	—	—	—	USB0_OVRCURA	—	USB0_OVRCURB	—
11000b	SEG03	SEG02	SEG01	SEG00	—	—	—	—

—: Do not specify this value.

Note 1. 5 V tolerant.



**Table 19.4 Register Settings for I/O Pin Functions in 64-Pin**

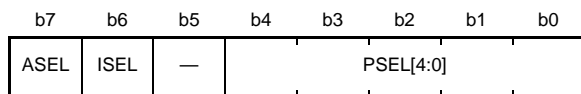
PSEL[4:0] Settings	Pin			
	P14	P15	P16	P17
0000b (initial value)	Hi-Z			
0001b	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
00010b	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
00011b	MTIOC0A	—	—	MTIOC0C
00101b	TMRI2	TMCI2	TMO2	TMO1
00111b	—	CACREF	RTCOUT	POE8#
01001b	—	CLKOUT	ADTRG0#	—
01010b	—	RXD1 SMISO1 SSCL1	TXD1 SMOS1 SSDA1	SCK1
01011b	CTS1# RTS1# SS1#	—	—	—
01100b	TXD12 SMOS12 SSDA12 TXDX12 SIOX12	—	—	RXD12 SMISO12 SSCL12 RXDX12
01101b	SSLA0	RSPCKA	MOSIA	MISOA
01111b	—	—	SCL0	SDA0
10001b	—	—	USB0_VBUSEN	—
10010b	—	—	USB0_VBUS*1	—
10011b	USB0_OVRCURA	—	USB0_OVRCURB	—

—: Do not specify this value.

Note 1. 5 V tolerant.

### 19.2.4 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



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, refer to the table below.	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 P27: IRQ3 (100 pins, 64 pins)	R/W
b7	ASEL	Analog Function Select	0: Used as other than an analog pin 1: Used as an analog pin P27: CMPA2 (100 pins, 64 pins)	R/W

**Table 19.5 Register Settings for I/O Pin Functions in 100-Pin**

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	—	—
00101b	TMRI0	TMCi0	TMO0	—	TMR11	—	TMO1	TMCi3
00111b	—	—	—	—	—	—	—	CACREF
01001b	—	—	—	—	—	ADTRG0#	—	ADTRG0#
01010b	TXD0 SMOSi0 SSDA0	RXD0 SMISO0 SSCL0	SCK0	—	—	—	TXD1 SMOSi1 SSDA1	SCK1
01011b	—	—	—	CTS0# RTS0# SS0#	—	—	—	RXD6 SMISO6 SSCL6
01100b	—	—	—	—	—	—	TXD6 SMOSi6 SSDA6	SCK12
10011b	—	—	—	—	—	—	USB0_VBUSEN	—
11001b	TS9	TS8	TS7	TS6	TS5	TS4	TSCAP	TS10

—: Do not specify this value.

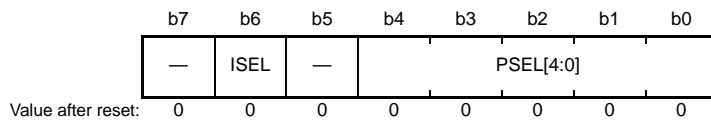
**Table 19.6 Register Settings for I/O Pin Functions in 64-Pin**

PSEL[4:0] Settings	Pin	
	P26	P27
00000b (initial value)	Hi-Z	
00001b	MTIOC2A	MTIOC2B
00101b	TMO1	TMCI3
00111b	—	CACREF
01001b	—	ADTRG0#
01010b	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	RXD6 SMISO6 SSCL6
01100b	TXD6 SMOSI6 SSDA6	SCK12
10011b	USB0_VBUSEN	—

—: Do not specify this value.

### 19.2.5 P3n Pin Function Control Registers (P3nPFS) (n = 0 to 2)

Address(es): P30PFS 0008 C158h, P31PFS 0008 C159h, P32PFS 0008 C15Ah



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, refer to the table below.	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 and NMI input pin 1: Used as IRQn and NMI input pin P30: IRQ0 (100 pins, 64 pins) P31: IRQ1 (100 pins, 64 pins) P32: IRQ2 (100 pins, 64 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 19.7 Register Settings for I/O Pin Functions in 100-Pin**

PSEL[4:0] Settings	Pin		
	P30	P31	P32
00000b (initial value)	Hi-Z		
00001b	MTIOC4B	MTIOC4D	MTIOC0C
00101b	TMRI3	TMCI2	TMO3
00111b	POE8#	—	RTCOUT
01010b	RXD1 SMISO1 SSCL1	—	—
01011b	—	CTS1# RTS1# SS1#	TXD6 SMOSI6 SSDA6
01100b	—	—	CTS6# RTS6# SS6#
11000b	CAPH	CAPL	—
11001b	—	—	TS11

—: Do not specify this value.

Note: When using the CAPH and CAPL pins as ports or pins for peripheral functions other than the LCD, set the value of the LCDM0 and LCDM1 registers to 00h.

**Table 19.8 Register Settings for I/O Pin Functions in 64-Pin**

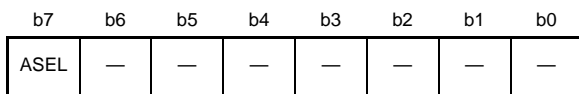
PSEL[4:0] Settings	Pin		
	P30	P31	P32
00000b (initial value)	Hi-Z		
00001b	MTIOC4B	MTIOC4D	MTIOC0C
00101b	TMRI3	TMCI2	TMO3
00111b	POE8#	—	RTCOUT
01010b	RXD1 SMISO1 SSCL1	—	—
01011b	—	CTS1# RTS1# SS1#	TXD6 SMOSI6 SSDA6
01100b	—	—	CTS6# RTS6# SS6#
11000b	CAPH	CAPL	—

—: Do not specify this value.

Note: When using the CAPH and CAPL pins as ports or pins for peripheral functions other than the LCD, set the value of the LCDM0 and LCDM1 registers to 00h.

### 19.2.6 P4n Pin Function Control Registers (P4nPFS) (n = 0 to 4, 6)

Address(es): P40PFS 0008 C160h, P41PFS 0008 C161h, P42PFS 0008 C162h, P43PFS 0008 C163h, P44PFS 0008 C164h, P46PFS 0008 C166h



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

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Function Select	0: Used as other than an analog pin 1: Used as an analog pin P40: AN002 (100 pins, 64 pins) P41: AN001/VREFH (100 pins, 64 pins) P42: AN002/VREFL (100 pins, 64 pins) P43: AN003 (100 pins) P44: AN004 (100 pins) P46: AN006 (100 pins)	R/W

### 19.2.7 P5n Pin Function Select Registers (P5nPFS) (n = 0 to 6)

Address(es): P50PFS 0008 C168h, P51PFS 0008 C169h, P52PFS 0008 C16Ah, P53PFS 0008 C16Bh,  
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, refer to the table below.	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 or NMI input pin 1: Used as IRQn or NMI input pin P56: IRQ5 (100 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 19.9 Register Settings for I/O Pin Functions in 100-Pin**

PSEL[4:0] Settings	Pin						
	P50	P51	P52	P53	P54	P55	P56
00000b (initial value)	Hi-Z						
00001b	MTIOC2A	MTIOC4C	—	MTIOC2B	MTIOC4B	MTIOC4D	MTIC5W
00011b	—	—	—	—	—	—	MTIOC1A
00101b	—	—	—	—	TMCI1	TMO3	—
00111b	—	—	—	—	—	—	POE2#
01010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	CTS2# RTS2# SS2#	—	—	TXD1 SMOSI1 SSDA1
01101b	MOSIA	RSPCKA	MISOA	SSLA0	—	—	—
11000b	SEG08	SEG07	SEG06	SEG05	VL2	VL1	SEG04

—: Do not specify this value.

Note: When using the CAPH and CAPL pins as ports or pins for peripheral functions other than the LCD, set the value of the LCDM0 and LCDM1 registers to 00h.

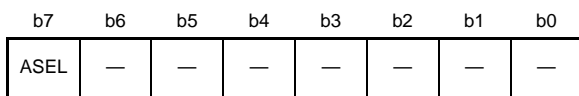
**Table 19.10 Register Settings for I/O Pin Functions in 64-Pin**

PSEL[4:0] Settings	Pin	
	P54	P55
00000b (initial value)	Hi-Z	
00001b	MTIOC4B	MTIOC4D
00101b	TMCI1	TMO3
11000b	VL2	VL1

Note: When using the CAPH and CAPL pins as ports or pins for peripheral functions other than the LCD, set the value of the LCDM0 and LCDM1 registers to 00h.

### 19.2.8 P9n Pin Function Select Registers (P9nPFS) (n = 0 to 2)

Address(es): P90PFS 0008 C188h, P91PFS 0008 C189h, P92PFS 0008 C18Ah



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

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Function Select	0: Used as a pin other than an analog pin 1: Used as an analog pin P90: AN005 (100 pins) P91: AN007 (100 pins) P92: AN021 (100 pins)	R/W



### 19.2.9 PAn Pin Function Control 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, refer to the table below.	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 PA3: IRQ6 (100 pins, 64 pins) PA4: IRQ5 (100 pins, 64 pins) PA6: IRQ3 (100 pins, 64 pins)	R/W
b7	ASEL	Analog Function Select	0: Used as a pin other than an analog pin 1: Used as an analog pin PA3: CMPB1 (100 pins, 64 pins) PA4: CVREFB1 (100 pins, 64 pins)	R/W

**Table 19.11 Register Settings for I/O Pin Functions in 100-Pin**

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	—
00101b	—	—	—	MTIOC1B	MTIOC2B	—	MTIOC2A	—
00011b	—	—	—	—	TMRI0	—	TMCI3	—
00111b	CACREF	RTCOUT	—	POE0#	—	—	POE2#	—
01010b	—	SCK5	RXD5 SMISO5 SSCL5 IRRXD5	RXD5 SMISO5 SSCL5 IRRXD5	TXD5 SMOSI5 SSDA5 IRTXD5	SCK8	RXD8 SMISO8 SSCL8 SSDA8	TXD8 SMOSI8
01011b	—	—	—	—	CTS8# RTS8# SS8#	—	CTS5# RTS5# SS5#	—
01101b	SSLA1	SSLA2	SSLA3	MISOA	SSLA0	—	MOSIA	—
01111b	—	—	—	—	—	—	SDA0	—
11000b	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	—	SEG18

—: Do not specify this value.

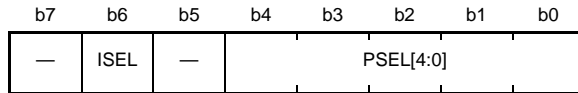
**Table 19.12 Register Settings for I/O Pin Functions in 64-Pin**

PSEL[4:0] Settings	Pin				
	PA0	PA1	PA3	PA4	PA6
00000b (initial value)	Hi-Z				
00001b	MTIOC4A	MTIOC0B	MTIOC0D	MTIC5U	MTIC5V
00010b	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
00101b	—	—	MTIOC1B	MTIOC2B	MTIOC2A
00011b	—	—	—	TMRI0	TMCI3
00111b	CACREF	RTCOUT	POE0#	—	POE2#
01010b	—	SCK5	RXD5 SMISO5 SSCL5 IRRXD5	TXD5 SMOSI5 SSDA5 IRTXD5	—
01011b	—	—	—	—	CTS5# RTS5# SS5#
01101b	SSLA1	SSLA2	MISOA	SSLA0	MOSIA
01111b	—	—	—	—	SDA0
11000b	SEG24	SEG23	SEG21	SEG20	—

—: Do not specify this value.

### 19.2.10 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 Input/Output Function	These bits select the peripheral function. For individual pin functions, refer to the table below.	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 PB0: IRQ2 (100 pins, 64 pins) PB1: IRQ4 (100 pins, 64 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 19.13 Register Settings for I/O Pin Functions in 100-Pin**

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	MTIOC0C	MTIOC4C	—	MTIOC4A	—	MTIOC1B	—	—
00011b	—	—	—	MTIOC3B	—	—	—	—
00101b	—	TMCIO	—	TMO0	—	TMR1	—	—
00111b	RTCOUT	—	—	POE3#	—	POE1#	—	—
01001b	ADTRG0#	—	—	—	—	—	—	—
01010b	—	—	—	—	—	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9
01011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	CTS6# RTS6# SS6#	SCK6	CTS9# RTS9# SS9#	—	—	—
01101b	RSPCKA	—	—	—	—	—	—	—
01111b	SCL0	—	—	—	—	—	—	—
10011b	—	—	—	USB0_OVR CURA	—	—	—	—
10111b	—	SSIWS0	—	AUDIO_CL K	—	SSISCK0	SSIRXD0	SSITXD0
11000b	—	SEG17	SEG16	COM7 SEG15	SEG14	COM6 SEG13	COM5 SEG12	COM4 SEG11

—: Do not specify this value.

**Table 19.14 Register Settings for I/O Pin Functions in 64-Pin**

PSEL[4:0] Settings	Pin					
	PB0	PB1	PB3	PB5	PB6	PB7
0000b (initial value)	Hi-Z					
00001b	MTIC5W	MTIOC0C	MTIOC0A	MTIOC2A	MTIOC3D	MTIOC3B
00010b	MTIOC0C	MTIOC4C	MTIOC4A	MTIOC1B	—	—
00011b	—	—	MTIOC3B	—	—	—
00101b	—	TMCIO	TMO0	TMR1	—	—
00111b	RTCOUT	—	POE3#	POE1#	—	—
01001b	ADTRG0#	—	—	—	—	—
01010b	—	—	—	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9
01011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	SCK6	—	—	—
01101b	RSPCKA	—	—	—	—	—
01111b	SCL0	—	—	—	—	—
10011b	—	—	USB0_OVRCURA	—	—	—
10111b	—	SSIWS0	AUDIO_MCLK	SSISCK0	SSIRXD0	SSITXD0
11000b	—	SEG17	COM7 SEG15	COM6 SEG13	COM5 SEG12	COM4 SEG11

—: Do not specify this value.

### 19.2.11 PCn Pin Function Control Registers (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 Input/Output Function	These bits select the peripheral function. For individual pin functions, refer to the table below.	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 PC4: IRQ2 (100 pins, 64 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 19.15 Register Settings for I/O Pin Functions in 100-Pin**

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
00101b	—	—	—	—	TMC11	TMR12	TMC112	TMO2
00111b	—	—	—	—	POE0#	—	—	CACREF
01001b	—	—	—	—	CLKOUT	—	—	—
01010b	—	SCK5	RXD5 SMISO5 SSCL5 IRRXD5	TXD5 SMOSI5 SSDA5 IRTXD5	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
01011b	CTS5# RTS5# SS5#	—	—	—	CTS8# RTS8# SS8#	SCK1	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
10001b	—	—	—	—	USB0_VBUSEN	—	—	—
10010b	—	—	—	—	USB0_VBUS*1	—	—	—
10011b	—	—	—	—	—	USB0_ID	USB0_EXICEN	USB0_OVRCURB
11000b	SEG10	SEG09	COM3	COM2	COM1	COM0	VL4	VL3

—: Do not specify this value.

Note: When using the CAPH and CAPL pins as ports or pins for peripheral functions other than the LCD, set the value of the LCDM0 and LCDM1 registers to 00h.

Note 1. Not 5 V tolerant.

**Table 19.16 Register Settings for I/O Pin Functions in 64-Pin**

PSEL[4:0] Settings	Pin					
	PC2	PC3	PC4	PC5	PC6	PC7
0000b (initial value)	Hi-Z					
00001b	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
00010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
00101b	—	—	TMC11	TMRI2	TMCI12	TMO2
00111b	—	—	POE0#	—	—	CACREF
01001b	—	—	CLKOUT	—	—	—
01010b	RXD5 SMISO5 SSCL5 IRRXD5	TXD5 SMOSI5 SSDA5 IRTXD5	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
01011b	—	—	CTS8# RTS8# SS8#	SCK1	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1
01101b	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
10001b	—	—	USB0_VBUSEN	—	—	—
10010b	—	—	USB0_VBUS*1	—	—	—
10011b	—	—	—	USB0_ID	USB0_EXICEN	USB0_OVRCURB
11000b	COM3	COM2	COM1	COM0	VL4	VL3

—: Do not specify this value.

Note: When using the CAPH and CAPL pins as ports or pins for peripheral functions other than the LCD, set the value of the LCDM0 and LCDM1 registers to 00h.

Note 1. Not 5 V tolerant.

### 19.2.12 PDn Pin Function Control Registers (PDnPFS) (n = 0 to 4)

Address(es): PD0PFS 0008 C1A8h, PD1PFS 0008 C1A9h, PD2PFS 0008 C1AAh, PD3PFS 0008 C1ABh, PD4PFS 0008 C1ACH



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 Input/Output Function	These bits select the peripheral function. For individual pin functions, refer to the table below.	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 PD0: IRQ0 (100 pins, 64 pins) PD1: IRQ1 (100 pins, 64 pins) PD2: IRQ2 (100 pins, 64 pins) PD3: IRQ3 (100 pins) PD4: IRQ4 (100 pins)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 19.17 Register Settings for I/O Pin Functions in 100-Pin**

PSEL[4:0] Settings	Pin				
	PD0	PD1	PD2	PD3	PD4
00000b (initial value)	Hi-Z				
00001b	—	MTIOC4B	MTIOC4BD	—	—
00111b	—	—	—	POE8#	POE3#
11000b	SEG39	SEG38	SEG37	SEG36	SEG35

—: Do not specify this value.

**Table 19.18 Register Settings for I/O Pin Functions in 64-Pin**

PSEL[4:0] Settings	Pin		
	PD0	PD1	PD2
00000b (initial value)	Hi-Z		
00001b	—	MTIOC4B	MTIOC4BD
11000b	SEG39	SEG38	SEG37

—: Do not specify this value.

### 19.2.13 PEn Pin Function Control Registers (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 to b0	PSEL[4:0]	PEn Pin Input/Output Function	These bits select the peripheral function. For individual pin functions, refer to the table below.	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 PE0: IRQ0 (100 pins, 64 pins) PE1: IRQ1 (100 pins, 64 pins) PE2: IRQ7 (100 pins, 64 pins) PE3: IRQ3 (100 pins, 64 pins) PE4: IRQ4 (100 pins, 64 pins) PE5: IRQ5 (100 pins, 64 pins) PE6: IRQ6 (100 pins, 64 pins) PE7: IRQ7 (100 pins, 64 pins)	R/W
b7	ASEL	PEn Analog Function Select	0: Used as other than an analog pin 1: Used as an analog pin PE0: AN008 (100 pins, 64 pins) PE1: AN009/CMPB0 (100 pins, 64 pins) PE2: AN010/CVREFB0 (100 pins, 64 pins) PE3: AN011 (100 pins, 64 pins) PE4: AN012 (100 pins, 64 pins) PE5: AN013 (100 pins, 64 pins) PE6: AN014 (100 pins, 64 pins) PE7: AN015 (100 pins, 64 pins)	R/W

**Table 19.19 Register Settings for I/O Pin Functions in 100-Pin and 64-Pin**

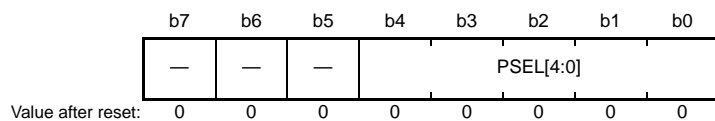
PSEL[4:0] Settings	Pin							
	PE0	PE1	PE2	PE3	PE4	PE5	PE6	PE7
00000b (initial value)	Hi-Z							
00001b	—	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D	MTIOC4C	—	—
00010b	MTIOC2A	—	—	MTIOC1B	MTIOC1A	MTIOC2B	—	—
00011b	—	—	—	MTIOC0A	MTIOC3A	—	—	—
00111b	POE3#	—	—	POE8#	—	CMPOB1	—	CMPOB0
01010b	CTS9# RTS9# SS9#	—	—	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9	—	—
01100b	SCK12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	CTS12# RTS12# SS12#	—	—	—	—
01101b	—	—	—	RSPCKA	MOSIA	MISOA	—	—
10111b	SSISCK0	SSITXD0	SSIRXD0	AUDIO_MCLK	SSIWS0	—	—	—
11000b	SEG32	SEG31	SEG30	SEG29	SEG28	SEG27	SEG34	SEG33

—: Do not specify this value.



### 19.2.14 PF<sub>n</sub> Pin Function Control Registers (PF<sub>n</sub>PFS) (n = 6, 7)

Address(es): PF6PFS 0008 C1BEh, PF7PFS 0008 C1BFh



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, refer to the table below.	R/W
b7 to b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 19.20 Register Settings for I/O Pin Functions in 100-Pin**

PSEL[4:0] Settings	Pin	
	PF6	PF7
00000b (initial value)	Hi-Z	
00001b	MTIOC3C	MTIOC3A
11000b	SEG26	SEG25

### 19.2.15 PJn Pin Function Control Registers (PJnPFs) (n = 0, 2, 3, 6, 7)

Address(es): PJ0PFS 0008 C1D0h, PJ2PFS 0008 C1D2h, PJ3.PFS 0008 C1D3h, PJ6PFS 0008 C1D6h, PJ7PFS 0008 C1D7h



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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	PJn Pin Input/Output Function	These bits select the peripheral function. For individual pin functions, refer to the table below.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Function Select	0: Used as a pin other than an analog pin 1: Used as an analog pin PJ0: DA0 (100 pins, 64 pins) PJ2: DA1 (100 pins, 64 pins)	R/W
			0: The AVCC0 pin is selected as the reference power supply pin for high-electric potential. 1: The VREFH0 pin is selected as the reference power supply pin for high-electric potential. PJ6: AVCC0/VREFH0 (100 pins, 64 pins)	
			0: The AVSS0 pin is selected as the reference power supply ground pin for low-electric potential. 1: The VREFL0 pin is selected as the reference power supply ground pin for low-electric potential. PJ7: AVSS0/VREFL0 (100 pins, 64 pins)	

**Table 19.21 Register Settings for I/O Pin Functions in 100-Pin**

PSEL[4:0] Settings	Pin PJ3
00000b (initial value)	Hi-Z
00001b	MTIOC3C
01011b	CTS6# RTS6# SS6#
11001b	TS3

## 19.3 Usage Notes

### 19.3.1 Procedure for Specifying I/O Pin Functions

Use the following procedure to specify the I/O pin functions.

1. Set the port mode register (PMR) to 0 to select the general I/O port function.
2. Set the I/O register in the peripheral module to set the I/O signal assigned to the target pin.
3. Set the write-protect register (PWPR) to enable writing to the Pmn pin function control register (PmnPFS) (m = 0 to 5, A to F, J, n = 0 to 7).
4. Specify the input/output function for the pin through the PSEL[4:0] bits settings in the PmnPFS register.
5. Set the PWPR.PFSWE bit to 0 to disable writing to the PmnPFS register.
6. Set the PMR register to 1 as necessary to switch to the selected input/output function for the pin.

### 19.3.2 Notes on MPC Register Setting

1. Only set the Pmn pin function control register (PmnPFS) while the PMR register for the target pin is cleared to 0. If the PmnPFS 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 PmnPFS register. If a value that is not allowed for the register is specified, correct operation is not guaranteed.
3. Do not assign a single function to multiple pins through the MPC register settings.
4. Analog input functions for the A/D converter are multiplexed with pins of ports 4, 9, and E. If a pin is to be used as an analog input, avoid loss of resolution by setting the given bits of the port mode register (PMR), of the port direction register (PDR), and of the pull-up control register (PCR) to 0, i.e. configuring the pin as a general input, and setting the PmnPFS.ASEL bit to 1.
5. Points to note regarding the port mode register (PMR), port direction register (PDR) and Pmn pin function control register (PmnPFS) settings for pins that have multiplexed pin functions are listed in Table 19.22.

**Table 19.22 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	×	Set the ISEL bit to 1 if these are multiplexed with interrupt inputs.
General output ports	0	1	0	0	×	
Peripheral functions	1	×	0	0/1	Peripheral functions (see Table 19.3 to Table 19.21)	Set the ISEL bit to 1 if these are multiplexed with interrupt inputs.
Interrupt inputs	0	0	0	1	×	
NMI	×	×	×	×*1	×	Register settings are not required.
Analog inputs and outputs	0	0	1	×*1	×	Set these as general input port pins so that the output buffers are turned off.
XCIN	0	0	×	×*1	×	Set these as general input port pins so that the output buffers are turned off.

×: 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. The pin does not function as the IRQn input pin even if the PmnPFS.ISEL bit is set to 1.

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, set the PCR.Bn bit to 0; pulling up is automatically turned off for outputs from peripheral modules other than the RIIC.

### 19.3.3 Notes on Using Analog Functions

When using an analog function, configure the pin as general input by setting the given bits of the port mode register (PMR), of the port direction register (PDR), and of the pull-up control register (PCR) to 0, and then set the ASEL bit in the port mn pin function select register (PmnPFS) to 1.

### 19.3.4 Notes on Using LCD Controller/Driver (LCDC) Function

When using pin functions (COM0 to COM7, SEG00 to SEG39, CAPH, CAPL, and VL1 to VL4) of the LCD controller/driver, configure the pins as general input by setting the given bits of the port mode register (PMR), of the port direction register (PDR), and of the pull-up control register (PCR) to 0, and then set the PmnPFS.PSEL[4:0] bits to select the LCD function before setting the PMR register to 1. Also, when using these pin functions, do not use the pins as IRQ input pins regardless of the ISEL setting of the given bits.

### 19.3.5 Notes on Using Capacitive Touch Sensing Unit (CTSUS) Function

When using pin functions (TS0 to TS11 and TSCAP) of the capacitive touch sensing unit, configure the pins as general input by setting the given bits of the port mode register (PMR), of the port direction register (PDR), and of the pull-up control register (PCR) to 0, and then set the PmnPFS.PSEL[4:0] bits to select the CTSUS function before setting the PMR register to 1. Also, when using these pin functions, do not use the pins as IRQ input pins regardless of the ISEL setting of the given bits.

## 20. Multi-Function Timer Pulse Unit 2 (MTU2a)

In this section, “PCLK” is used to refer to PCLKB.

### 20.1 Overview

This MCU 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 20.1 lists the specifications of the MTU, and Table 20.2 lists the function list. Figure 20.1 shows a block diagram of the MTU.

**Table 20.1 MTU Specifications**

Item	Description
Pulse input/output	16 lines max.
Pulse input	3 lines
Count clocks	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 6-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> <li>• Input capture function (noise filter set function)</li> <li>• Counter clear operation</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	A/D converter start trigger can be generated
Low power consumption function	Module stop state can be set.

**Table 20.2 MTU Functions (1/2)**

Item	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
Count clocks	PCLK/1 PCLK/4 PCLK/16 PCLK/64 MTCLKA MTCLKB MTCLKC MTCLKD	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 MTCLKA MTCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 MTCLKA MTCLKB MTCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 MTCLKA MTCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64
General registers (TGR)	TGRA TGRB TGRE	TGRA TGRB	TGRA TGRB	TGRA TGRB	TGRA TGRB	TGRU TGRV TGRW
General registers/ buffer registers	TGRC TGRD TGRF	—	—	TGRC TGRD	TGRC TGRD	—
I/O pins	MTIOC0A MTIOC0B MTIOC0C MTIOC0D	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D	Input pins MTIC5U MTIC5V MTIC5W
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	Low output	○	○	○	○	○
	High 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	—	—	—	—	—	○
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
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	—

Table 20.2 MTU Functions (2/2)

Item	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
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>
Event link function (output)	—	4 sources <ul style="list-style-type: none"> <li>• Compare match 1A</li> <li>• Compare match 1B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match 2A</li> <li>• Compare match 2B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	6 sources <ul style="list-style-type: none"> <li>• Compare match 3A</li> <li>• Compare match 3B</li> <li>• Compare match 3C</li> <li>• Compare match 3D</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	6 sources <ul style="list-style-type: none"> <li>• Compare match 4A</li> <li>• Compare match 4B</li> <li>• Compare match 4C</li> <li>• Compare match 4D</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	—
Event link function (input)	—	(1) Count start operation (2) Input capture operation (TRGA capture) (3) Count restart operation	(1) Count start operation (2) Input capture operation (TRGA capture) (3) Count restart operation	(1) Count start operation (2) Input capture operation (TRGA capture) (3) Count restart operation	(1) Count start operation (2) Input capture operation (TRGA capture) (3) Count restart operation	—
A/D converter start request delaying function	—	—	—	—	<ul style="list-style-type: none"> <li>• 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</li> </ul>	—
Interrupt skipping function	—	—	—	<ul style="list-style-type: none"> <li>• Skips TGRA compare match interrupts</li> </ul>	<ul style="list-style-type: none"> <li>• Skips TCIV interrupts</li> </ul>	—
Module stop function	MSTPCRA.MSTPA9 <sup>1</sup>					

○: Possible

—: Not possible

Note 1. For details on the module stop function, refer to section 11, Low Power Consumption.

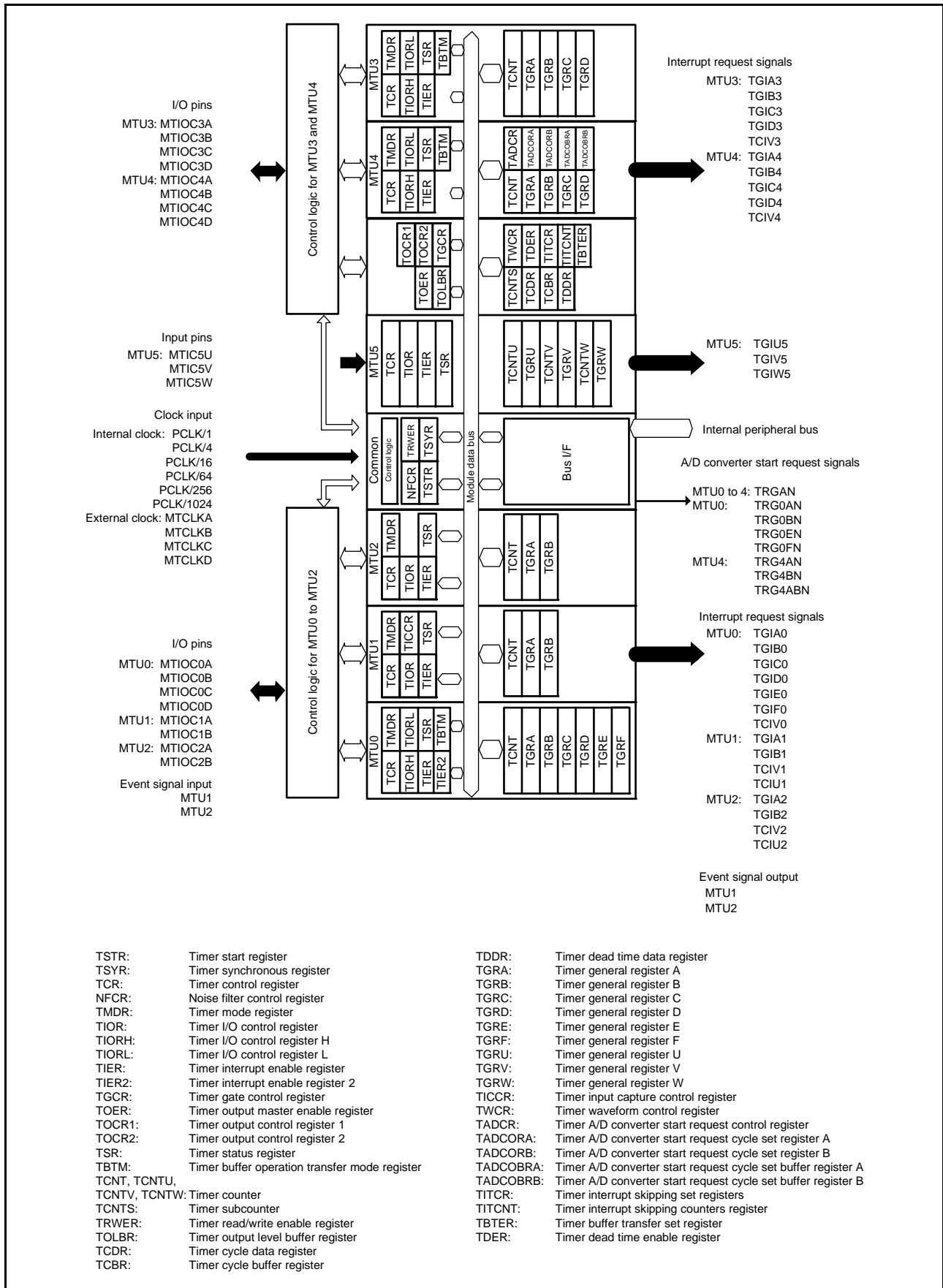


Figure 20.1 MTU Block Diagram



Table 20.3 lists the I/O pins to be used by the MTU.

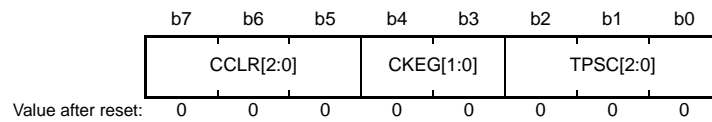
**Table 20.3 MTU I/O Pins**

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

## 20.2 Register Descriptions

### 20.2.1 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 20.6 to Table 20.10.	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 20.4 and Table 20.5.	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.

The TCR register 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 count clock. The clock source can be selected for each channel. See Table 20.6 to Table 20.10 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 clock at both edges = PCLK/2 clock 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 clock or slower. When PCLK/1 clock 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 20.4 and Table 20.5 for details.

**Table 20.4 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.SYNCn bit (n = 0, 3, 4) 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 20.5 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.SYNCn bit (n = 1, 2) to 1.

Note 2. Bit 7 is reserved in MTU1 and MTU2. This bit is read as 0. The write value should be 0.

**Table 20.6 TPSC[2:0] (MTU0)**

Channel	Bit 2	Bit 1	Bit 0	Description
	TPSC2	TPSC1	TPSC0	
MTU0	0	0	0	Internal clock: counts on PCLK/1 clock
	0	0	1	Internal clock: counts on PCLK/4 clock
	0	1	0	Internal clock: counts on PCLK/16 clock
	0	1	1	Internal clock: counts on PCLK/64 clock
	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 20.7 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 clock
	0	0	1	Internal clock: counts on PCLK/4 clock
	0	1	0	Internal clock: counts on PCLK/16 clock
	0	1	1	Internal clock: counts on PCLK/64 clock
	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 clock
	1	1	1	Counts on MTU2.TCNT overflow/underflow

Note: This setting is ignored when MTU1 is in phase counting mode.

**Table 20.8 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 clock
	0	0	1	Internal clock: counts on PCLK/4 clock
	0	1	0	Internal clock: counts on PCLK/16 clock
	0	1	1	Internal clock: counts on PCLK/64 clock
	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 clock

Note: This setting is ignored when MTU2 is in phase counting mode.

**Table 20.9 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 clock
	0	0	1	Internal clock: counts on PCLK/4 clock
	0	1	0	Internal clock: counts on PCLK/16 clock
	0	1	1	Internal clock: counts on PCLK/64 clock
	1	0	0	Internal clock: counts on PCLK/256 clock
	1	0	1	Internal clock: counts on PCLK/1024 clock
	1	1	0	External clock: counts on MTCLKA pin input
	1	1	1	External clock: counts on MTCLKB pin input

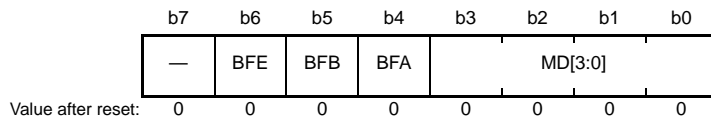
**Table 20.10 TPSC[1:0] (MTU5)**

Channel	Bit 1	Bit 0	Description
	TPSC1	TPSC0	
MTU5	0	0	Internal clock: counts on PCLK/1 clock
	0	1	Internal clock: counts on PCLK/4 clock
	1	0	Internal clock: counts on PCLK/16 clock
	1	1	Internal clock: counts on PCLK/64 clock

Note: Bits 7 to 2 are reserved in MTU5. These bits are read as 0. The write value should be 0.

## 20.2.2 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



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	These bits specify the timer operating mode. See Table 20.11 for details.	R/W
b4	BFA	Buffer Operation A	0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation	R/W
b5	BFB	Buffer Operation B	0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation	R/W
b6	BFE	Buffer Operation E	0: MTU0.TGRE and MTU0.TGRF operate normally 1: MTU0.TGRE and MTU0.TGRF used together for buffer operation	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The TMDR register specifies the operating mode of each channel. TMDR values should be specified only while TCNT operation is stopped.

**Table 20.11 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. 0 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 MTU4.TIER.TGIEC bit 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. See Figure 20.40 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 MTU3.TIER.TGIED or MTU4.TIER.TGIED bit 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. See Figure 20.40 for an illustration of the Tb interval in complementary PWM mode.

**BFE Bit (Buffer Operation E)**

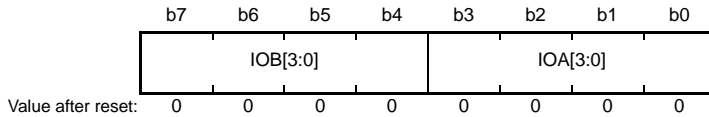
This bit specifies normal operation or buffered operation for MTU0.TGRE and MTU0.TGRF. Compare match with TGRF occurs even when TGRF is used as a buffer register.

In MTU1 to MTU4, this bit is reserved. It is read as 0. The write value should be 0.

### 20.2.3 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

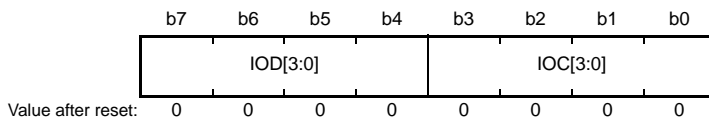


Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	I/O Control A	See the following tables.*1 MTU0.TIORH: Table 20.20 MTU1.TIOR: Table 20.22 MTU2.TIOR: Table 20.23 MTU3.TIORH: Table 20.24 MTU4.TIORH: Table 20.26	R/W
b7 to b4	IOB[3:0]	I/O Control B	See the following tables.*1 MTU0.TIORH: Table 20.12 MTU1.TIOR: Table 20.14 MTU2.TIOR: Table 20.15 MTU3.TIORH: Table 20.16 MTU4.TIORH: Table 20.18	R/W

Note 1. If the IOn[3:0] (n = A, B) bits are changed to an “output prohibited” setting (0000b or 0100b) while output of the low or high level 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

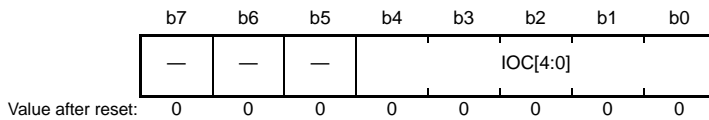


Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	I/O Control C	See the following tables.*1 MTU0.TIORL: Table 20.21 MTU3.TIORL: Table 20.25 MTU4.TIORL: Table 20.27	R/W
b7 to b4	IOD[3:0]	I/O Control D	See the following tables.*1 MTU0.TIORL: Table 20.13 MTU3.TIORL: Table 20.17 MTU4.TIORL: Table 20.19	R/W

Note 1. If the IOn[3:0] (n = C, D) bits are changed to an “output prohibited” setting (0000b or 0100b) while output of the low or high level 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 20.28	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/TIORV/TIORW) 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 TSTR.CST bit is set to 0). Note also that, in PWM mode 2, the output at the point at which the counter is set 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 20.12 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 low. Low output at compare match.
0	0	1	0	Initial output is low. High output at compare match.
0	0	1	1	Initial output is low. Toggle output at compare match.
0	1	0	0	Output prohibited
0	1	0	1	Initial output is high. Low output at compare match.
0	1	1	0	Initial output is high. High output at compare match.
0	1	1	1	Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register Input capture at rising edge.
1	0	0	1	Input capture at falling edge.
1	0	1	x	Input capture at both edges.
1	1	x	x	Capture input source is count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care



Table 20.13 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is 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 20.14 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Input capture at generation of MTU0.TGRC compare match/input capture.

x: Don't care

Table 20.15 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 20.16 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 20.17 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU3.TMDR.BFB bit 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 20.18 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 20.19 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU4.TMDR.BFB bit 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 20.20 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Table 20.21 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Note 1. When the MTU0.TMDR.BFA bit 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 20.22 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Input capture at generation of MTU0.TGRA compare match/input capture.

x: Don't care

Table 20.23 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 20.24 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 20.25 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU3.TMDR.BFA bit 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 20.26 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 20.27 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 low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU4.TMDR.BFA bit 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.



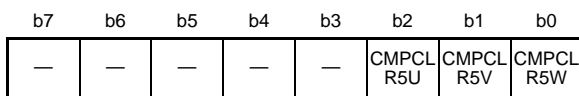
**Table 20.28** TIORU, TIORV, and TIORW (MTU5)

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	Setting prohibited
1	0	0	0	1		Input capture at rising edge.
1	0	0	1	0		Input capture at falling edge.
1	0	0	1	1		Input capture at both edges.
1	0	1	x	x		Setting prohibited
1	1	0	0	0		Setting prohibited
1	1	0	0	1		Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	0	1	0		Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	0	1	1		Measurement of low pulse width of external input signal. Capture at trough in 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 trough in complementary PWM mode.
1	1	1	1	1		Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode.

x: Don't care

### 20.2.4 Timer Compare Match Clear Register (TCNTCMPCLR)

Address(es): MTU5.TCNTCMPCLR 0008 88B6h



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

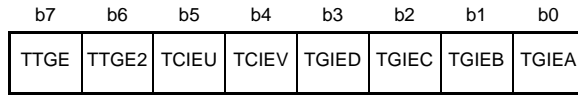
Bit	Symbol	Bit Name	Description	R/W
b0	CMPCLR5W	TCNT Compare Clear 5W	0: Disables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture 1: Enables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture	R/W
b1	CMPCLR5V	TCNT Compare Clear 5V	0: Disables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture 1: Enables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture	R/W
b2	CMPCLR5U	TCNT Compare Clear 5U	0: Disables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture 1: Enables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TCNTCMPCLR register specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW.

## 20.2.5 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



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.

The TIER register 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, 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, 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.TIER2 0008 8724h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	TGIEF	TGIEE

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

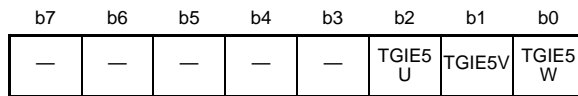
Bit	Symbol	Bit Name	Description	R/W
b0	TGIEE	TGR Interrupt Enable E	0: Interrupt requests (TGIE) disabled 1: Interrupt requests (TGIE) enabled	R/W
b1	TGIEF	TGR Interrupt Enable F	0: Interrupt requests (TGIF) disabled 1: Interrupt requests (TGIF) enabled	R/W
b7 to b2	—	Reserved	These bits are read as 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.TGRn (n = E, 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 5n)**

Each bit enables or disables interrupt requests (TGI5n) (n = W, V, U).

## 20.2.6 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

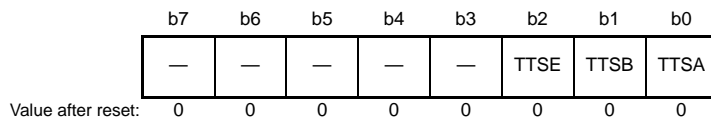
The MTU has a total of five TSR registers, one each for MTU0 to MTU4. The TSR register indicates the status of each channel.

### 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.

## 20.2.7 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.

The TBTM register 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)

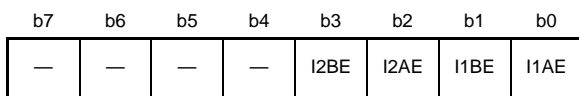
This bit specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSB bit in the channel to 1.

### 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 read as 0. The write value should be 0. When MTU0 is not set to PWM mode, do not set the TTSE bit to 1.

### 20.2.8 Timer Input Capture Control Register (TICCR)

Address(es): MTU1.TICCR 0008 8790h



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

Bit	Symbol	Bit Name	Description	R/W
b0	I1AE	Input Capture Enable	0: Does not include the MTIOC1A pin in the MTU2.TGRA input capture conditions 1: Includes the MTIOC1A pin in the MTU2.TGRA input capture conditions	R/W
b1	I1BE	Input Capture Enable	0: Does not include the 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.

The TICCR register specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded.



## 20.2.9 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

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/TADCOBRB Transfer Timing Select	See Table 20.29 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 TITCR.T3AEN and T4VEN bits are cleared to 0 or the interrupt skipping count setting bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), do not link A/D converter start requests with interrupt skipping operation (clear the TADCR.ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits 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.

The TADCR register enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping operation.

**Table 20.29 Setting of Transfer Timing by BF[1:0] Bits**

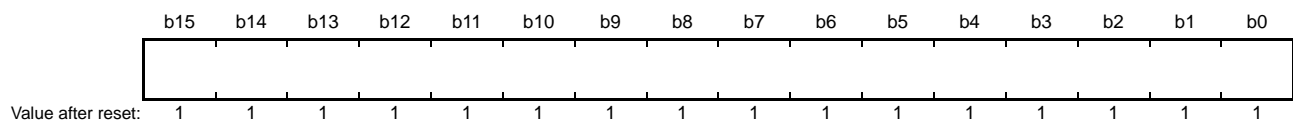
Bit 15	Bit 14	Description
BF1	BF0	
0	0	Does not transfer data from the cycle set buffer register to the cycle set register.
0	1	Transfers data from the cycle set buffer register to the cycle set register at the crest of the MTU4.TCNT count.*1
1	0	Transfers data from the cycle set buffer register to the 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 operation mode.

Note 2. These settings are prohibited when complementary PWM mode is not selected.

### 20.2.10 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

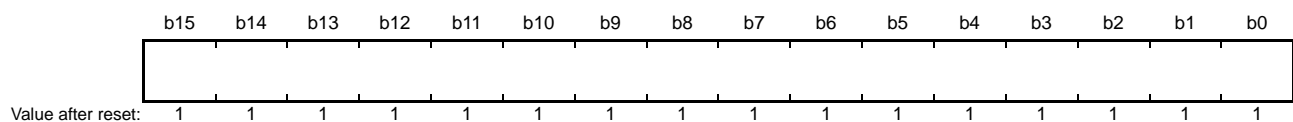


Note: MTU4.TADCORA and MTU4.TADCORB must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TADCOBRA and TADCOBRB registers specify the A/D converter start request cycle. When the MTU4.TCNT count reaches the value in TADCORA or TADCORB, a corresponding A/D converter start request will be issued.

### 20.2.11 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

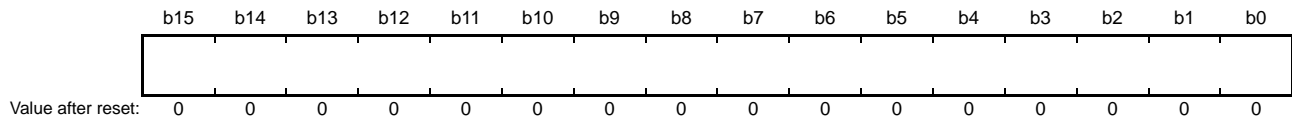


Note: MTU4.TADCORA and MTU4.TADCORB must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TADCOBRA and TADCOBRB registers function as buffer registers for TADCORA and TADCORB, respectively. These registers specify the A/D converter start request cycle. When the crest or trough of the MTU4.TCNT count is reached, these register values are transferred to TADCORA and TADCORB, respectively.

### 20.2.12 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

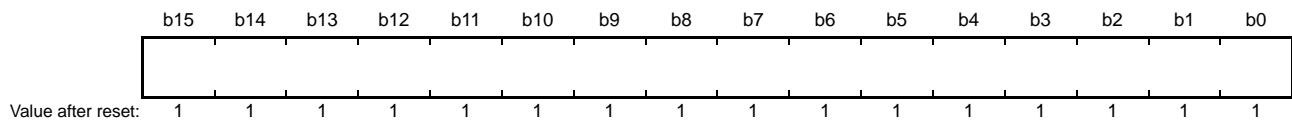


Note: The TCNT counters must not be accessed in 8-bit units; they should be accessed in 16-bit units.

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 readable/writable counter.

### 20.2.13 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: The TGR registers must not be accessed in 8-bit units; they should 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.

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.

### 20.2.14 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

The TSTR registers start or stop 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.

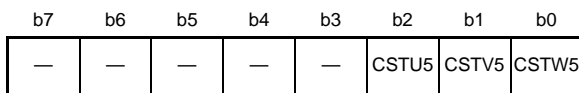
#### CSTn Bits (Counter Start n) (n = 0 to 4)

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



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

## 20.2.15 Timer Synchronous Registers (TSYR)

Address(es): MTU.TSYR 0008 8681h

b7	b6	b5	b4	b3	b2	b1	b0
SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC0	Timer Synchronous Operation 0	0: MTU0.TCNT operates independently (TCNT setting/clearing is not related to other channels) 1: MTU0.TCNT performs synchronous operation. TCNT synchronous setting/synchronous clearing is enabled	R/W
b1	SYNC1	Timer Synchronous Operation 1	0: MTU1.TCNT operates independently (TCNT setting/clearing is not related to other channels) 1: MTU1.TCNT performs synchronous operation. TCNT synchronous setting/synchronous clearing is enabled	R/W
b2	SYNC2	Timer Synchronous Operation 2	0: MTU2.TCNT operates independently (TCNT setting/clearing is not related to other channels) 1: MTU2.TCNT performs synchronous operation. TCNT synchronous setting/synchronous clearing is enabled	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SYNC3	Timer Synchronous Operation 3	0: MTU3.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation. TCNT synchronous setting/synchronous clearing is enabled.	R/W
b7	SYNC4	Timer Synchronous Operation 4	0: MTU4.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation. TCNT synchronous setting/synchronous clearing is enabled.	R/W

The TSYR registers select 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)

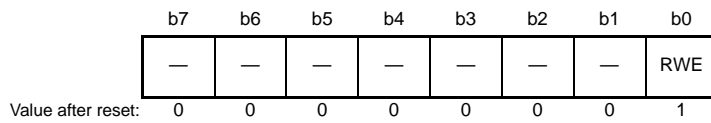
Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous setting 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 the TCR.CCLR[2:0] bits.

## 20.2.16 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

The TRWER registers enable or disable access to the registers and counters that have write-protection capability against accidental modification in MTU3 and MTU4.

### RWE Bit (Read/Write Enable)

This bit enables or disables access to the registers that have write-protection capability against accidental modification.  
[Clearing condition]

- When 0 is written to the RWE bit after reading the RWE bit = 1

- Registers and Counters having Write-Protection Capability against Accidental Modification

22 registers: MTUn.TCR, MTUn.TMDR, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, TOER, TOCR1, TOCR2, TGCR, TCDR, TDDR, and MTUn.TCNT (n = 3, 4)

## 20.2.17 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*1 1: MTU output is enabled	R/W
b1	OE4A	Master Enable MTIOC4A	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b2	OE4B	Master Enable MTIOC4B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b3	OE3D	Master Enable MTIOC3D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b4	OE4C	Master Enable MTIOC4C	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b5	OE4D	Master Enable MTIOC4D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. To output a non-active level from each pin when MTU output is disabled, make necessary settings for non-active level output from general I/O ports in the data direction registers (PDR), port output data registers (PODR), and port mode register (PMR) in advance. For details, refer to I/O Ports section.

The TOER registers enable or disable output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

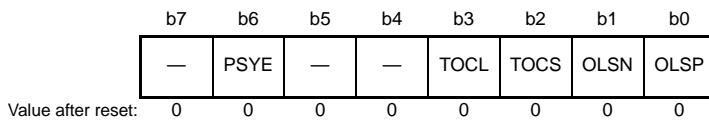
These pins do not output correctly if the TOER bits have not been set. In MTU3 and MTU4, set TOER prior to setting TIOR.

Set TOER after clearing the TSTR.CST3 and CST4 bits to 0 (see Figure 20.35 and Figure 20.38).



## 20.2.18 Timer Output Control Registers 1 (TOCR1)

Address(es): MTU.TOCR 0008 860Eh



Bit	Symbol	Bit Name	Description	R/W
b0	OLSP	Output Level Select P*2,*3	See Table 20.30.	R/W
b1	OLSN	Output Level Select N*2,*3	See Table 20.31.	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.

The TOCR1 registers 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 20.30 Output Level Select Function**

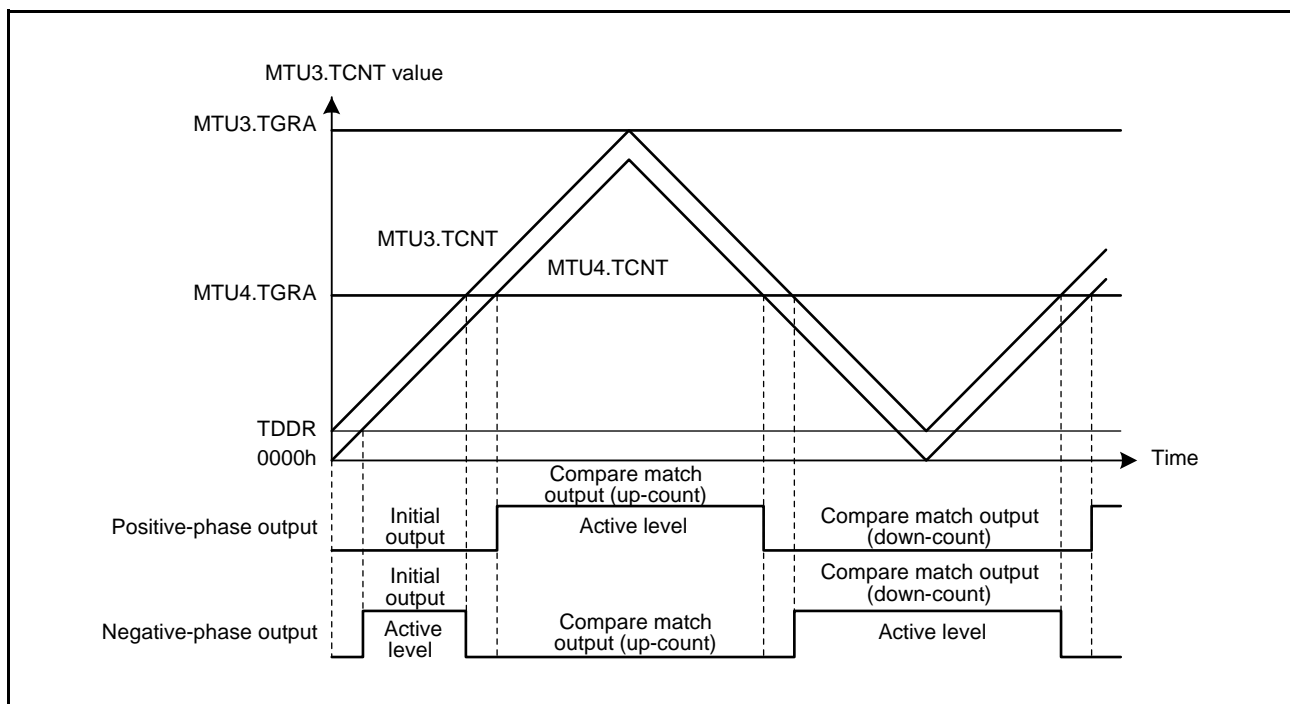
Bit 0	Function			
OLSP	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High	Low	Low	High
1	Low	High	High	Low

**Table 20.31 Output Level Select Function**

Bit 1	Function			
OLSN	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High	Low	High	Low
1	Low	High	Low	High

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

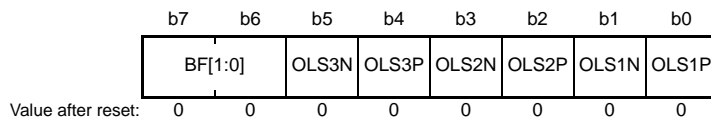
Figure 20.2 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.



**Figure 20.2 Example of Output in Complementary PWM Mode**

## 20.2.19 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 <sup>*1, *2</sup>	This bit selects the output level on MTIOC3B in reset-synchronized PWM mode and complementary PWM mode. See Table 20.32.	R/W
b1	OLS1N	Output Level Select 1N <sup>*1, *2</sup>	This bit selects the output level on MTIOC3D in reset-synchronized PWM mode and complementary PWM mode. See Table 20.33.	R/W
b2	OLS2P	Output Level Select 2P <sup>*1, *2</sup>	This bit selects the output level on MTIOC4A in reset-synchronized PWM mode and complementary PWM mode. See Table 20.34.	R/W
b3	OLS2N	Output Level Select 2N <sup>*1, *2</sup>	This bit selects the output level on MTIOC4C in reset-synchronized PWM mode and complementary PWM mode. See Table 20.35.	R/W
b4	OLS3P	Output Level Select 3P <sup>*1, *2</sup>	This bit selects the output level on MTIOC4B in reset-synchronized PWM mode and complementary PWM mode. See Table 20.36.	R/W
b5	OLS3N	Output Level Select 3N <sup>*1, *2</sup>	This bit selects the output level on MTIOC4D in reset-synchronized PWM mode and complementary PWM mode. See Table 20.37.	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 20.38 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).

The TOCR2 registers control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

**Table 20.32 MTIOC3B Output Level Select Function**

Bit 0	Function			
OLS1P	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High	Low	Low	High
1	Low	High	High	Low

**Table 20.33 MTIOC3D Output Level Select Function**

Bit 1	Function			
OLS1N	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High	Low	High	Low
1	Low	High	Low	High

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

**Table 20.34 MTIOC4A Output Level Select Function**

Bit 2	Function			
OLS2P	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High	Low	Low	High
1	Low	High	High	Low

**Table 20.35 MTIOC4C Output Level Select Function**

Bit 3	Function			
OLS2N	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High	Low	High	Low
1	Low	High	Low	High

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

**Table 20.36 MTIOC4B Output Level Select Function**

Bit 4	Function			
OLS3P	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High	Low	Low	High
1	Low	High	High	Low

**Table 20.37 MTIOC4D Output Level Select Function**

Bit 5	Function			
OLS3N	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High	Low	High	Low
1	Low	High	Low	High

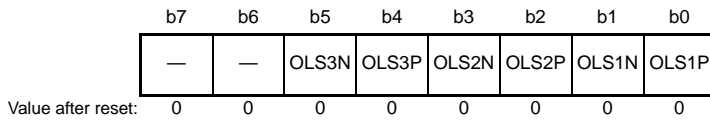
Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

**Table 20.38 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 TOLBR to TOCR2.	Does not transfer data from TOLBR to TOCR2.
0	1	Transfers data from TOLBR to TOCR2 at the crest of the MTU4.TCNT count.	Transfers data from TOLBR to TOCR2 when MTU4.TCNT or MTU3.TCNT is cleared.
1	0	Transfers data from TOLBR to TOCR2 at the trough of the MTU4.TCNT count.	Setting prohibited
1	1	Transfers data from TOLBR to TOCR2 at the crest and trough of the MTU4.TCNT count.	Setting prohibited

### 20.2.20 Timer Output Level Buffer Registers (TOLBR)

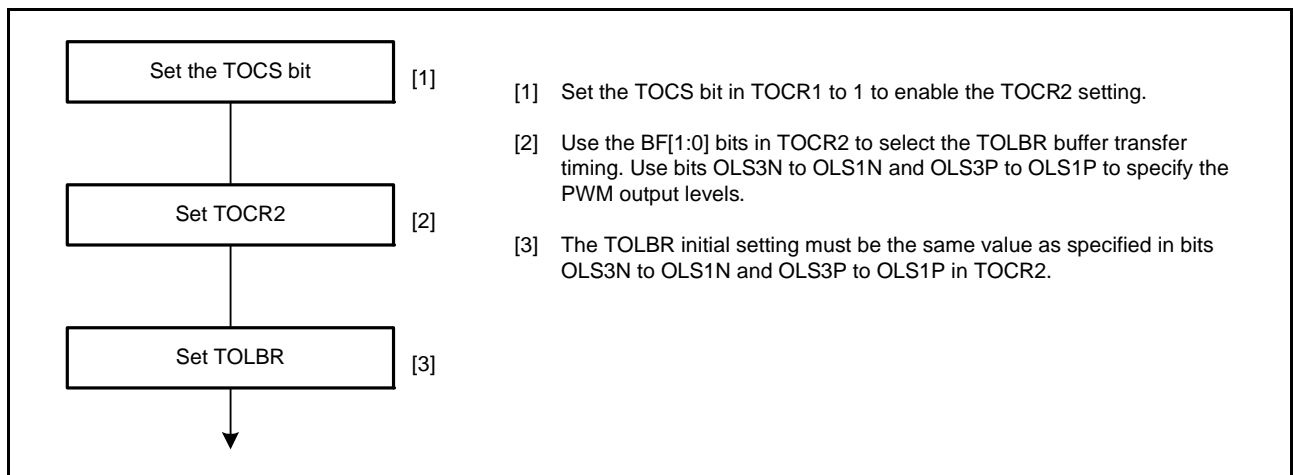
Address(es): MTU.TOLBR 0008 8636h



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

The TOLBR registers function as buffer registers for TOCR2 and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 20.3 shows an example of the PWM output level setting procedure in buffer operation.



**Figure 20.3 Example of PWM Output Level Setting Procedure in Buffer Operation**

## 20.2.21 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 is 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 20.39.	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

The TGCR registers 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 20.39.

### 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)

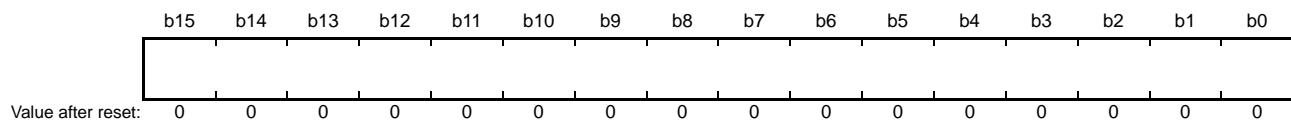
This bit selects whether to make the functions of TGCR effective or ineffective.

**Table 20.39 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

### 20.2.22 Timer Subcounters (TCNTS)

Address(es): MTU.TCNTS 0008 8620h

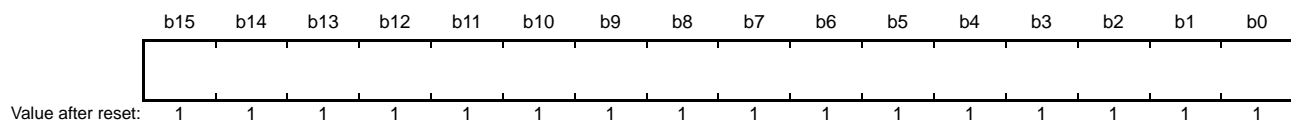


Note: The TCNTS counters must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TCNTS counters are read-only counters that are used only in complementary PWM mode.

### 20.2.23 Timer Dead Time Data Registers (TDDR)

Address(es): MTU.TDDR 0008 8616h



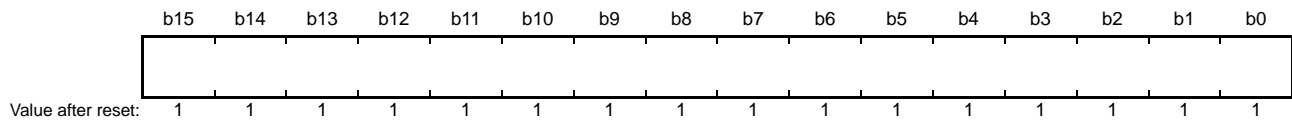
Note: The TDDR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TDDR registers specify the MTU3.TCNT and MTU4.TCNT counter offset value in complementary PWM mode. 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.



### 20.2.24 Timer Cycle Data Registers (TCDR)

Address(es): MTU.TCDR 0008 8614h



Note: The TCDR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TCDR registers specify the count value to switch the count direction of the TCNTS counter. These registers are used only in complementary PWM mode. Set half the PWM 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).

### 20.2.25 Timer Cycle Buffer Registers (TCBR)

Address(es): MTU.TCBR 0008 8622h

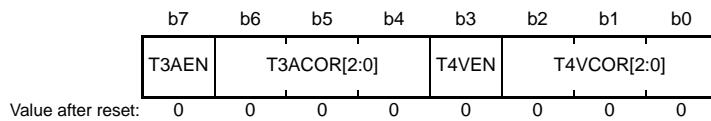


Note: The TCBR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TCBR registers function as buffer registers for TCDR, and specify the count value to switch the count direction of the TCNTS counter. These registers are used only in complementary PWM mode. The TCBR value is transferred to TCDR with the transfer timing set in TMDR.

## 20.2.26 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.* <sup>1</sup> For details, see Table 20.40.	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.* <sup>1</sup> For details, see Table 20.41.	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 TITCNT counter.

**Table 20.40 Setting of Interrupt Skipping Count by T4VCOR[2:0] Bits**

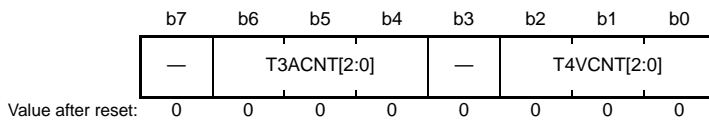
Bit 2	Bit 1	Bit 0	Description
T4VCOR2	T4VCOR1	T4VCOR0	
0	0	0	Does not perform TCIV4 interrupt skipping.
0	0	1	Sets the TCIV4 interrupt skipping count to 1.
0	1	0	Sets the TCIV4 interrupt skipping count to 2.
0	1	1	Sets the TCIV4 interrupt skipping count to 3.
1	0	0	Sets the TCIV4 interrupt skipping count 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 20.41 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits**

Bit 6	Bit 5	Bit 4	Description
T3ACOR2	T3ACOR1	T3ACOR0	
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.

## 20.2.27 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 source 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 source 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.

The TITCNT counters count the number of interrupt source occurrences for interrupt skipping. 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 T4VCOR[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

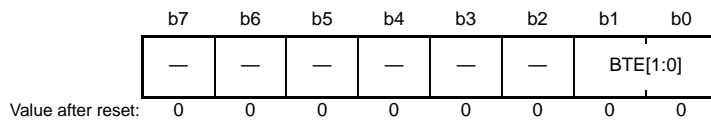
### T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)

[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 T3ACOR[2:0] bits in TITCR are cleared to 000b

## 20.2.28 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 20.42 for details.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TBTER registers 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 20.42 Setting of TBTER.BTE[1:0] Bits**

Bit 1	Bit 0	Description
BTE1	BTE0	
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: Target buffer registers: MTU3.TGRC, MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, and MTU.TCBRA

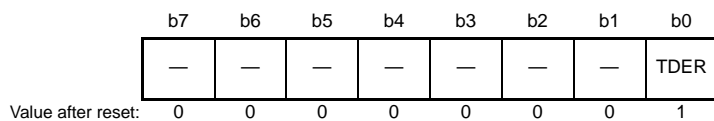
Note 1. Data is transferred in accordance with the TMDT.MD[3:0] bit setting. For details, refer to section 20.3.8, Complementary PWM Mode.

Note 2. When interrupt skipping is disabled (the TITCR.T3AEN and T4VEN bits or the interrupt skipping count setting bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), be sure to disable link of buffer transfer with interrupt skipping (clear the TBTER.BTE1 bit to 0).

If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

### 20.2.29 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.

The TDER registers specify 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)

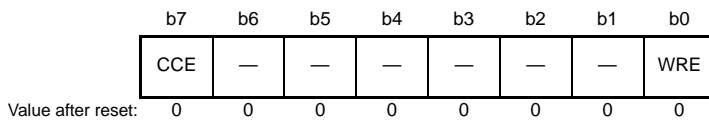
This bit specifies whether to generate dead time.

[Clearing condition]

- When 0 is written to the TDER bit after reading the TDER bit = 1

### 20.2.30 Timer Waveform Control Registers (TWCR)

Address(es): MTU.TWCR 0008 8660h



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.

The TWCR registers control 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 TWCR.CCE bit and TWCR.WRE bit 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 20.40.

[Setting condition]

- When 1 is written to the WRE bit after reading the WRE bit = 0

#### CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at TGRA3 compare match in complementary PWM mode 1.

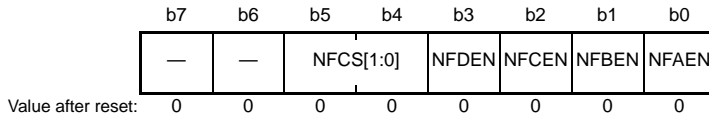
[Setting condition]

- When 1 is written to the CCE bit after reading the CCE bit = 0

### 20.2.31 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	0: The noise filter for the MTIOCnA pin is disabled 1: The noise filter for the MTIOCnA pin is enabled	R/W
b1	NFBEN	Noise Filter B Enable	0: The noise filter for the MTIOCnB pin is disabled 1: The noise filter for the MTIOCnB pin is enabled	R/W
b2	NFCEN	Noise Filter C Enable	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	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	b5 b4 0 0: PCLK/1 0 1: PCLK/8 1 0: PCLK/32 1 1: 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 NFCR for MTU1 and MTU2. These bits are read as 0, and writing to them is not possible.

The MTUn.NFCR registers (n = 0 to 4) enable and disable 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.

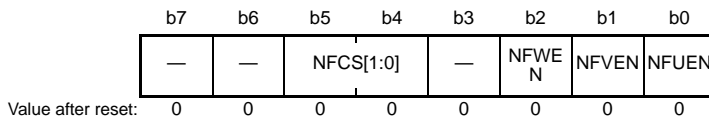


**NFCS[1:0] Bits (Noise Filter Clock Select)**

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	b5 b4 0 0: PCLK/1 0 1: PCLK/8 1 0: PCLK/32 1 1: 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.

### 20.2.32 Bus Master Interface

The timer counters (TCNT), timer 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/TADCORB), and timer A/D converter start request cycle set buffer registers (TADCORA/TADCORB) 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. 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.

## 20.3 Operation

### 20.3.1 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 20.4 shows an example of the count operation setting procedure.

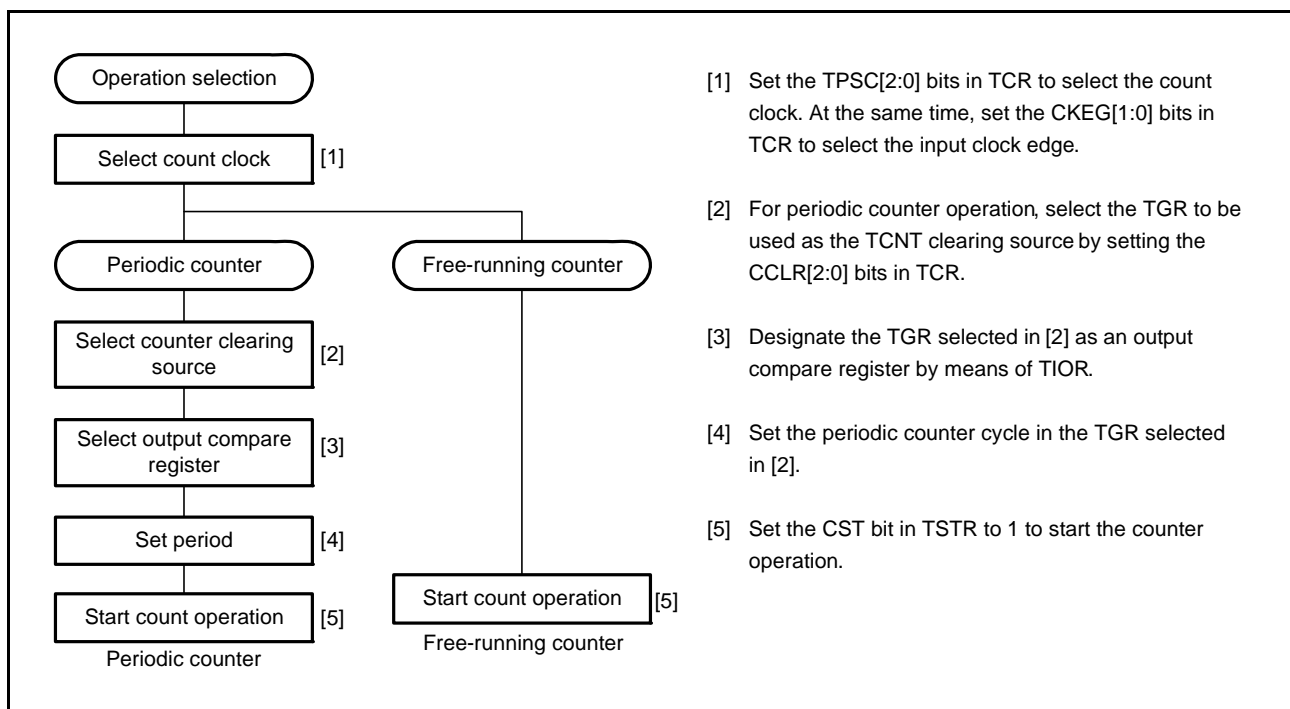
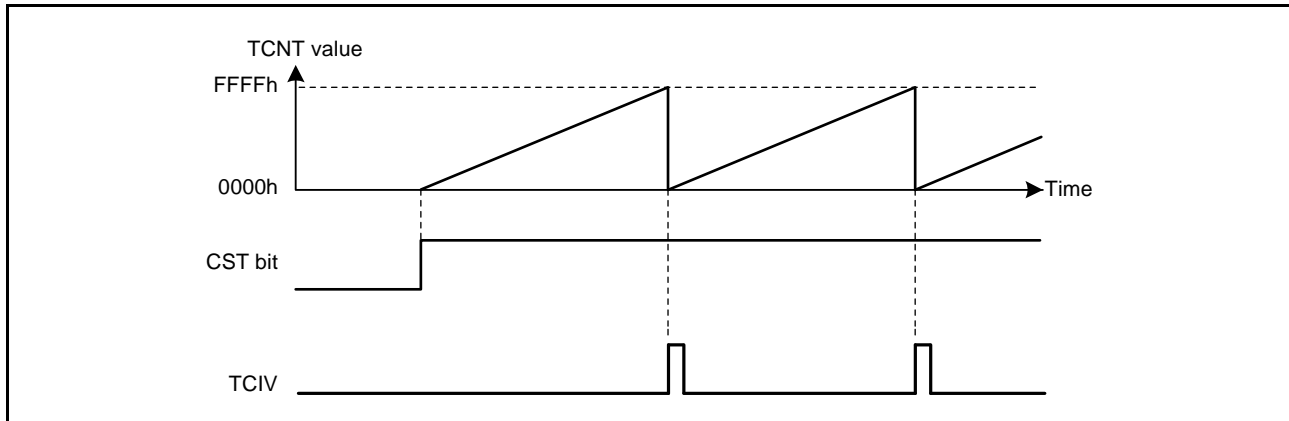


Figure 20.4 Example of Counter Operation Setting Procedure

### (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 CSTn 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 20.5 illustrates free-running counter operation.

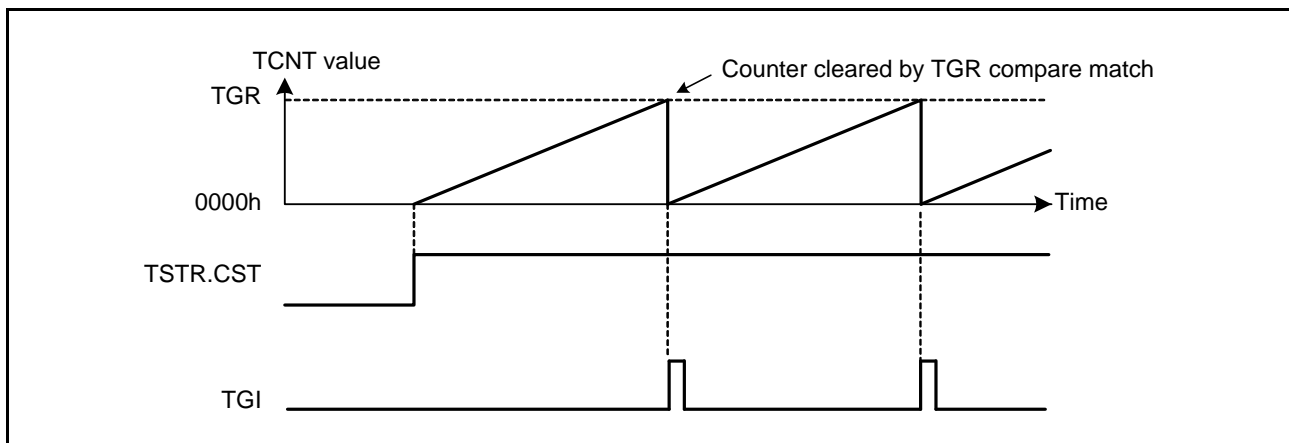


**Figure 20.5** Free-Running Counter Operation

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 the TCR.CCLR[2:0] bits. 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 set to 0000h.

If the value of the corresponding TIER.TGIE bit is 1 at this point, the MTU requests an interrupt. After a compare match, TCNT starts counting up again from 0000h.

Figure 20.6 illustrates periodic counter operation.



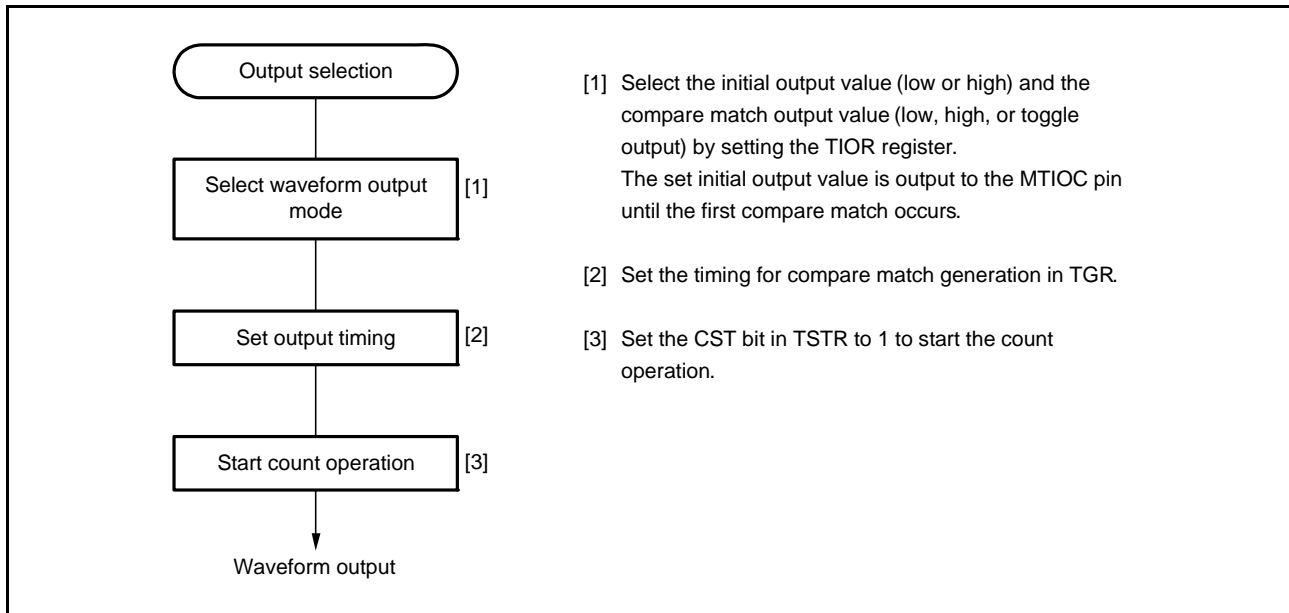
**Figure 20.6** Periodic Counter Operation

## (2) Waveform Output by Compare Match

The MTU can output low or high or toggle output from the corresponding output pin using compare match.

### (a) Example of Procedure for Setting Waveform Output by Compare Match

Figure 20.7 shows an example of the procedure for setting waveform output by compare match.



**Figure 20.7** Example of Procedure for Setting Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 20.8 shows an example of low output and high output.

In this example, TCNT has been designated as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the pin level is the same as the specified level, the pin level does not change.

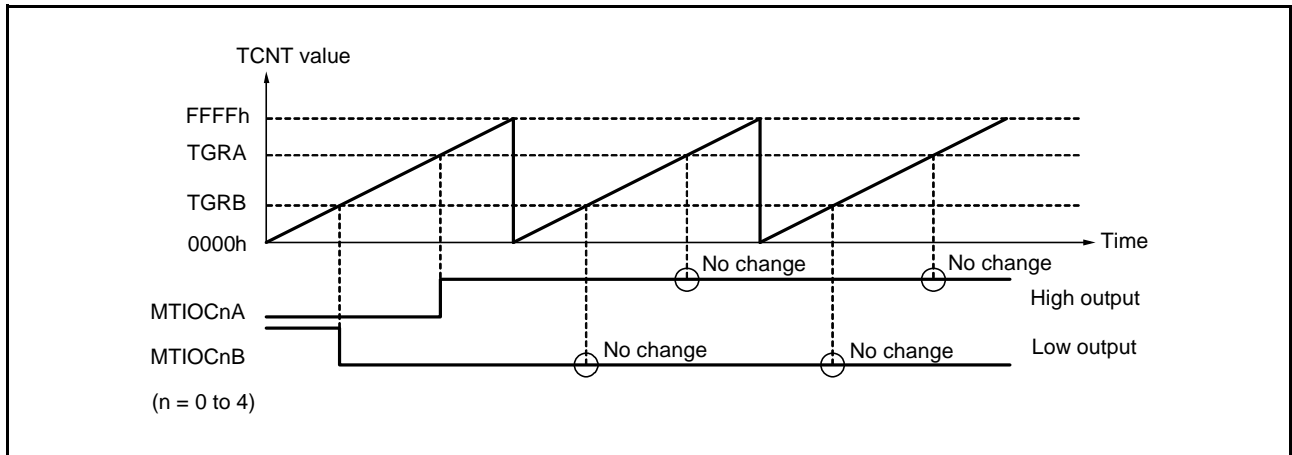


Figure 20.8 Example of Low Output and High Output Operation

Figure 20.9 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.

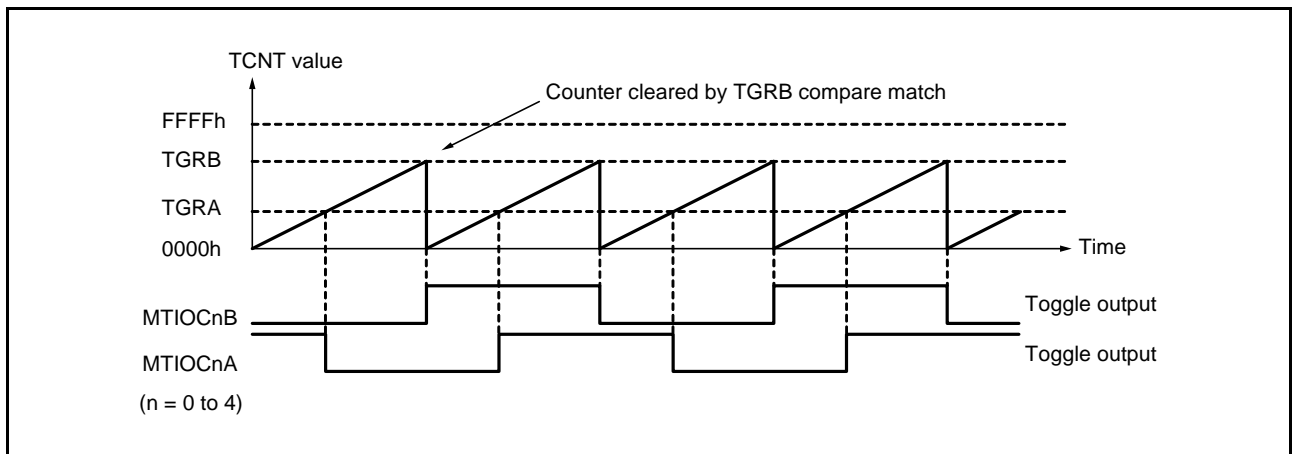


Figure 20.9 Example of Toggle Output Operation

### (3) Input Capture Function

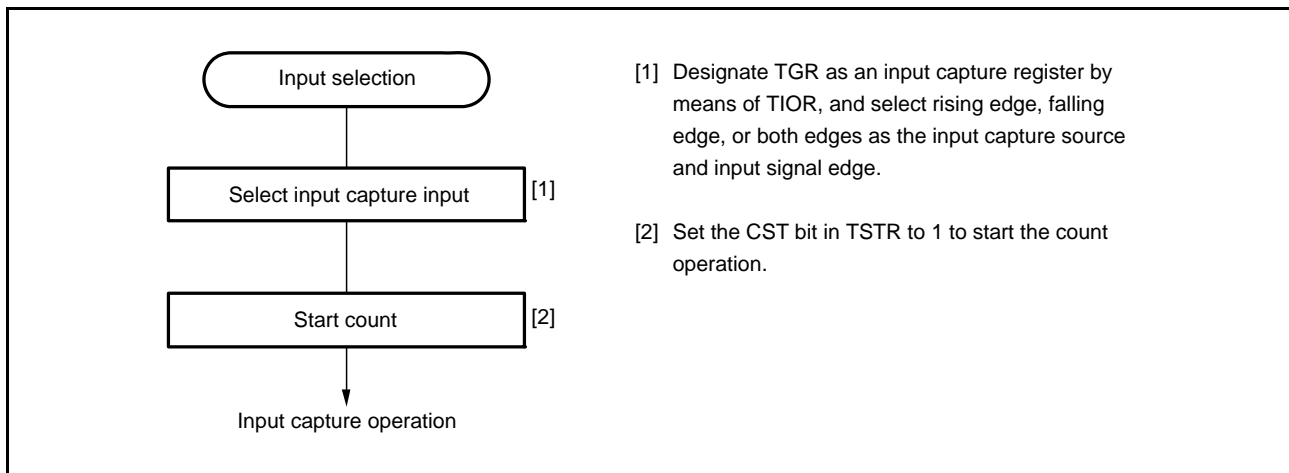
The TCNT value can be transferred to TGR on detection of the input edge of the MTIOC<sub>n</sub>m (n = 0 to 4; m = A to D) pin and MTIC5<sub>m</sub> (m = W, V, U) pin.

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 clock should not be selected as the counter input clock used for input capture input. Input capture will not be generated if PCLK/1 clock is selected.

#### (a) Example of Input Capture Operation Setting Procedure

Figure 20.10 shows an example of the input capture operation setting procedure.



**Figure 20.10** Example of Input Capture Operation Setting Procedure



(b) Example of Input Capture Operation

Figure 20.11 shows an example of input capture operation.

In this example, both rising and falling edges have been selected as the MTIOCnA pin input capture input edge, the falling edge has been selected as the MTIOCnB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

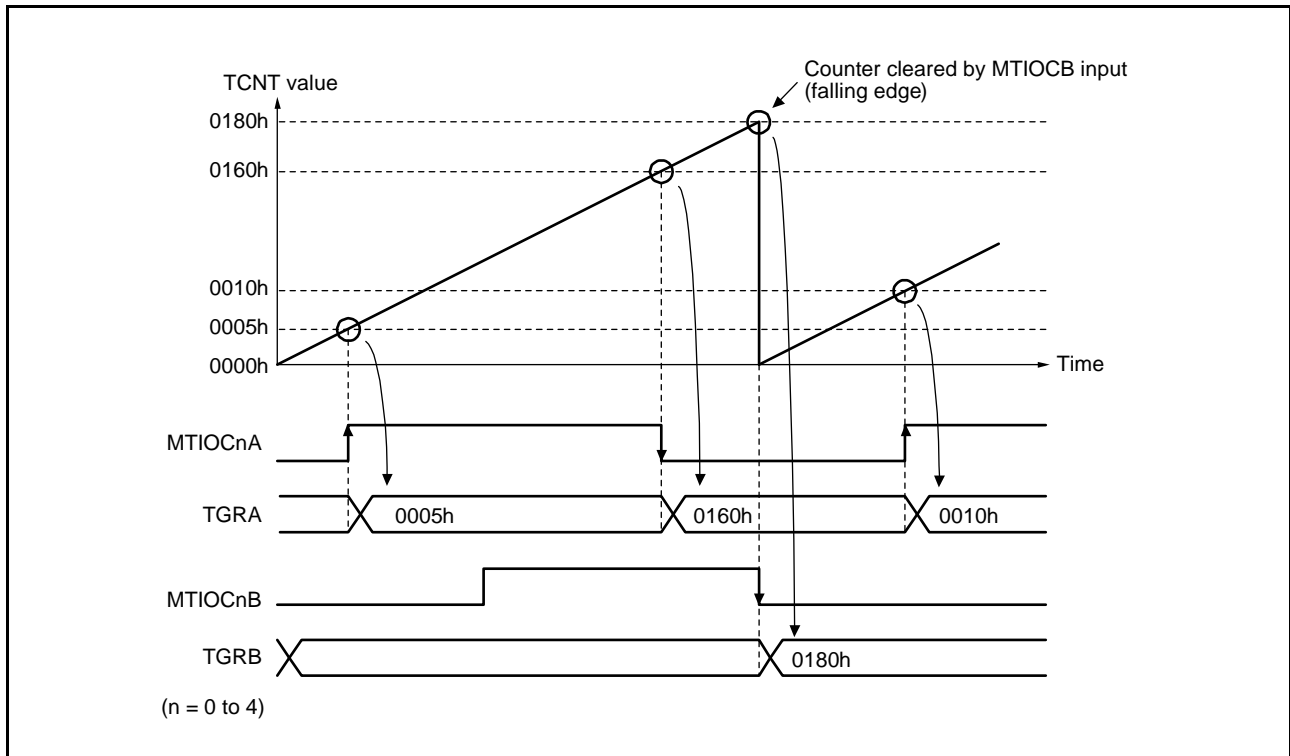


Figure 20.11 Example of Input Capture Operation

### 20.3.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous setting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.

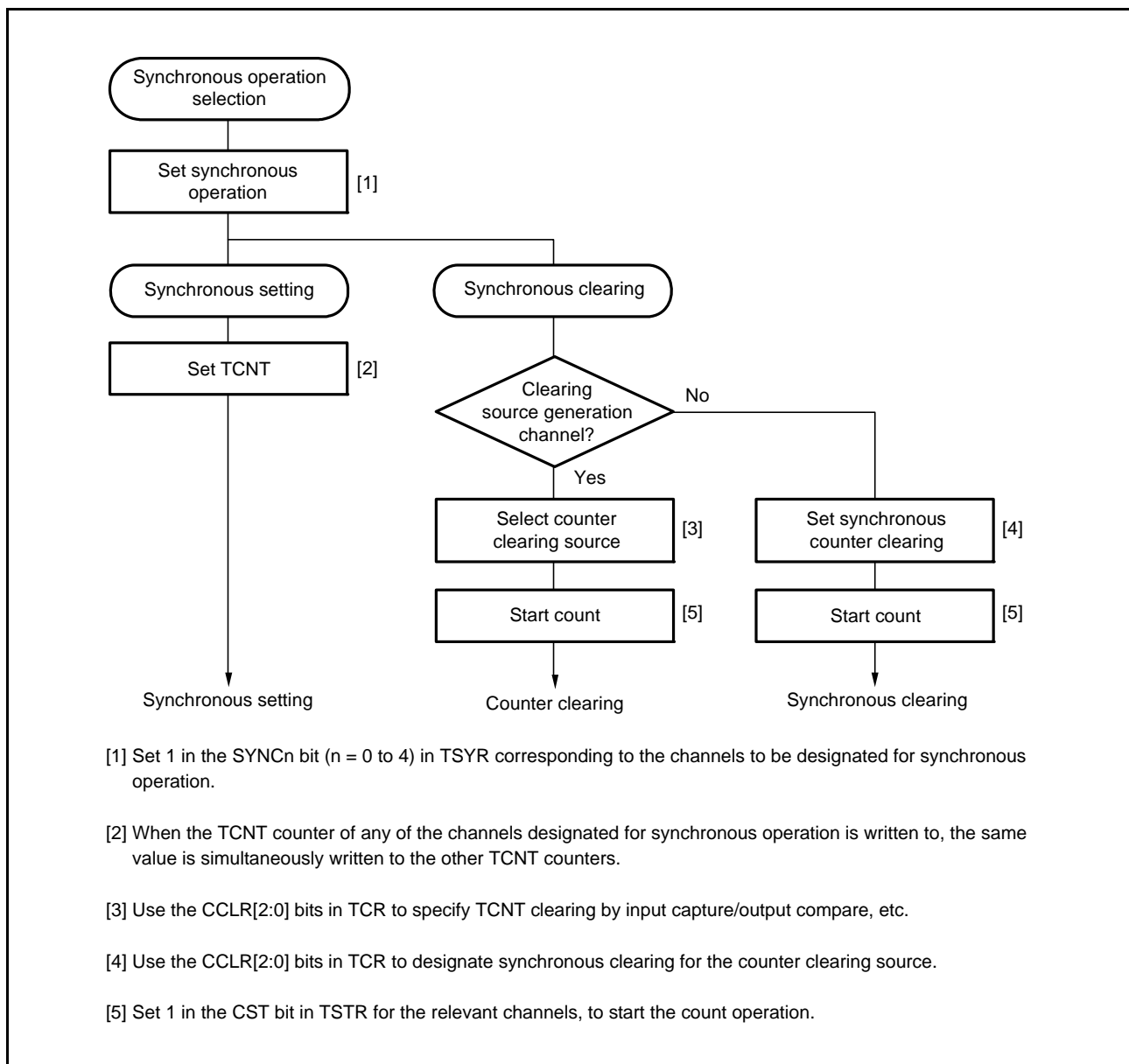
Synchronous operation increases the number of TGR registers assigned to a single time base.

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 20.12 shows an example of the synchronous operation setting procedure.



**Figure 20.12 Example of Synchronous Operation Setting Procedure**

(2) Example of Synchronous Operation

Figure 20.13 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 setting and synchronous clearing by MTU0.TGRB compare match are performed for the TCNT counters in MTU0 to MTU2, and the data set in MTU0.TGRB is used as the PWM cycle.

For details of PWM modes, refer to section 20.3.5, PWM Modes.

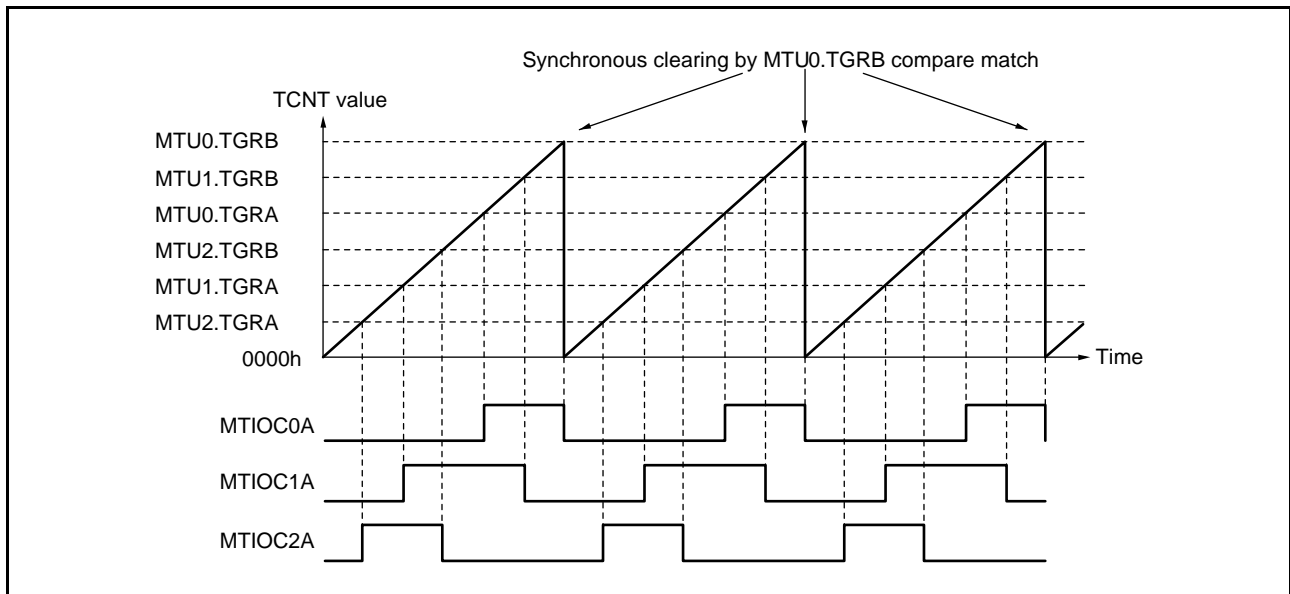


Figure 20.13 Example of Synchronous Operation

### 20.3.3 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 20.43 shows the register combinations used in buffer operation.

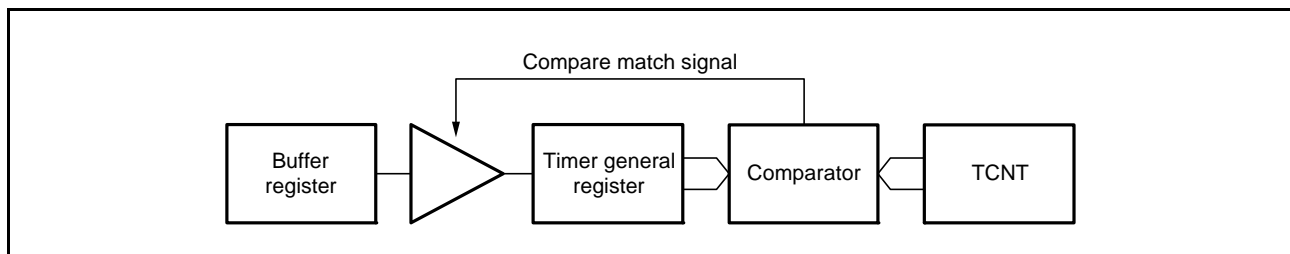
**Table 20.43 Register Combinations in Buffer Operation**

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 20.14.



**Figure 20.14 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 20.15.

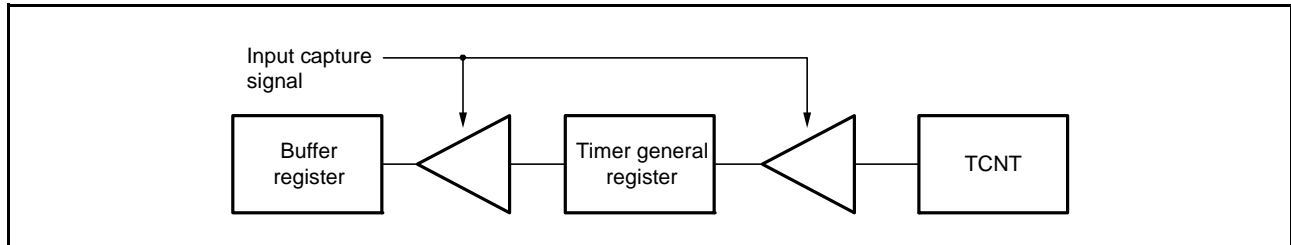


Figure 20.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 20.16 shows an example of the buffer operation setting procedure.

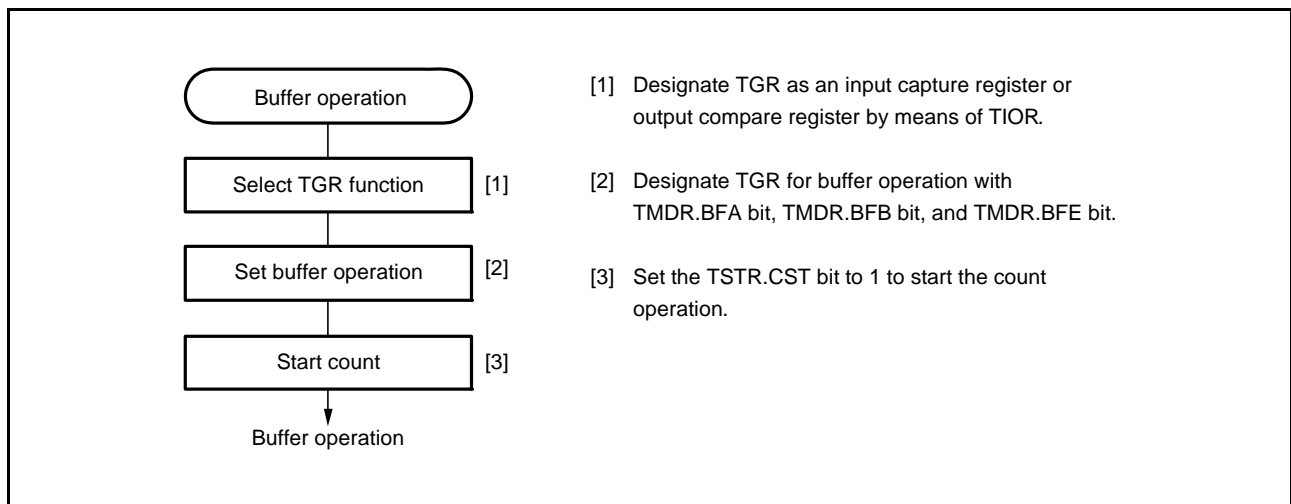


Figure 20.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an Output Compare Register

Figure 20.17 shows an operation example in which PWM mode 1 has been designated for MTU0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. In this example, the TBTM.TTSA bit is set to 0. As buffer operation has been set, when compare match A occurs, the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

For details of PWM modes, refer to section 20.3.5, PWM Modes.

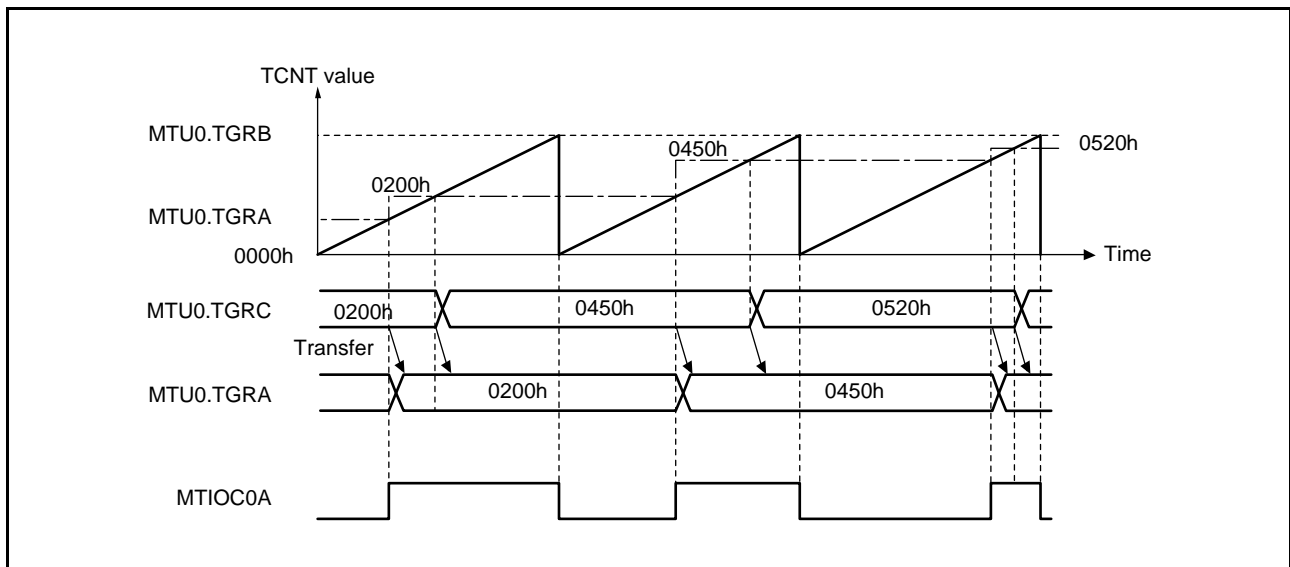


Figure 20.17 Example of Buffer Operation (1)

(b) When TGR is an Input Capture Register

Figure 20.18 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 transferred to TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

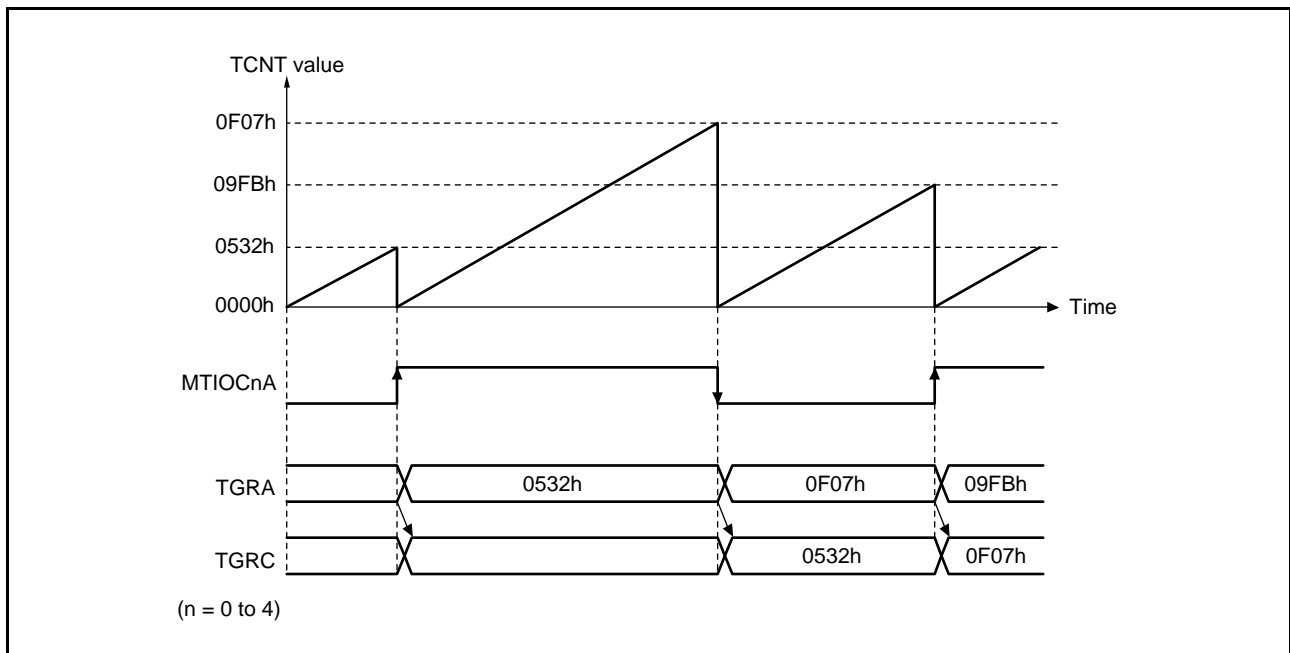


Figure 20.18 Example of Buffer Operation (2)

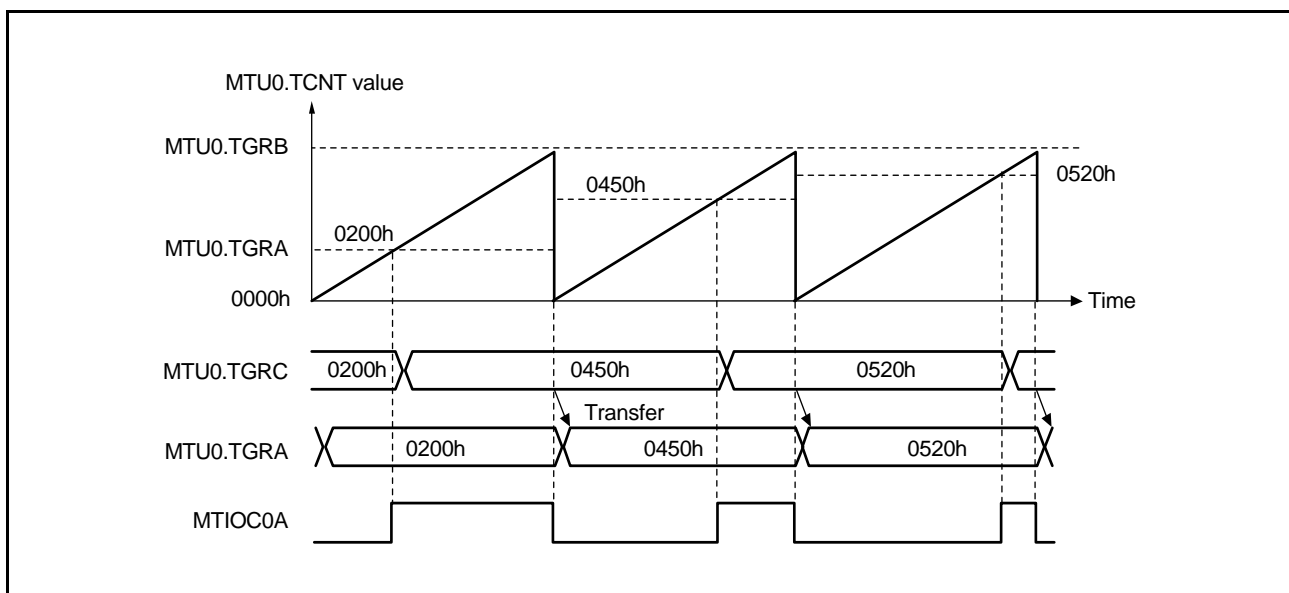
### (3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation

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 timer 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 set to 0000h under the condition specified in the TCR.CCLR[2:0] bits

Note: TBTM must be modified only while TCNT stops.

Figure 20.19 shows an operation example in which PWM mode 1 is designated for MTU0 and buffer operation is designated for MTU0.TGRA and MTU0.TGRC. The settings used in this example are MTU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. The MTU0.TBTM.TTSA bit is set to 1.



**Figure 20.19** Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC-to-MTU0.TGRA Transfer Timing



### 20.3.4 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 count clock for MTU1 through the TCR.TPSC[2:0] bits.

Underflow occurs only when the lower 16 bits of TCNT is in phase counting mode.

Table 20.44 lists the register combinations used in cascaded operation.

Note: When phase counting mode is set for MTU1 or MTU2, the count clock setting is invalid and the counters operate independently in phase counting mode.

**Table 20.44 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 TICCR register. 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, 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 20.6.22, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.

Table 20.45 lists the TICCR setting and input capture input pins.

**Table 20.45 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 20.20 shows an example of the cascaded operation setting procedure.

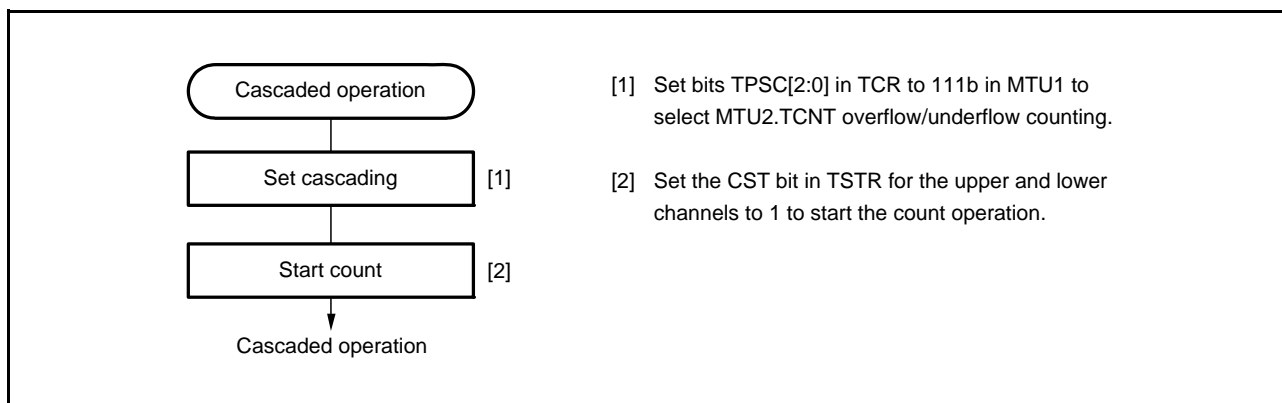


Figure 20.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 20.21 shows the operation when MTU1.TCNT is set for counting at MTU2.TCNT overflow/underflow and MTU2 is set for phase counting mode 1 while MTU1.TCNT and MTU2.TCNT are cascaded. MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

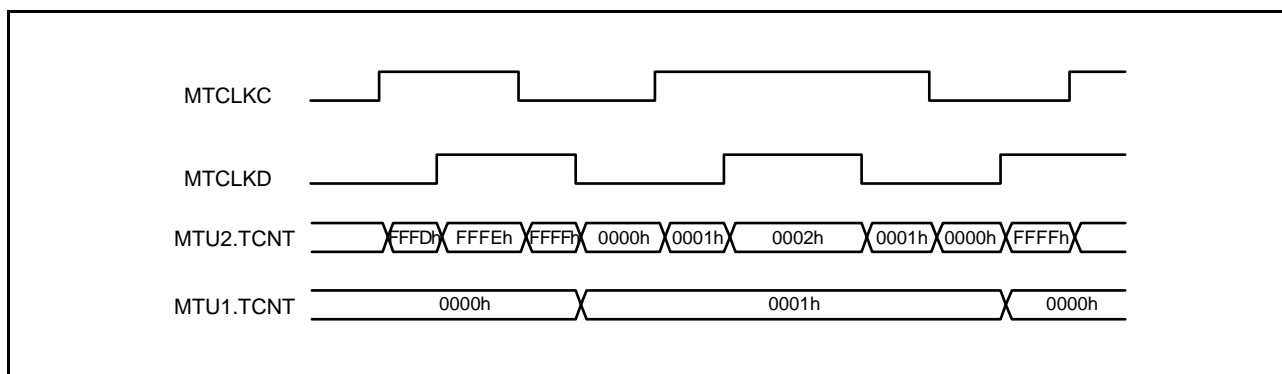


Figure 20.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 20.22 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE bit has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA[3:0] bits in MTU1.TIOR have selected the MTIOC1A rising edge for the input capture timing while the IOA[3:0] bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, 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.

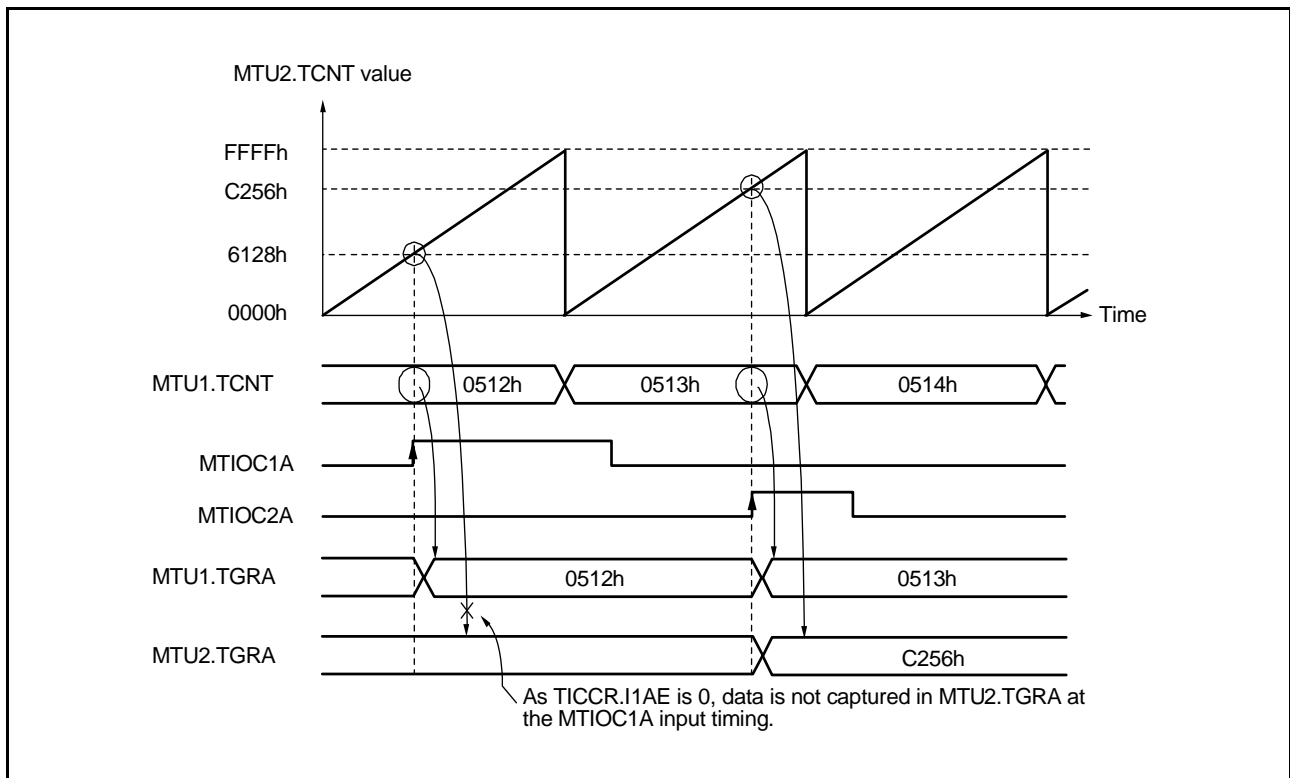


Figure 20.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 20.23 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE and I1AE bits in TICCR have been set to 1 to include the MTIOC2A and MTIOC1A pins in the MTU1.TGRA and MTU2.TGRA input capture conditions, respectively. In this example, the IOA[3:0] bits in both MTU1.TIOR and MTU2.TIOR have selected both the rising and falling edges for the input capture timing. Under these conditions, the OR result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.

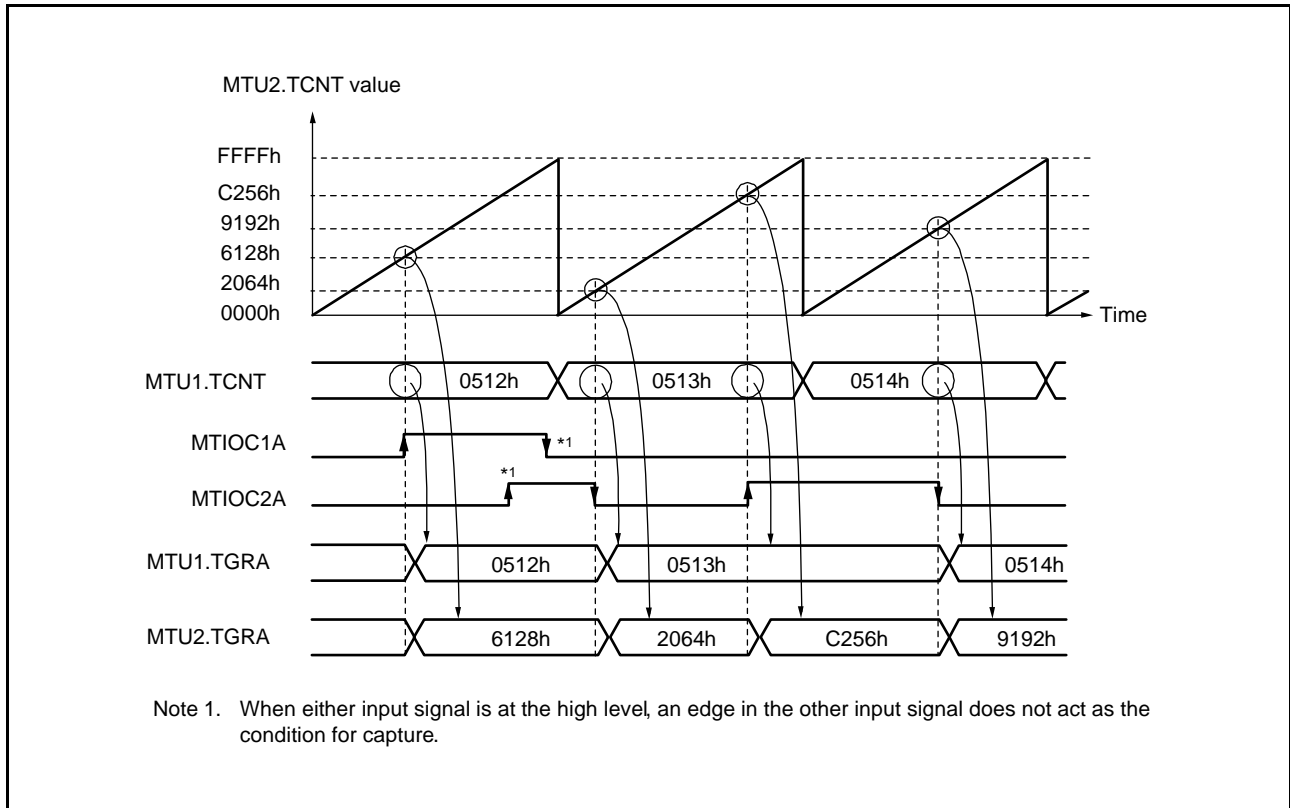


Figure 20.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 20.24 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE bit has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA[3:0] bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA[3:0] bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, as MTU1.TIOR has selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing, the MTIOC2A edge is not used for MTU1.TGRA input capture condition although the TICCR.I2AE bit has been set to 1.

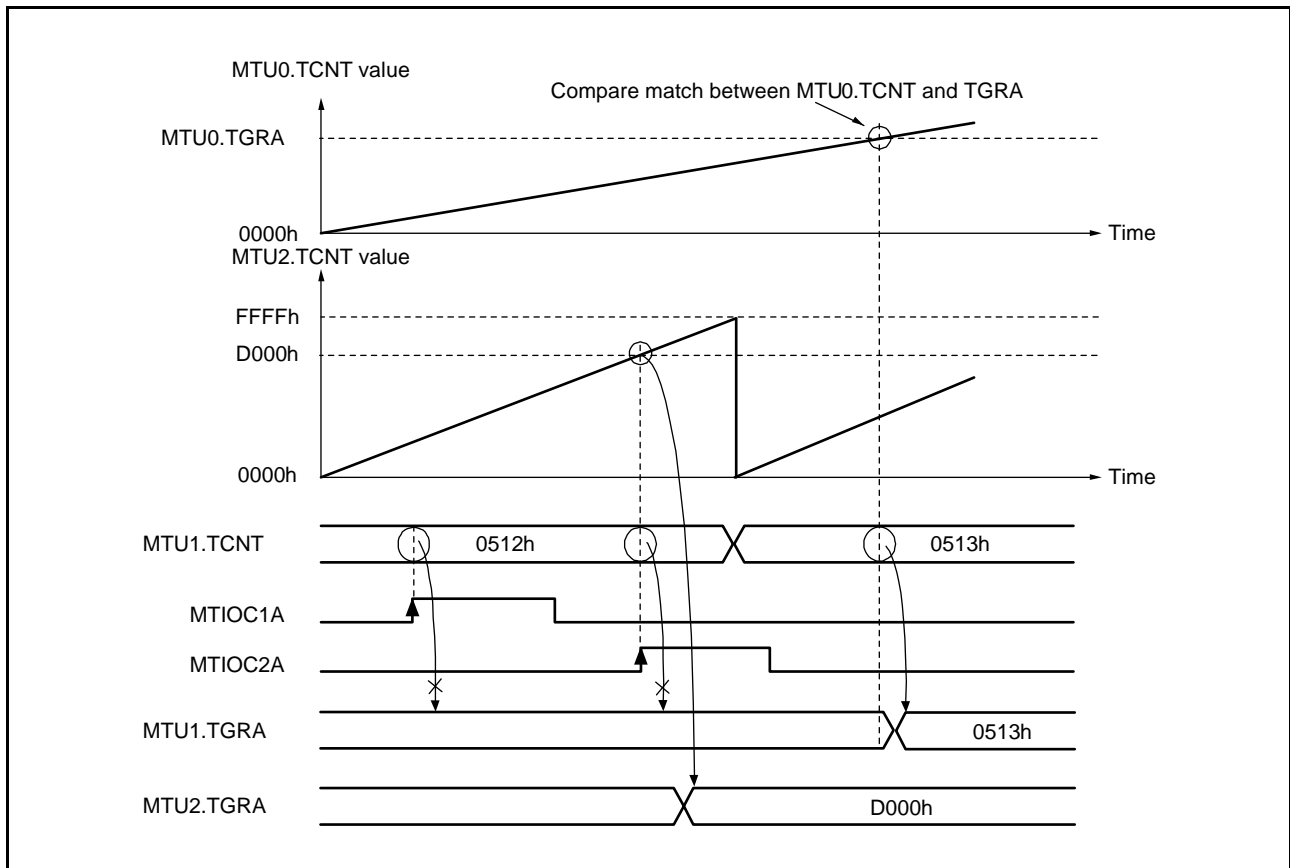


Figure 20.24 Cascaded Operation Example (d)

### 20.3.5 PWM Modes

PWM modes are provided to output PWM waveforms from the external pins. The output level can be selected as low, high, or toggle output in response to a compare match of each TGR.

PWM waveforms in the range of 0% to 100% duty cycle can be output according to the TGR settings.

By designating TGR compare match as the counter clearing source, the PWM cycle can be specified in that register.

Each 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 the TIOR.IOA[3:0] and IOC[3:0] bits are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the levels specified by the TIOR.IOB[3:0] and IOD[3:0] bits 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 cycle 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 20.46.

**Table 20.46 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 20.25 shows an example of the PWM mode setting procedure.

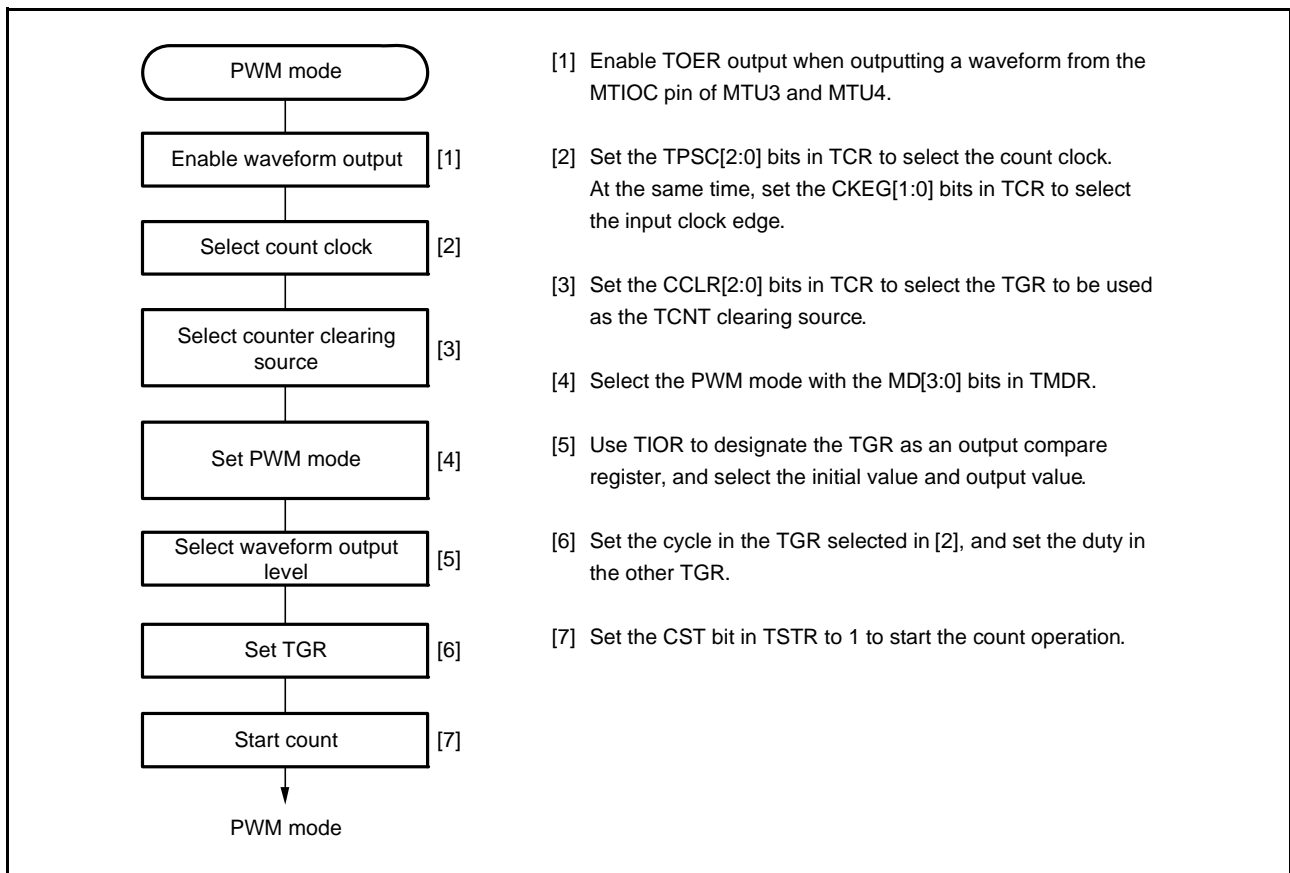


Figure 20.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 20.26 shows an example of operation in PWM mode 1.

In this example, TGRA compare match is set as the TCNT clearing source, a low level is set as the initial output value and output value for TGRA, and a high level is set as the output value for TGRB.

In this case, the value set in TGRA is used as the cycle, and the value set in TGRB is used as the duty.

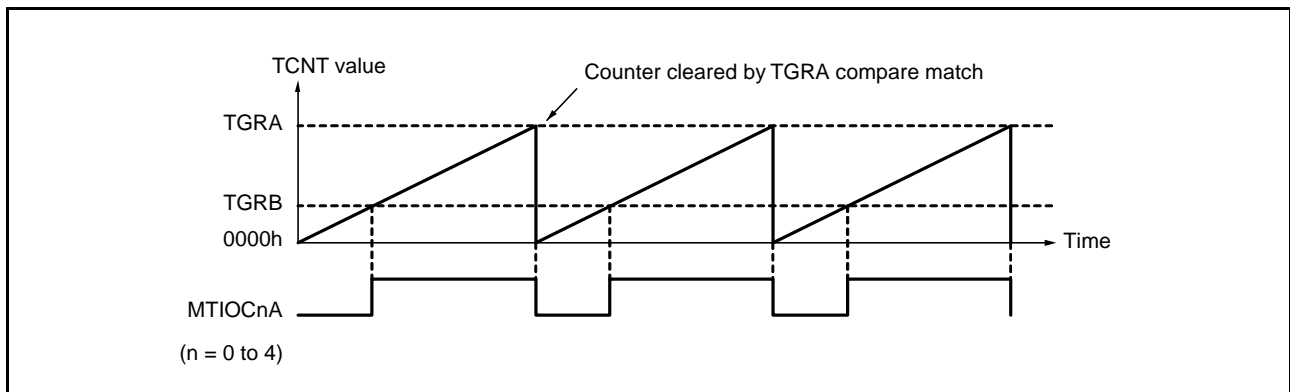


Figure 20.26 Example of PWM Mode Operation

Figure 20.27 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 a low level is set as the initial output value and a high level 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.

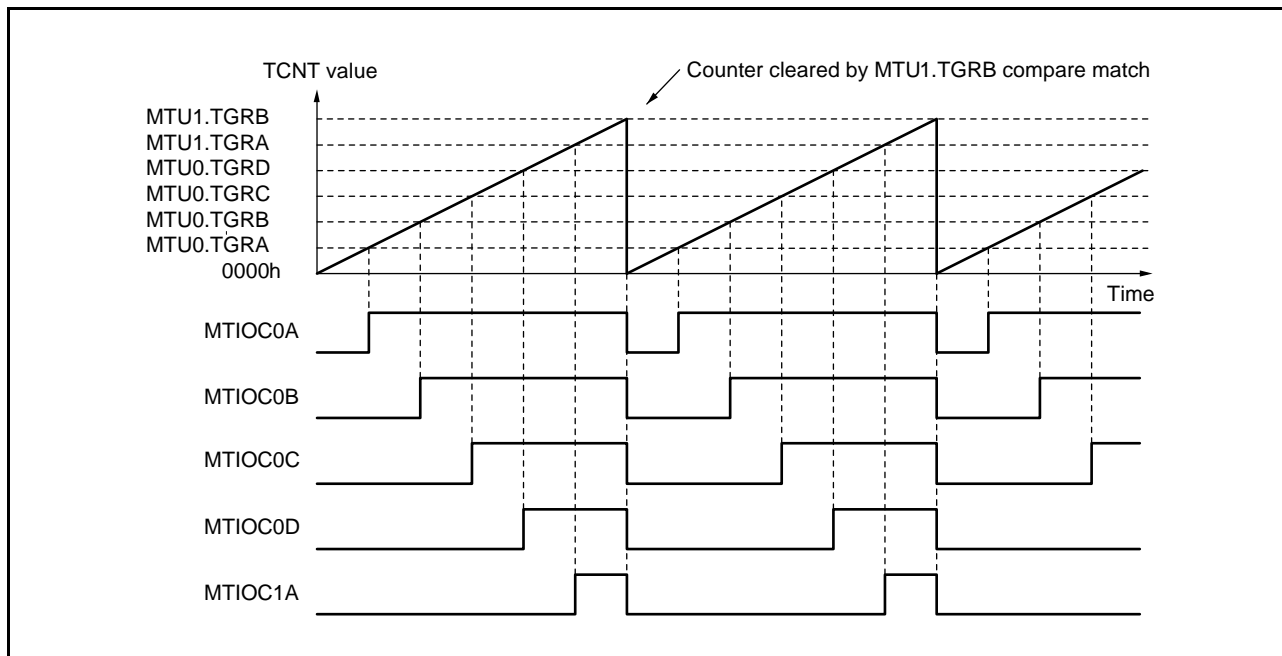


Figure 20.27 Example of PWM Mode Operation



Figure 20.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

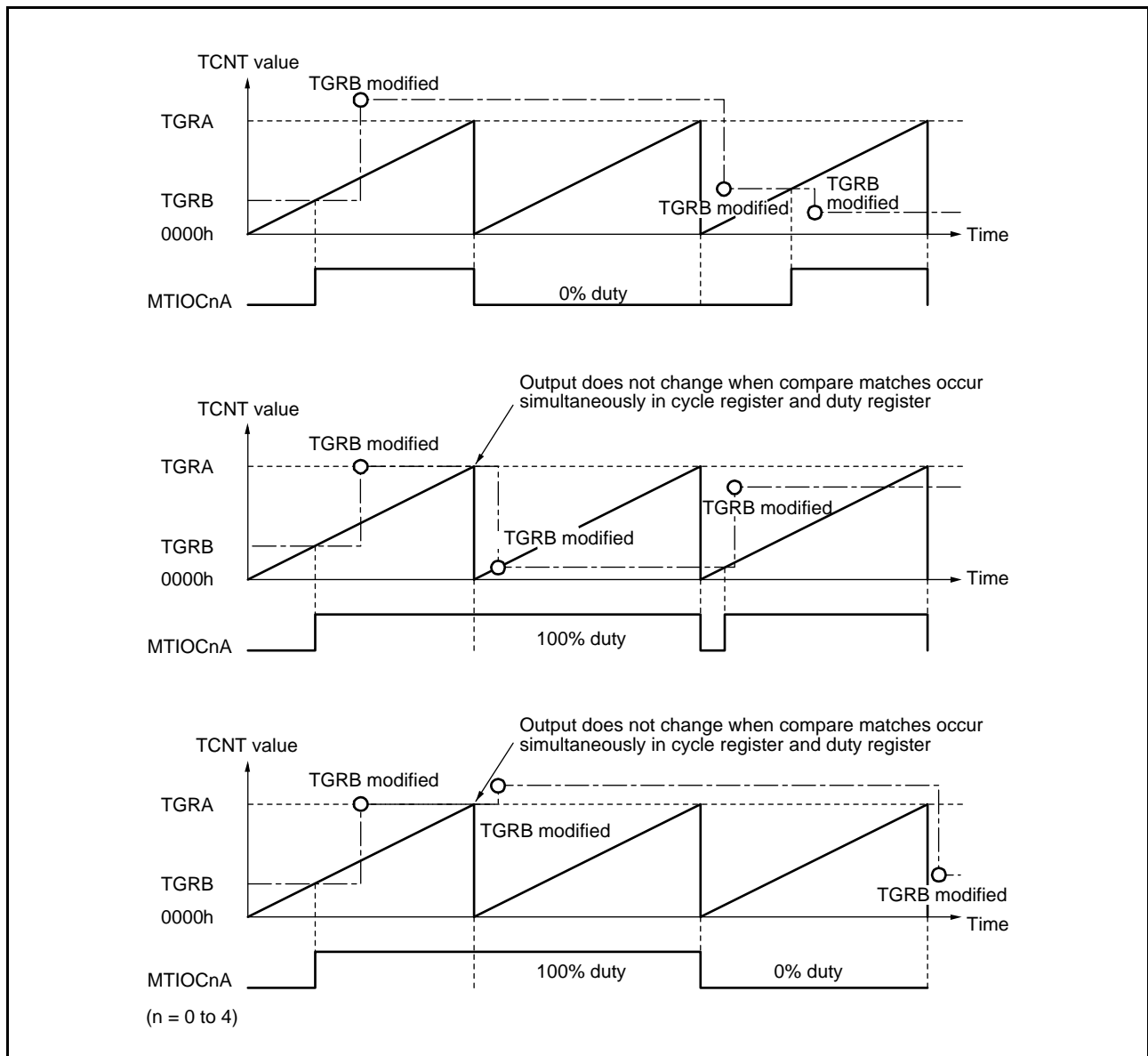


Figure 20.28 Examples of PWM Mode Operation

### 20.3.6 Phase Counting Mode

When phase counting mode is specified, an external clock is selected as the counter input clock and TCNT operates as an up-counter/down-counter regardless of the setting of the TCR.TPSC[2:0] bits and TCR.CKEG[1:0] bits. However, the functions of the TCR.CCLR[2:0] bits and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

This can be used for 2-phase encoder pulse input.

If an overflow occurs while TCNT is counting up, a TCIV interrupt is generated while the corresponding TIER.TCIEV bit is 1. If an underflow occurs while TCNT is counting down, a TCIU interrupt is generated while the corresponding TIER.TCIEU bit is 1.

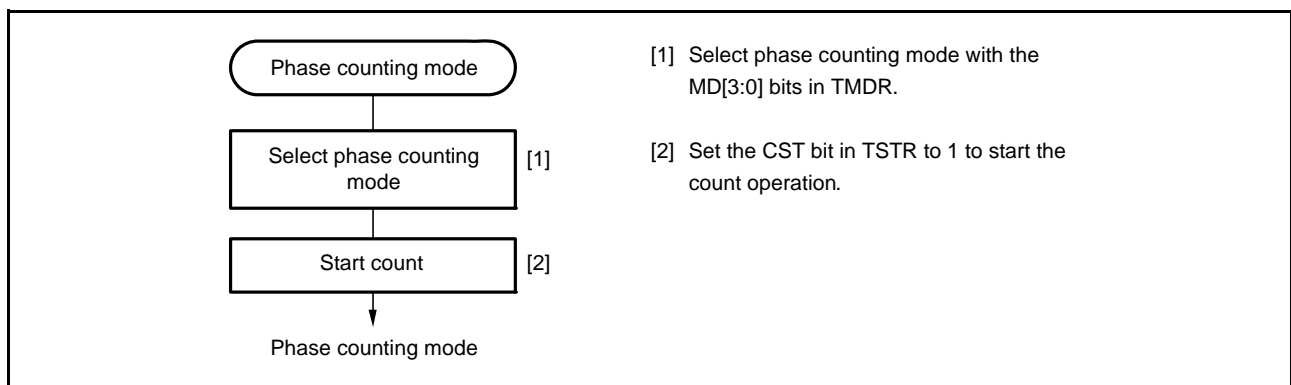
The TSR.TCFD flag is the count direction flag. Read the TCFD flag to check whether TCNT is counting up or down. In phase counting mode, the external clock pins MTCLKA, MTCLKB, MTCLKC, and MTCLKD can be used as 2-phase encoder pulse input pins. Table 20.47 lists the correspondence between external clock pins and channels.

**Table 20.47 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 20.29 shows an example of the phase counting mode setting procedure.



**Figure 20.29 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 20.30 shows an example of operation in phase counting mode 1, and Table 20.48 lists the TCNT up-counting and down-counting conditions.

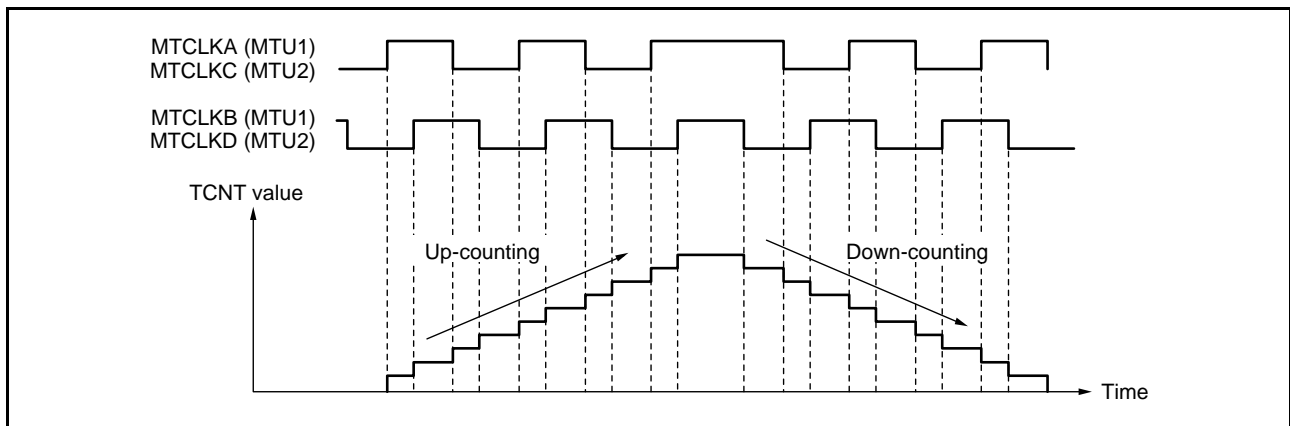


Figure 20.30 Example of Operation in Phase Counting Mode 1

Table 20.48 Up-Counting and Down-Counting Conditions in Phase Counting Mode 1

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	
↓	Low	

↑ : Rising edge  
 ↓ : Falling edge

(b) Phase Counting Mode 2

Figure 20.31 shows an example of operation in phase counting mode 2, and Table 20.49 lists the TCNT up-counting and down-counting conditions.

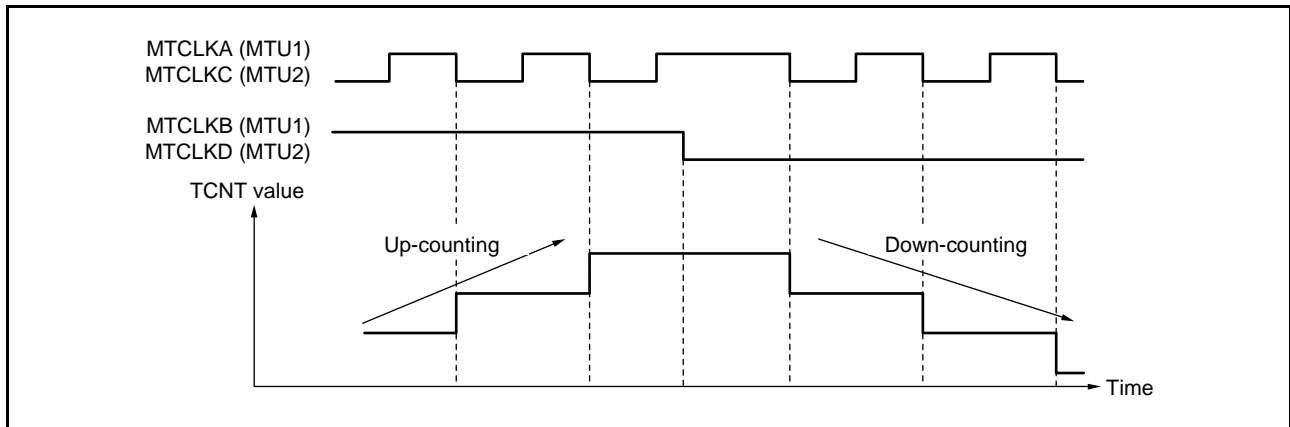


Figure 20.31 Example of Operation in Phase Counting Mode 2

Table 20.49 Up-Counting and Down-Counting 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 20.32 shows an example of operation in phase counting mode 3, and Table 20.50 lists the TCNT up-counting and down-counting conditions.

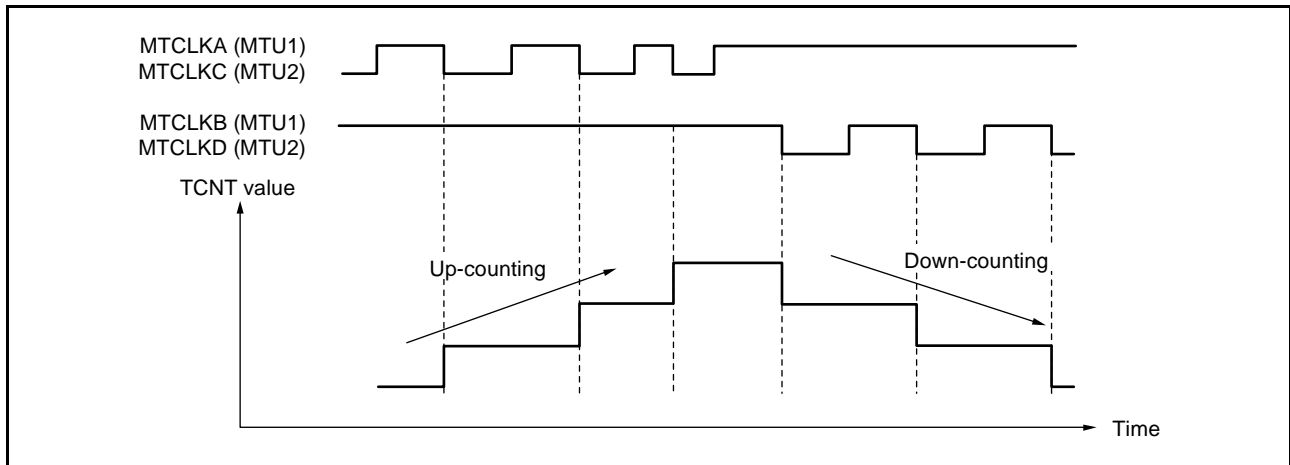


Figure 20.32 Example of Operation in Phase Counting Mode 3

Table 20.50 Up-Counting and Down-Counting 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 20.33 shows an example of operation in phase counting mode 4, and Table 20.51 lists the TCNT up-counting and down-counting conditions.

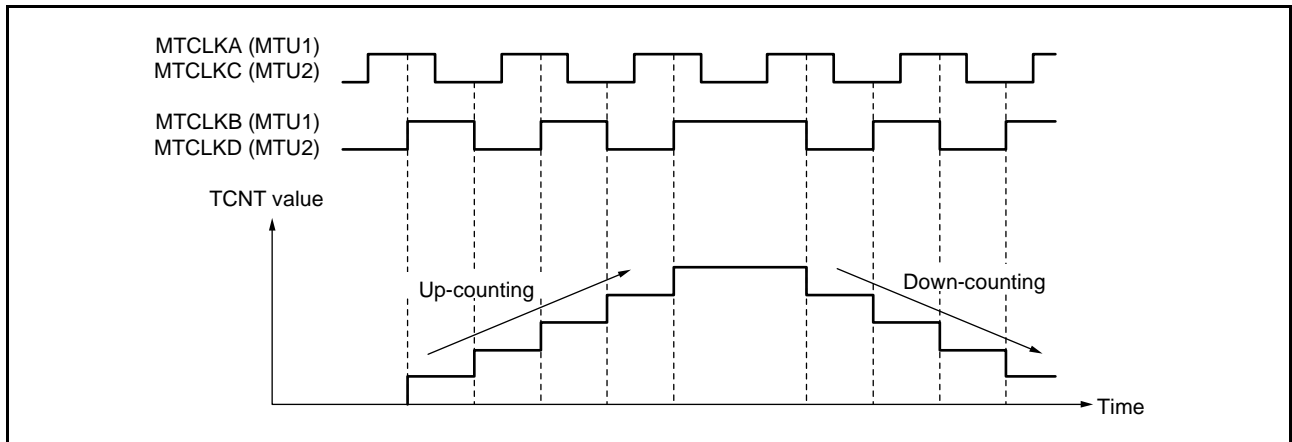


Figure 20.33 Example of Operation in Phase Counting Mode 4

Table 20.51 Up-Counting and Down-Counting Conditions in Phase Counting Mode 4

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High		Up-counting
Low		
	Low	None (Don't care)
	High	
High		Down-counting
Low		
	High	None (Don't care)
	Low	

: Rising edge  
 : Falling edge

### (3) Phase Counting Mode Application Example

Figure 20.34 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.

MTU0.TGRC compare match is specified as the MTU0.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 MTU0.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 MTU1.TGRB for MTU1 are designated for the input capture function and MTU0.TGRA and MTU0.TGRC compare matches in MTU0 are selected as the input capture sources to store the up-counter/down-counter values for the control cycles.

This procedure enables the accurate detection of position and speed.

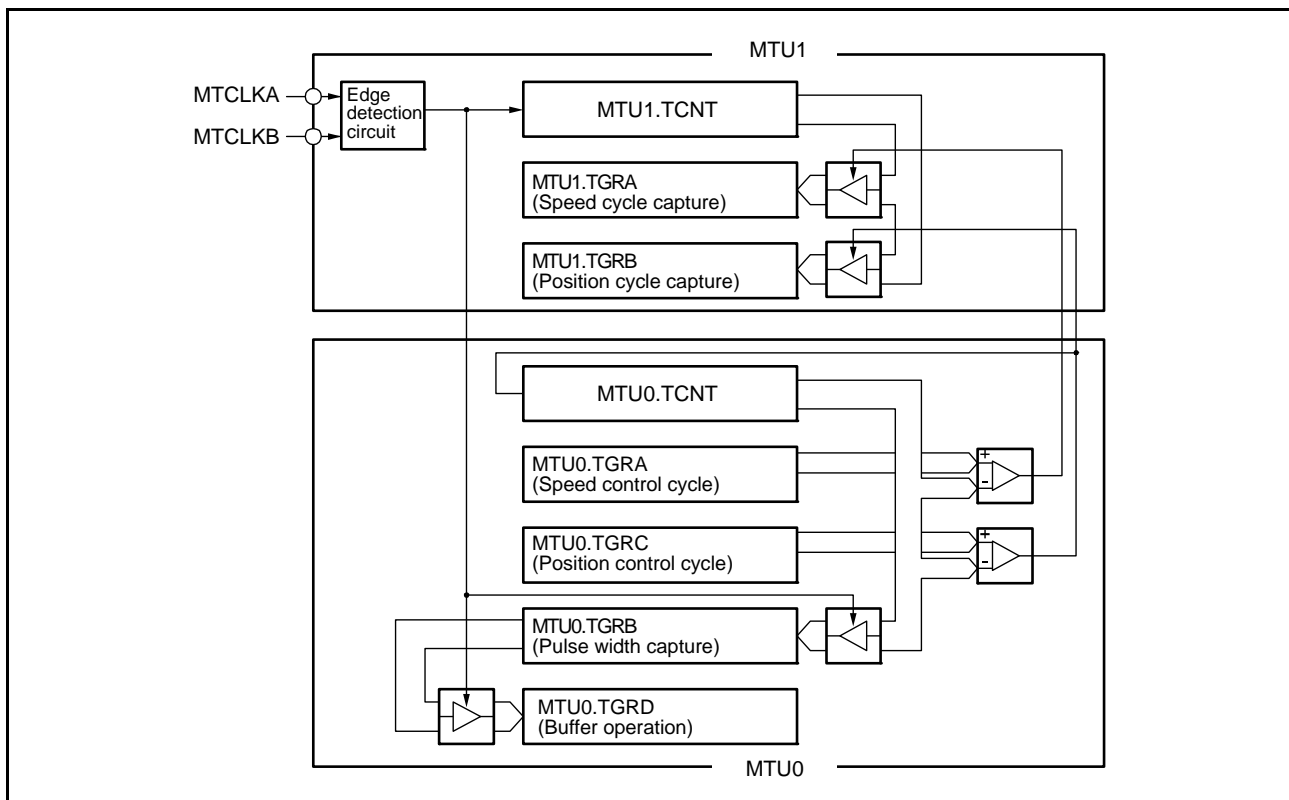


Figure 20.34 Phase Counting Mode Application Example

### 20.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, six 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 the MTU3.TCNT counter functions as an up-counter.

Table 20.52 lists the PWM output pins. Table 20.53 lists the settings of the registers.

**Table 20.52 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 20.53 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 20.35 shows an example of procedure for setting the reset-synchronized PWM mode.

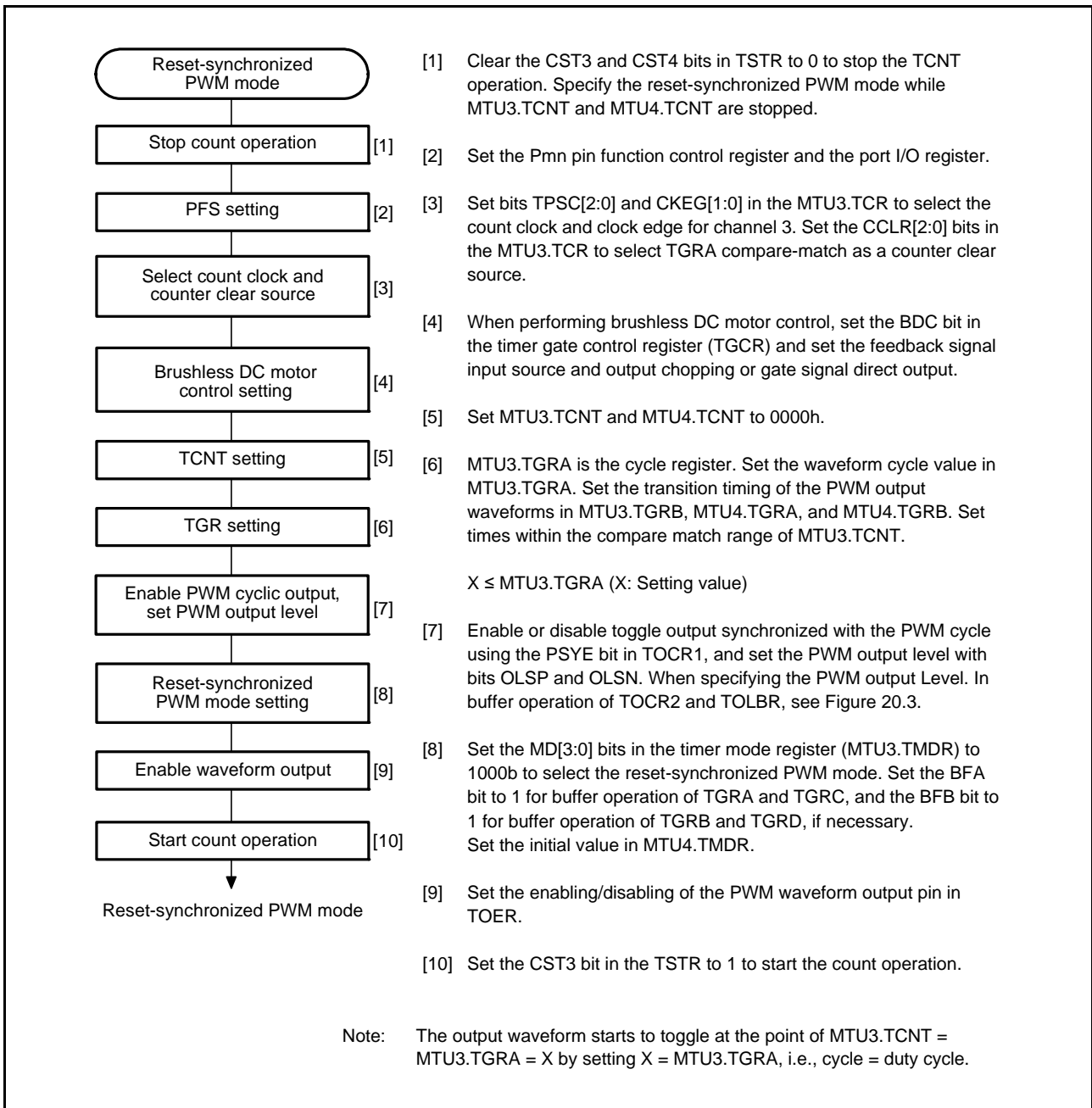


Figure 20.35 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 20.36 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.

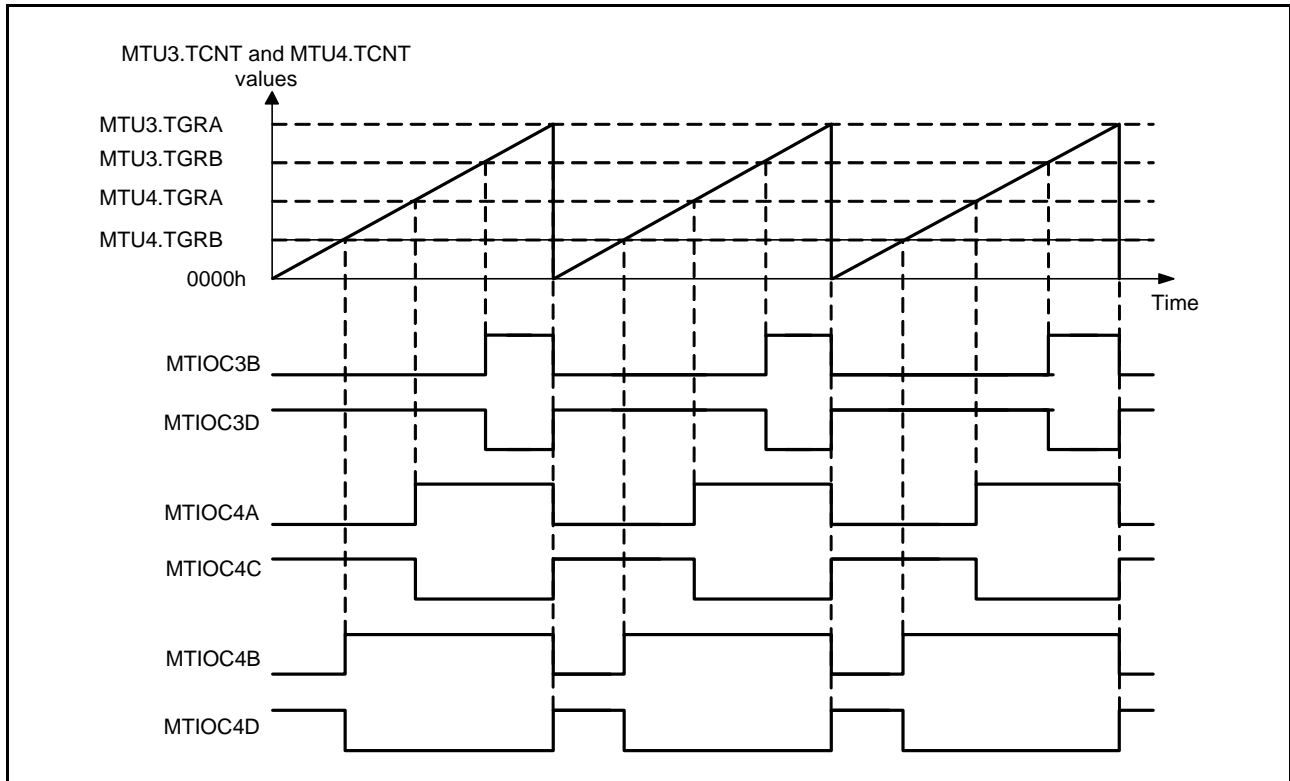


Figure 20.36 Example of Reset-Synchronized PWM Mode Operation (When TOCR1.OLSN = 1 and OLSP = 1)

### 20.3.8 Complementary PWM Mode

In complementary PWM mode, dead time can be set for PWM waveforms to be output. The dead time is the period during which the upper and lower arm transistors are set to the inactive level in order to prevent short-circuiting of the arms. Six phases of positive and negative PWM waveforms with dead time can be output by combining MTU3 and MTU4. PWM waveforms without dead time can also be output.

In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins function as PWM output pins, and the MTIOC3A pin can be set for toggle output synchronized with the PWM cycle. MTU3.TCNT and MTU4.TCNT function as up/down-counters.

Table 20.54 lists the PWM output pins used. Table 20.55 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 20.54 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' (negative-phase waveform output of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform output of PWM output 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform output of PWM output 3)

Note 1. Avoid setting the MTIOC3C pin as a timer I/O pin in complementary PWM mode.

**Table 20.55 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	Readable/writable
	MTU3.TGRD	PWM output 1/MTU3.TGRB buffer register	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	Readable/writable
	MTU4.TGRD	PWM output 3/MTU4.TGRB buffer register	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	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).

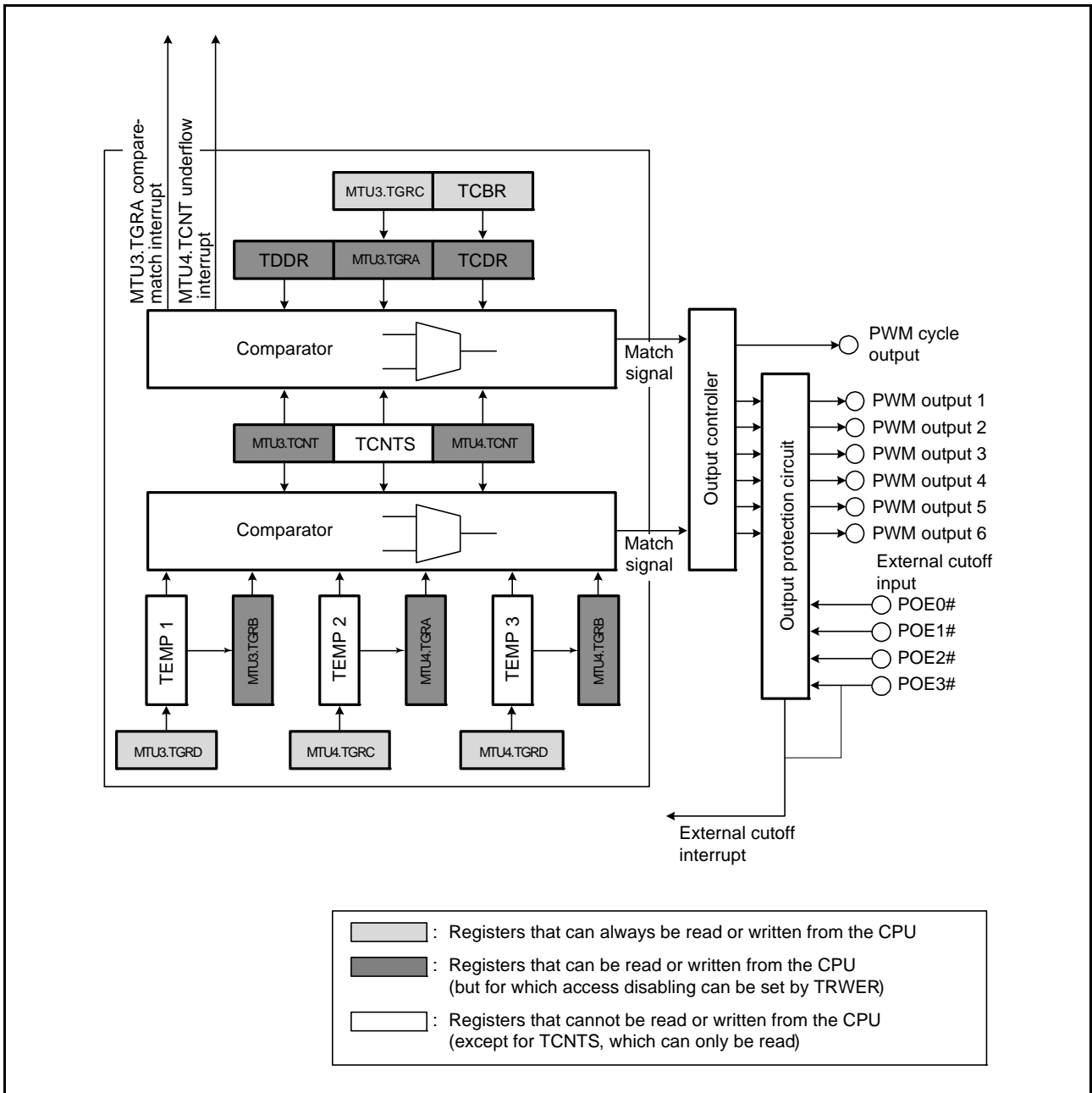


Figure 20.37 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

Figure 20.38 shows an example of the complementary PWM mode setting procedure.

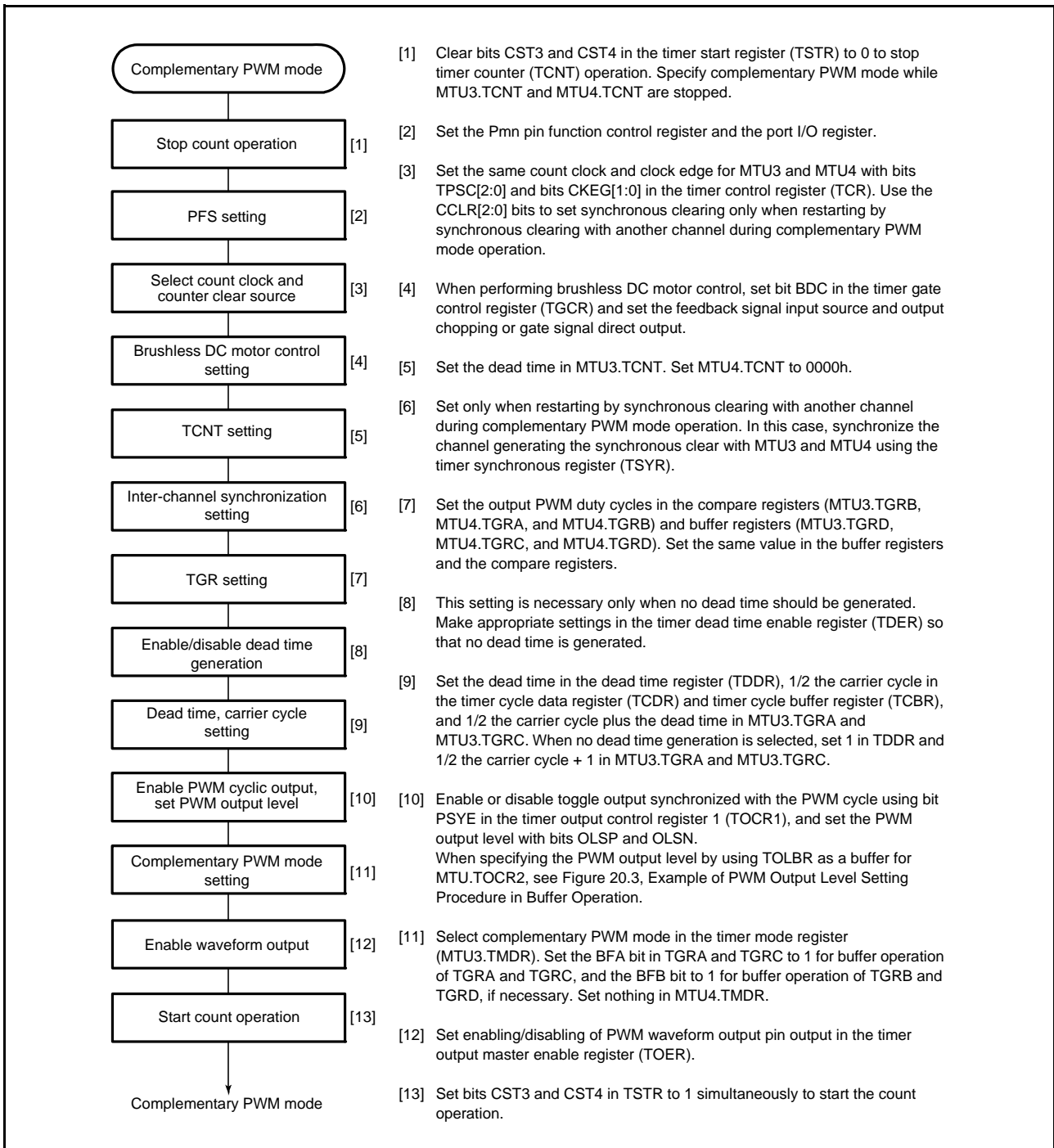


Figure 20.38 Example of Complementary PWM Mode Setting Procedure

## (2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases (three positive and three negative) of PWM waveforms can be output. Figure 20.39 illustrates counter operation in complementary PWM mode, and Figure 20.40 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 TSTR.CST bit 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.

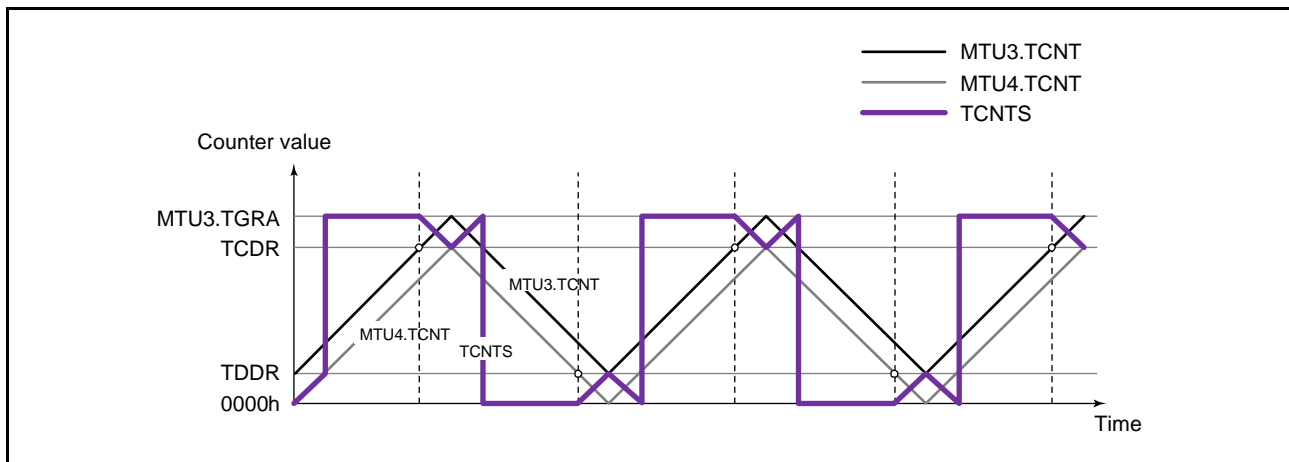


Figure 20.39 Counter Operation in Complementary PWM Mode

### (b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used to control the duty ratio for the PWM output. Figure 20.40 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 the TOCR1.OLSN and OLSP bits is output from the PWM output pin.

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.

When modifying data in a buffer register, be sure to write to MTU4.TGRD last and enable data transfer from the buffer register to a temporary register. At this time, transfer from TCBR and MTU3.TGRC, which operate as buffer registers for the timer cycle registers, to temporary registers is also enabled. Data is transferred to all five temporary registers at the same time. When transfer is enabled in the Ta interval, data written to a buffer register is immediately transferred to the temporary register. Data is not transferred to the temporary register in the Tb1 and Tb2 intervals. Data enabled for transfer in this interval is transferred to the temporary register at the end of this interval.

The value transferred to a temporary register is transferred to the compare register at the end of the Tb1 interval (when matches MTU3.TGRA while TCNTS is counting up), or at the end of the Tb2 interval (when matches 0000h while TCNTS is counting down). The timing for transfer from the temporary register to the compare register can be selected with the TMDR.MD[3:0] bits. Figure 20.40 shows an example in which the trough is selected for the transfer timing. In the Tb (Tb2 in Figure 20.40) 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.



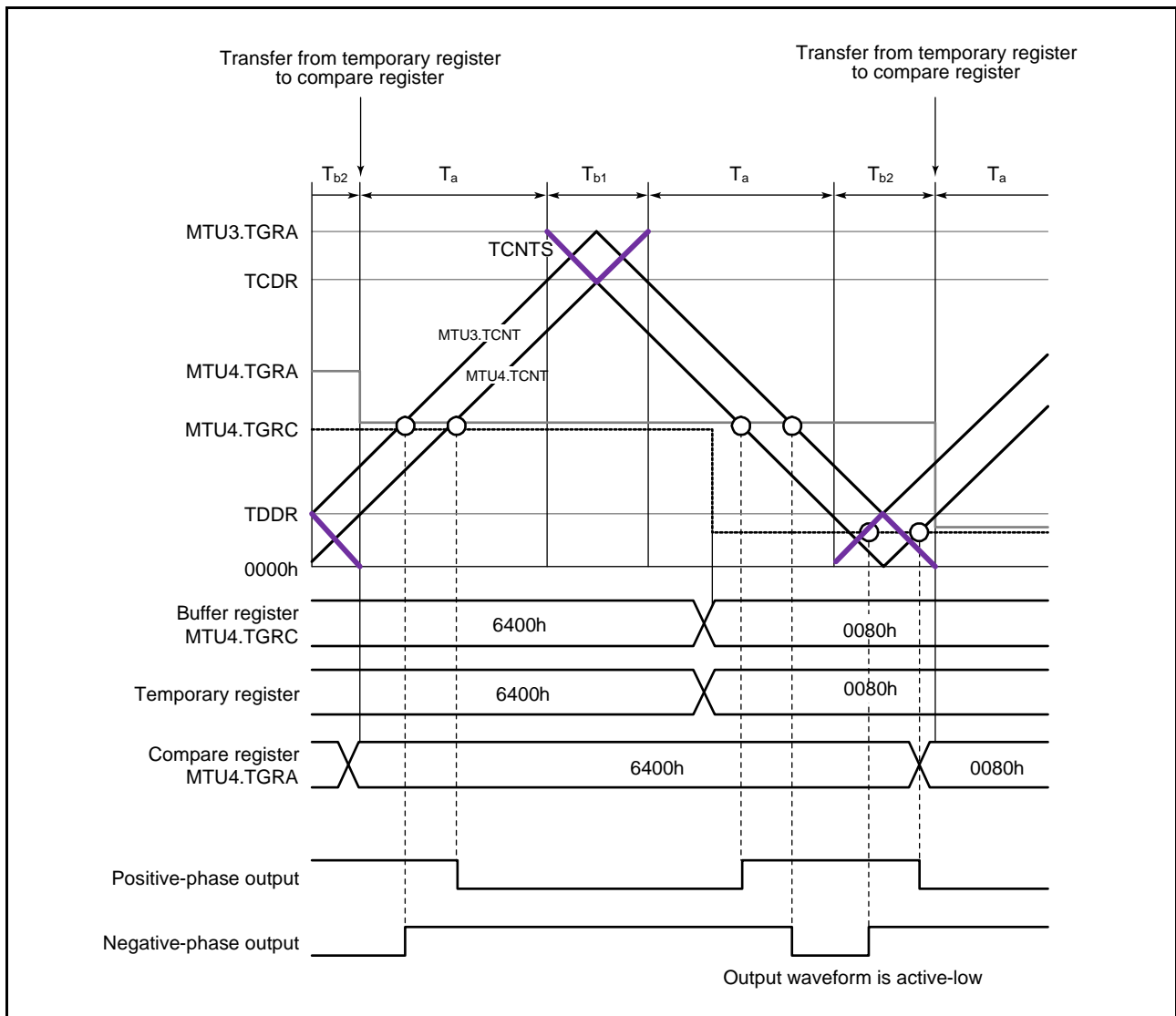


Figure 20.40 Example of Operation in Complementary PWM Mode

### (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 the TMDR.MD[3:0] bits, 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 cycle + dead time Td. 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 cycle. Set dead time Td 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 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 20.56 Registers and Counters Requiring Initial Setting**

Register and Counter	Setting
MTU3.TGRC	1/2 PWM cycle + dead time Td (1/2 PWM 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 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 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 cycle + 1.

### (d) PWM Output Level Setting

In complementary PWM mode, the PWM output level is set with bits OLSN and OLSP in timer output control register 1 (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, dead time can be set for PWM output.

The dead 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 dead time 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.TDER bit to 0. The TDER bit can be cleared to 0 only when 0 is written to it after reading it as 1.

MTU3.TGRA and MTU3.TGRC should be set to 1/2 PWM cycle + 1 and the TDDR register should be set to 1.

By the above settings, PWM waveforms without dead time can be obtained. Figure 20.41 shows an example of operation without dead time.

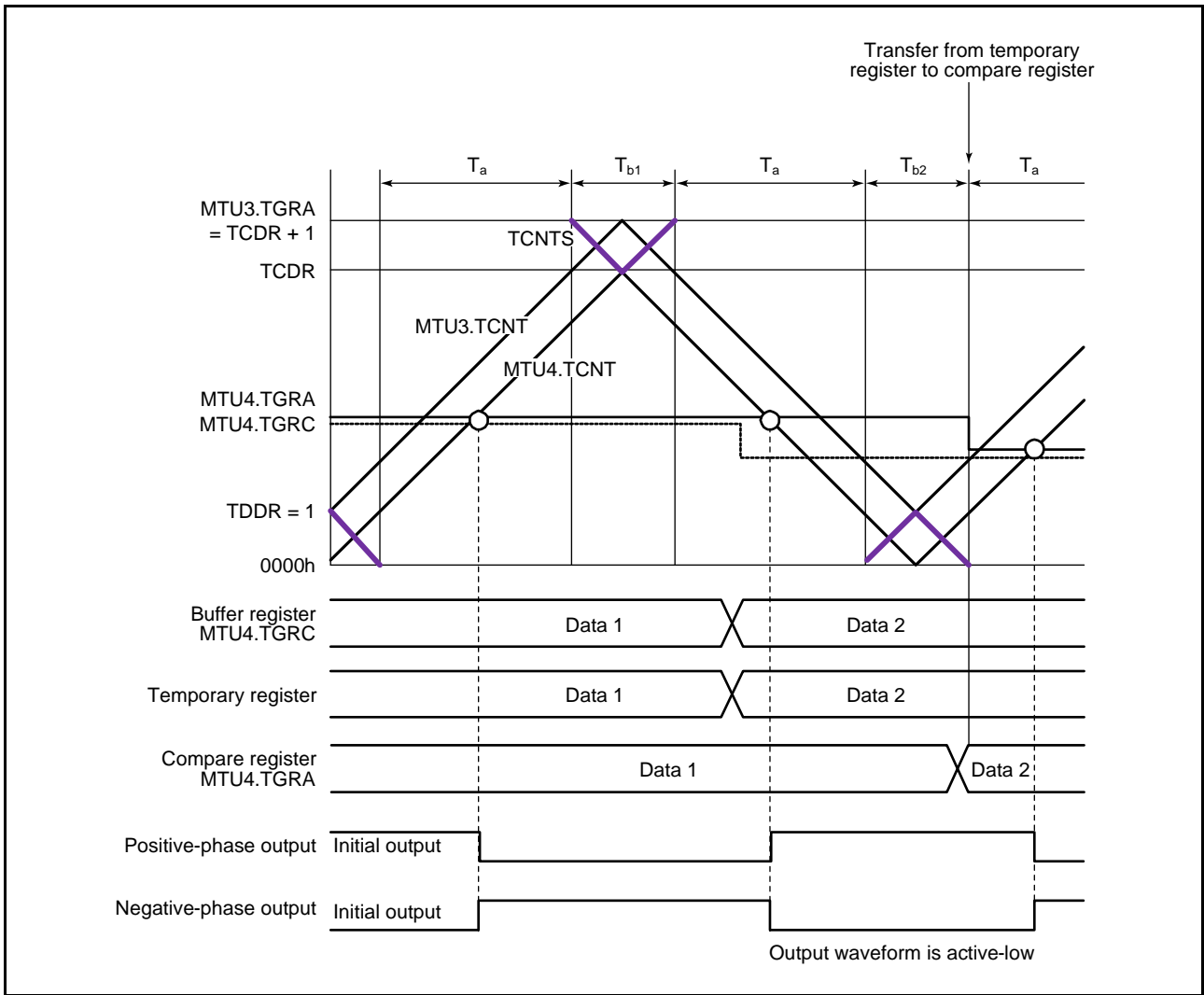


Figure 20.41 Example of Operation without Dead Time

### (g) PWM Cycle Setting

In complementary PWM mode, the PWM 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:  $\text{MTU3.TGRA setting} = \text{TCDR setting} + \text{TDDR setting}$

Without dead time:  $\text{MTU3.TGRA setting} = \text{TCDR setting} + 1$

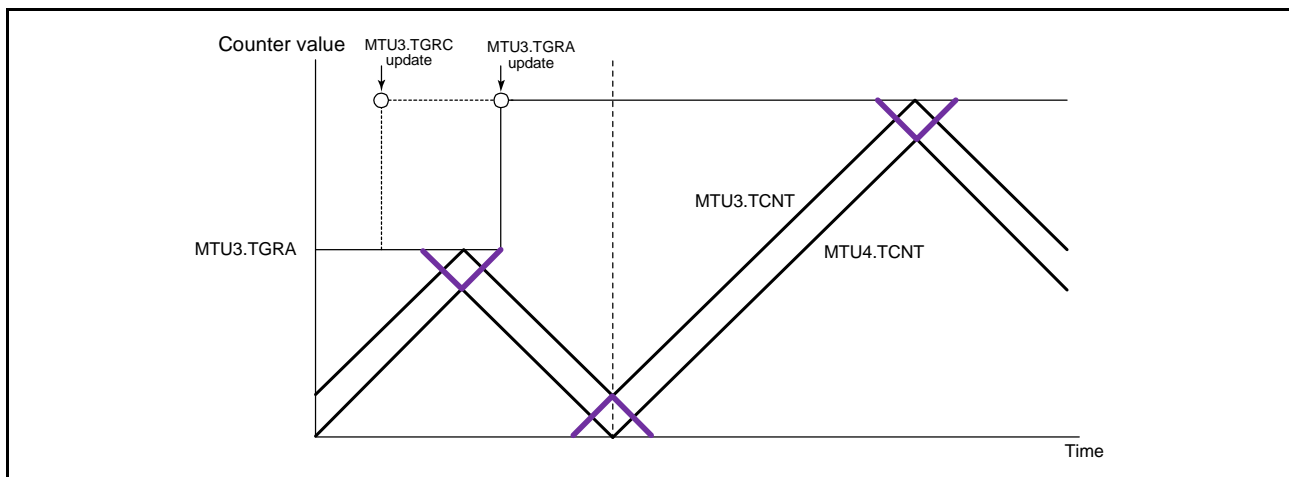
The settings should be made so as to achieve the following relationship between registers TCDR and TDDR.

$\text{TCDR setting} > \text{TDDR setting} \times 2 + 2$

The MTU3.TGRA and TCDR settings are made by setting values in buffer registers MTU3.TGRC and TCBR. When data is written to MTU4.TGRD to enable transfers, the values set in MTU3.TGRC and TCBR are transferred simultaneously to MTU3.TGRA and TCDR with the transfer timing selected with the TMDR.MD[3:0] bits.

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 20.42 illustrates the operation when the PWM cycle is updated at the crest.

Refer to the following section (h), Register Data Updating, for the method of updating the data in each buffer register.



**Figure 20.42** 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 PMW 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 the TMDR.MD[3:0] bits. Figure 20.43 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.

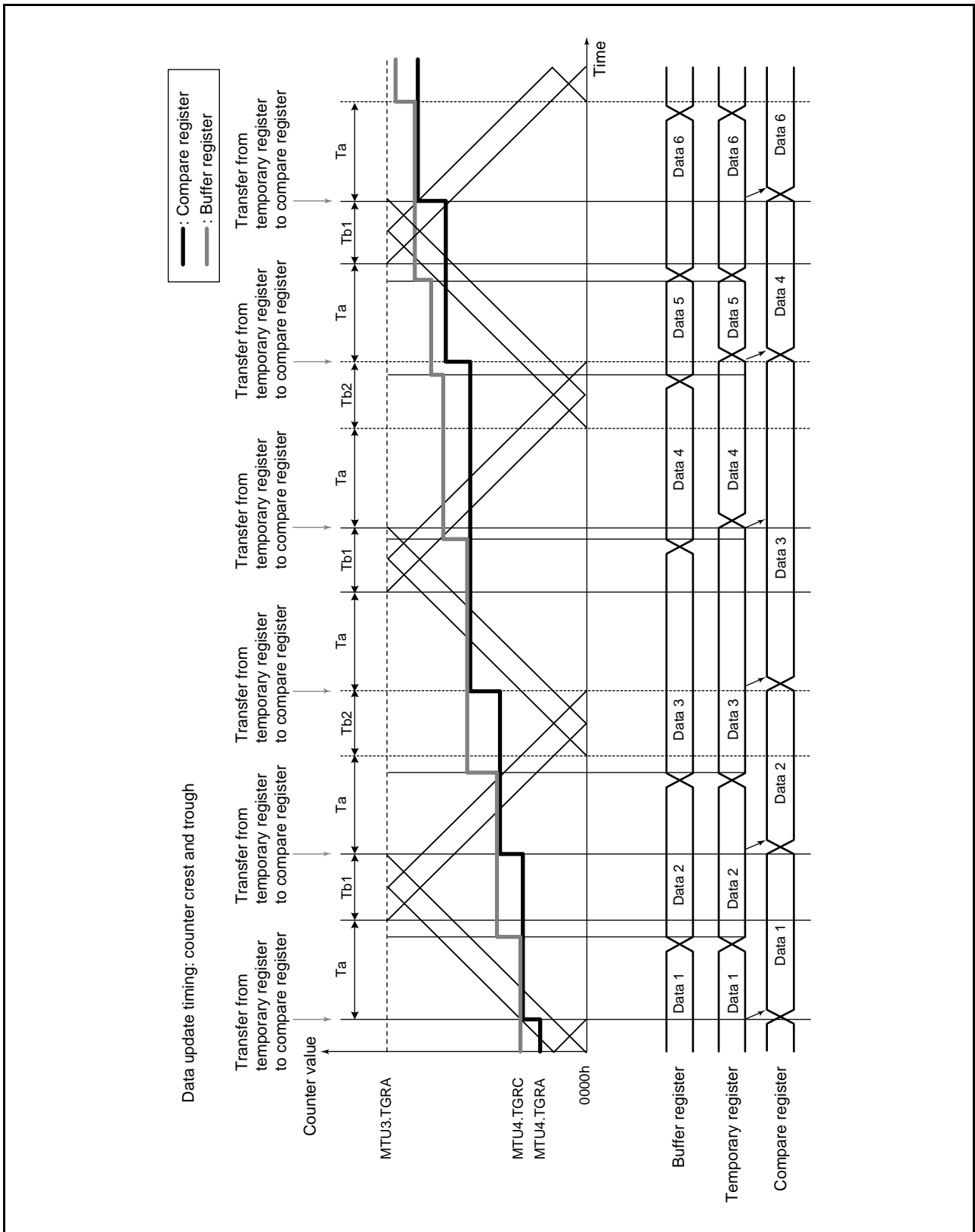


Figure 20.43 Example of Data Updating in Complementary PWM Mode

(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 output 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 20.44 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 20.45.

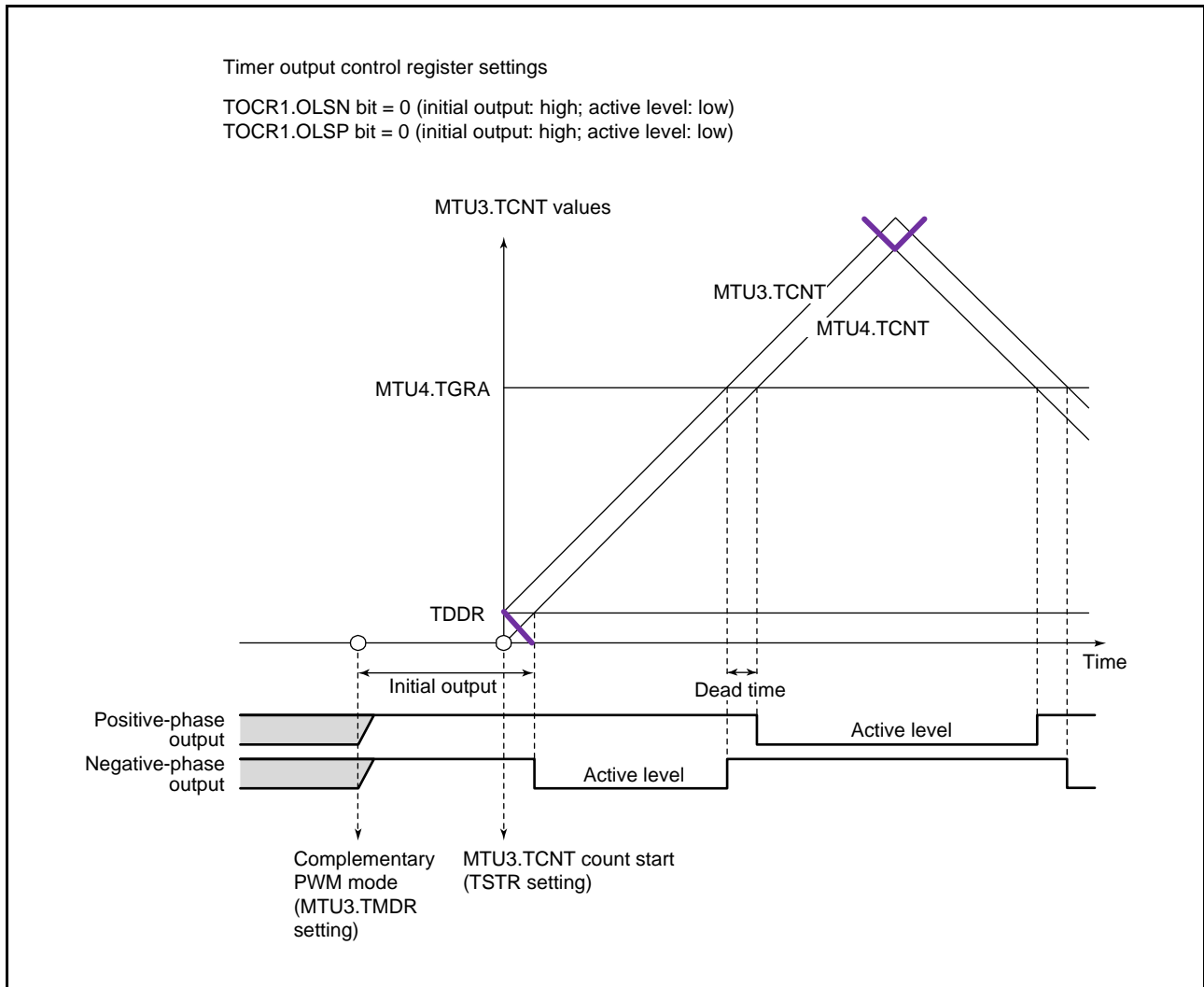


Figure 20.44 Example of Initial Output in Complementary PWM Mode (1)

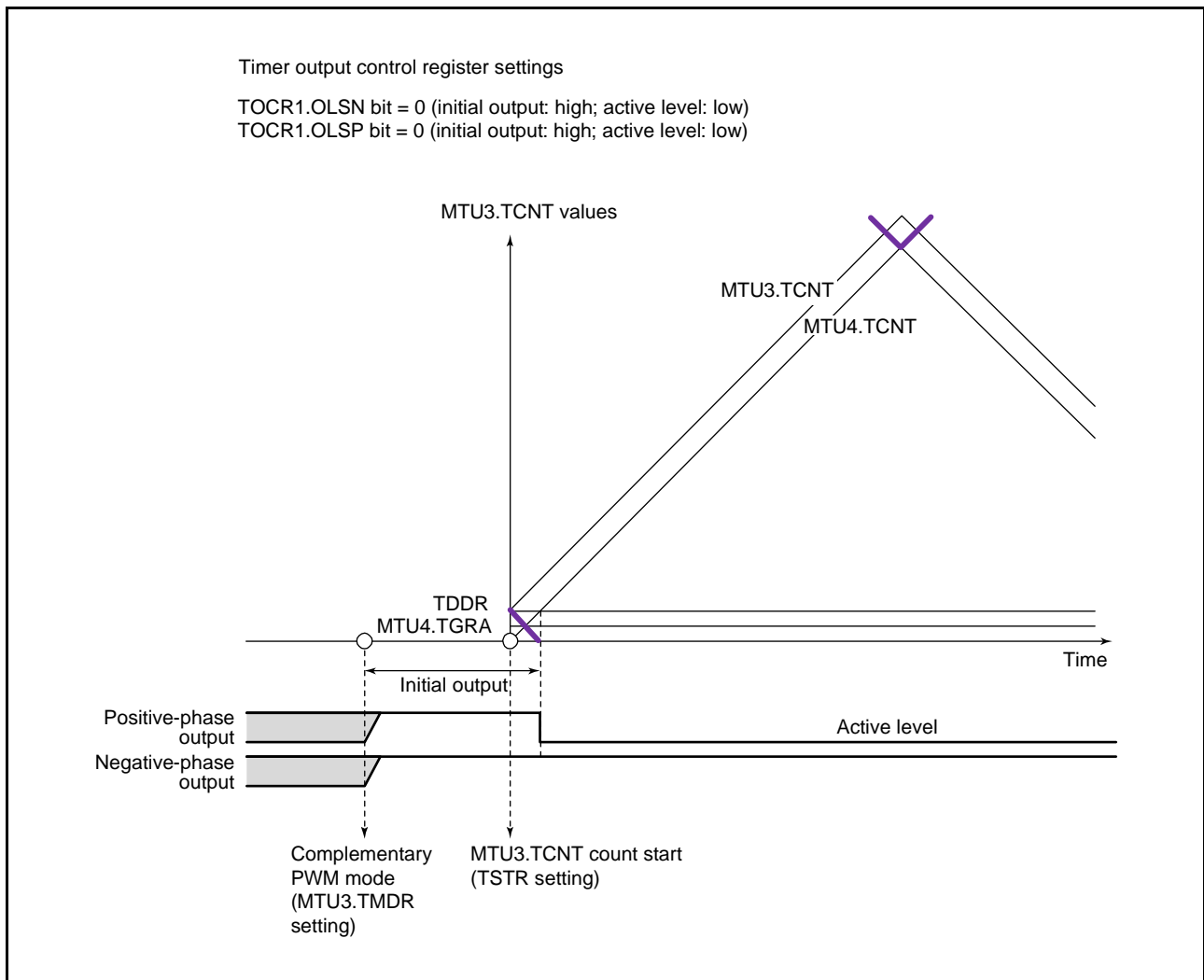


Figure 20.45 Example of Initial Output in Complementary PWM Mode (2)

(j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, six phases (three positive and three negative) of PWM waveforms can be output. Dead time can be set for PWM waveforms to be output.

A PWM waveform is generated by output of the level selected in the timer output control register in the event of a compare match between a counter and a compare register. While TCNTS is counting, the compare register and temporary register values are simultaneously compared to generate consecutive PWM waveforms from 0 to 100% duty cycle. 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 20.46 to Figure 20.48 show examples of waveform generation in complementary PWM mode.

The positive-phase and negative-phase turn-off timing is generated by a compare match with the counter indicated by a solid line, and the turn-on timing is generated by a compare match with the counter indicated by a dotted line, which operates with a delay equal to the dead time behind the counter indicated by a solid line. In the T1 period, compare match a that turns off the negative phase has the highest priority, and compare matches before a are ignored. In the T2 period, compare match c that turns off the positive phase has the highest priority, and compare matches before c are ignored.

In most cases, compare matches occur in the order a → b → c → d (or c → d → a' → b') as shown in Figure 20.46. 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 20.47, 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 20.48, turning off the negative phase has priority due to the occurrence of compare match a' (negative-phase off timing) before compare match d (negative-phase on timing). As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare matches at turn-off timings take precedence, and turn-on timing compare matches that occur before a turn-off timing compare match are ignored.

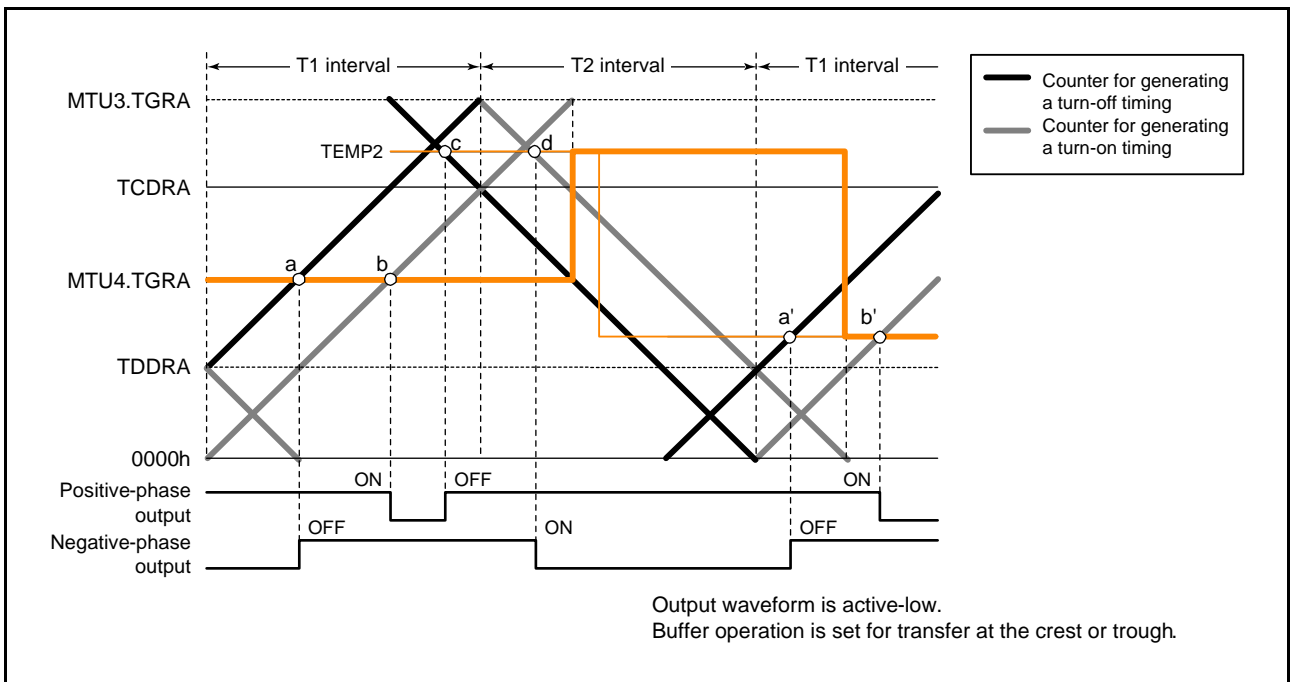


Figure 20.46 Example of Waveform Output in Complementary PWM Mode (1)



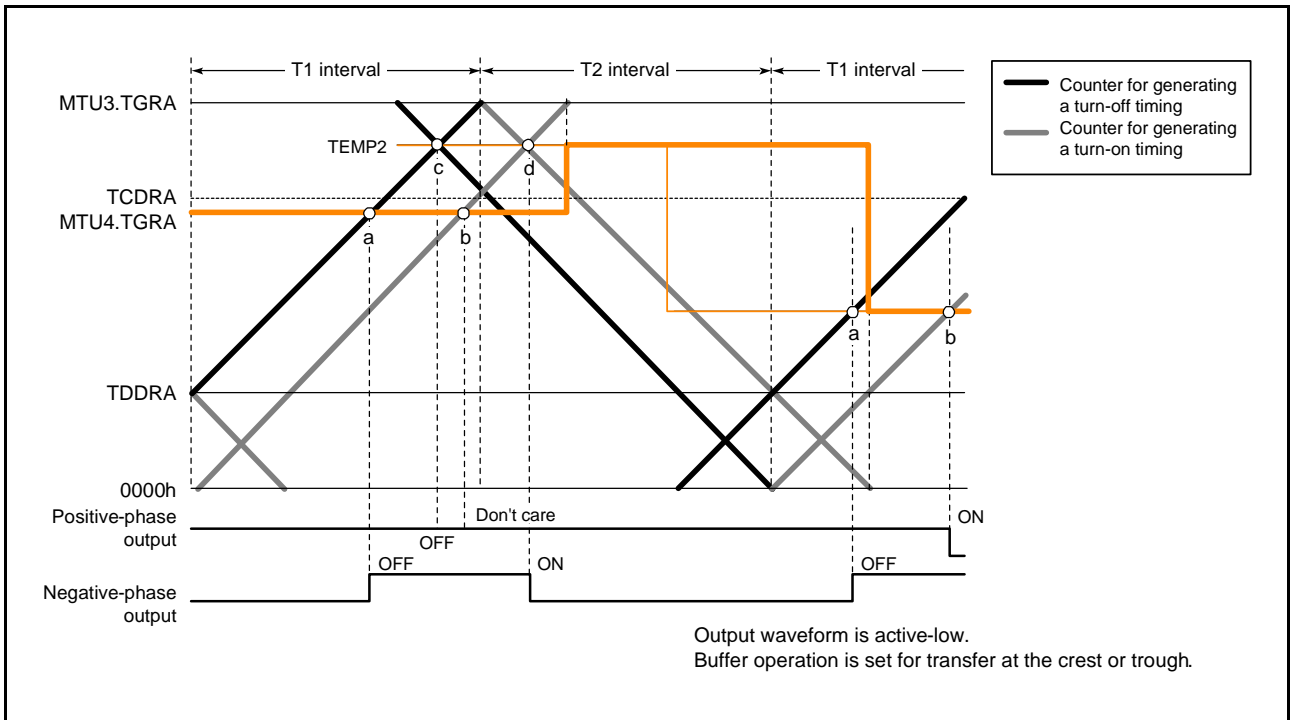


Figure 20.47 Example of Waveform Output in Complementary PWM Mode (2)

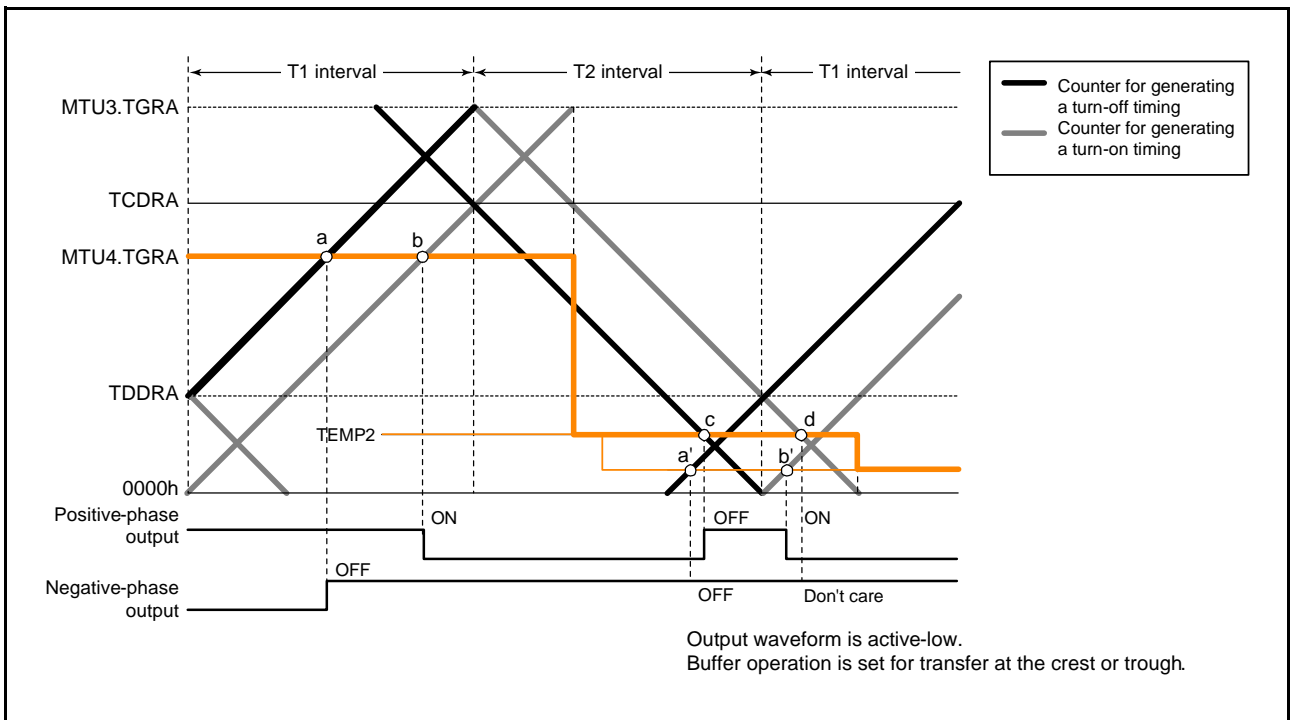


Figure 20.48 Example of Waveform Output in Complementary PWM Mode (3)

(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 20.49 to Figure 20.53 show output examples.

A 100% duty cycle waveform is output when the data 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 data 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.

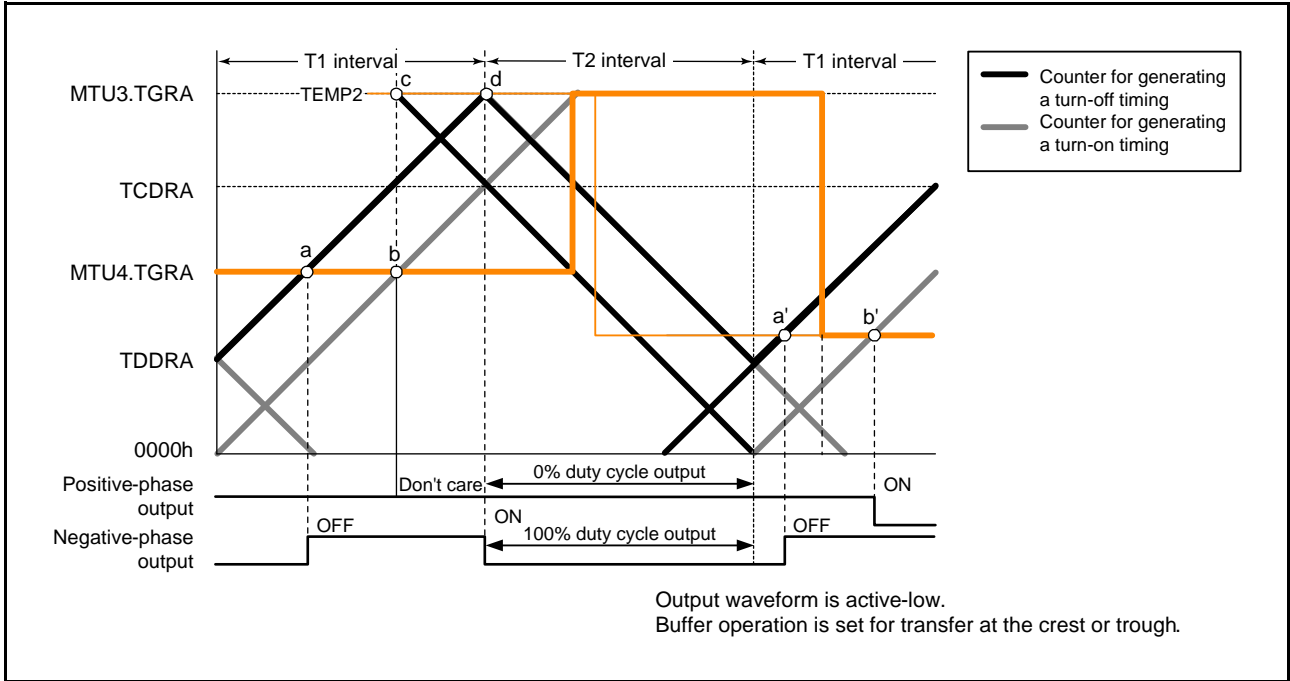


Figure 20.49 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1)

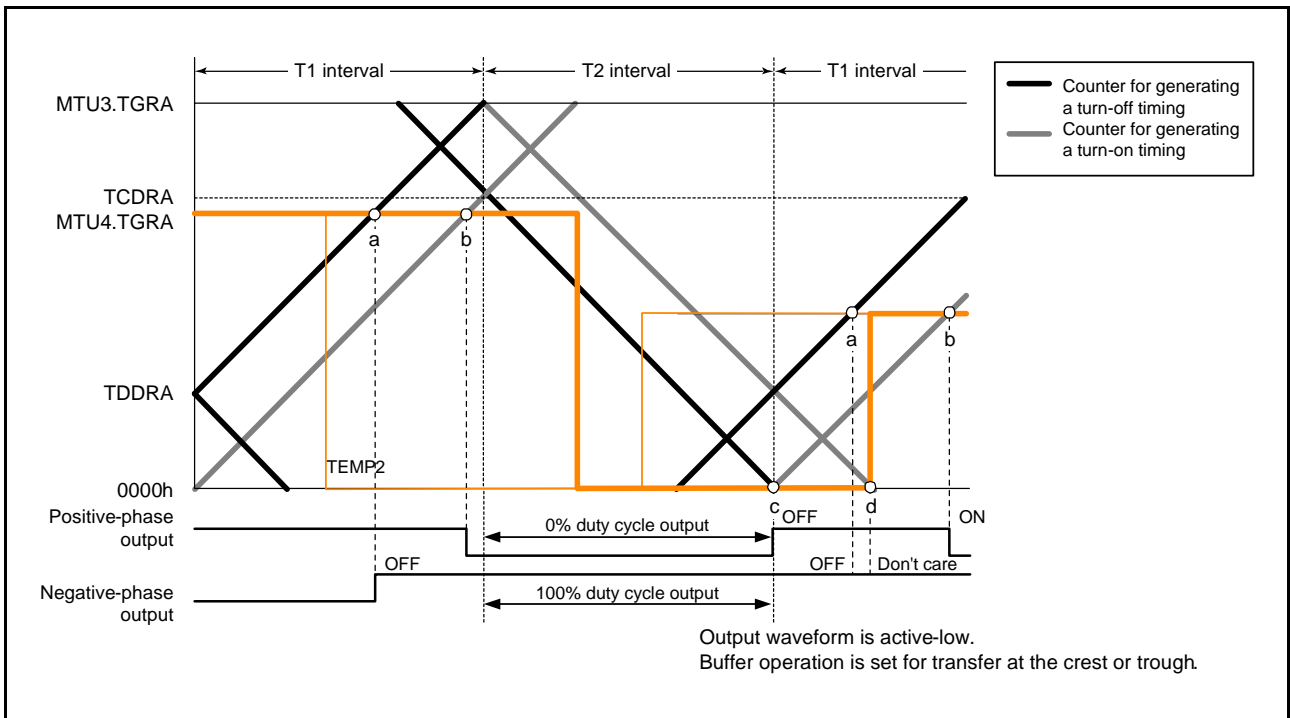


Figure 20.50 Example of 0% and 100% Waveform Output in Complementary PWM Mode (2)

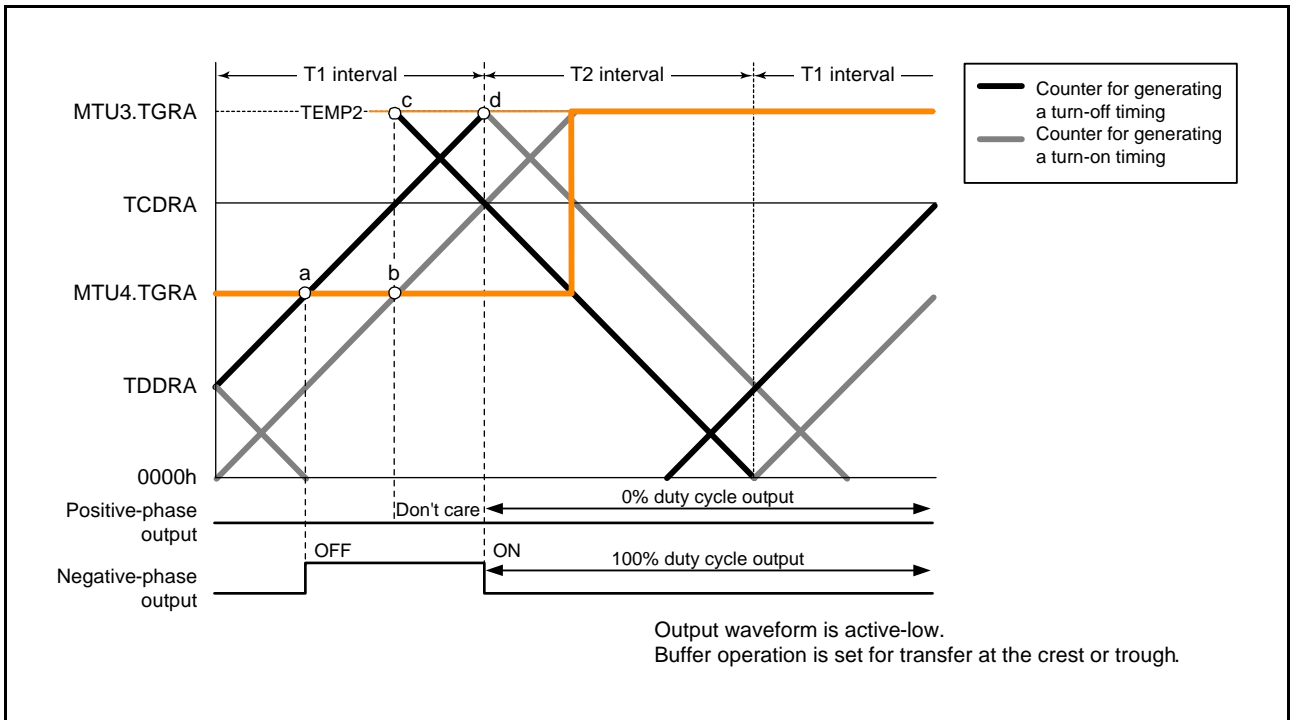


Figure 20.51 Example of 0% and 100% Waveform Output in Complementary PWM Mode (3)

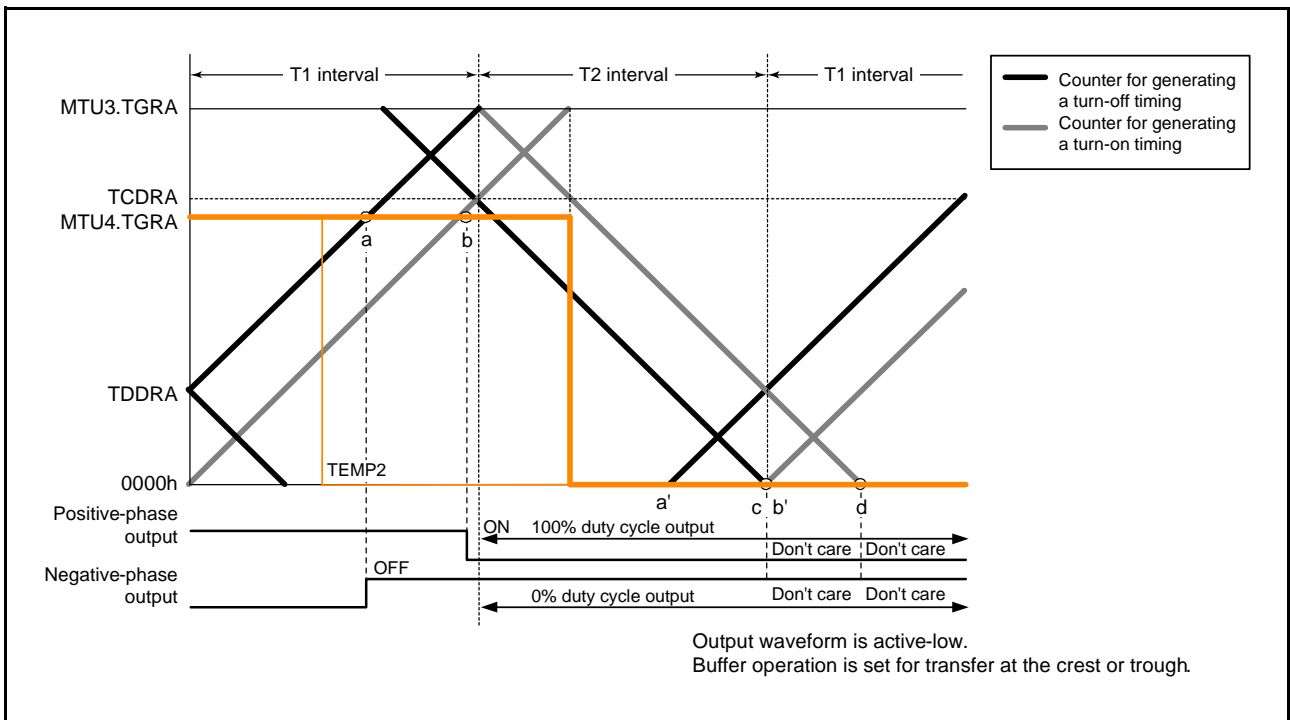


Figure 20.52 Example of 0% and 100% Waveform Output in Complementary PWM Mode (4)

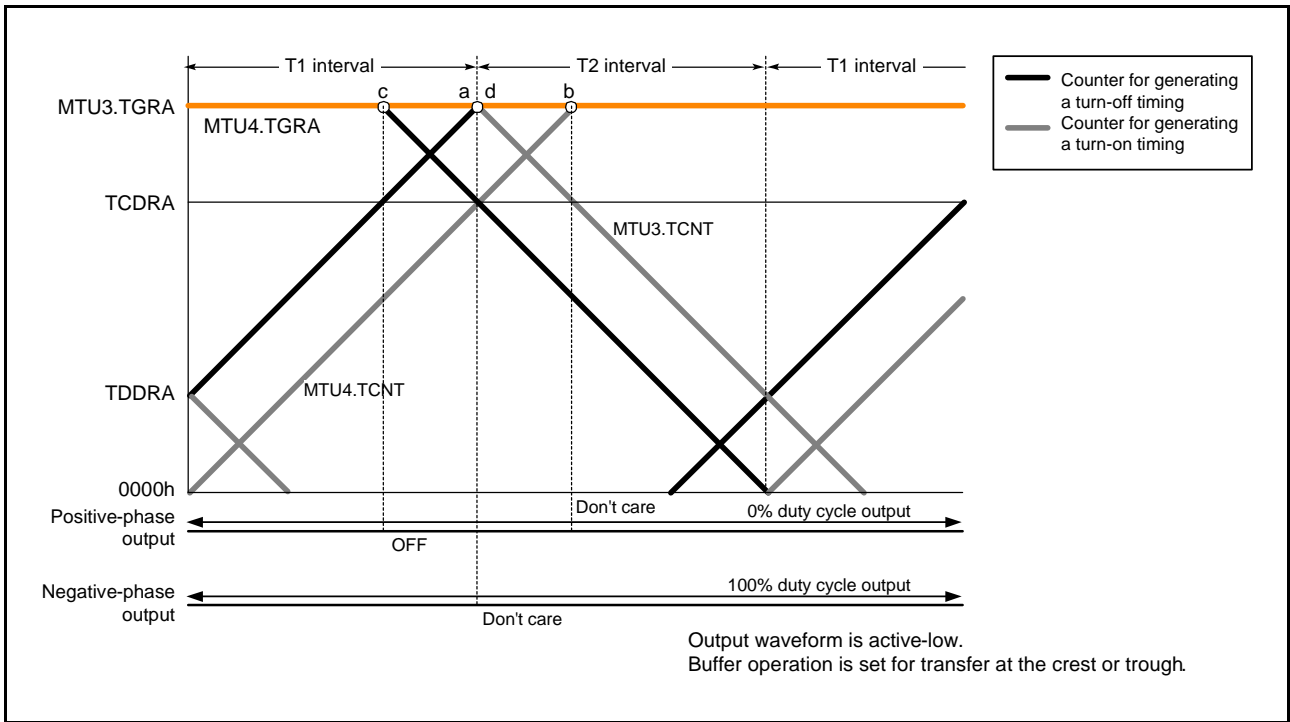


Figure 20.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5)

(l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output from the PWM output pin in synchronization with the PWM period can be enabled by setting the TOCR1.PSYE bit to 1. An example of a toggle output waveform is shown in Figure 20.54.

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 a high level.

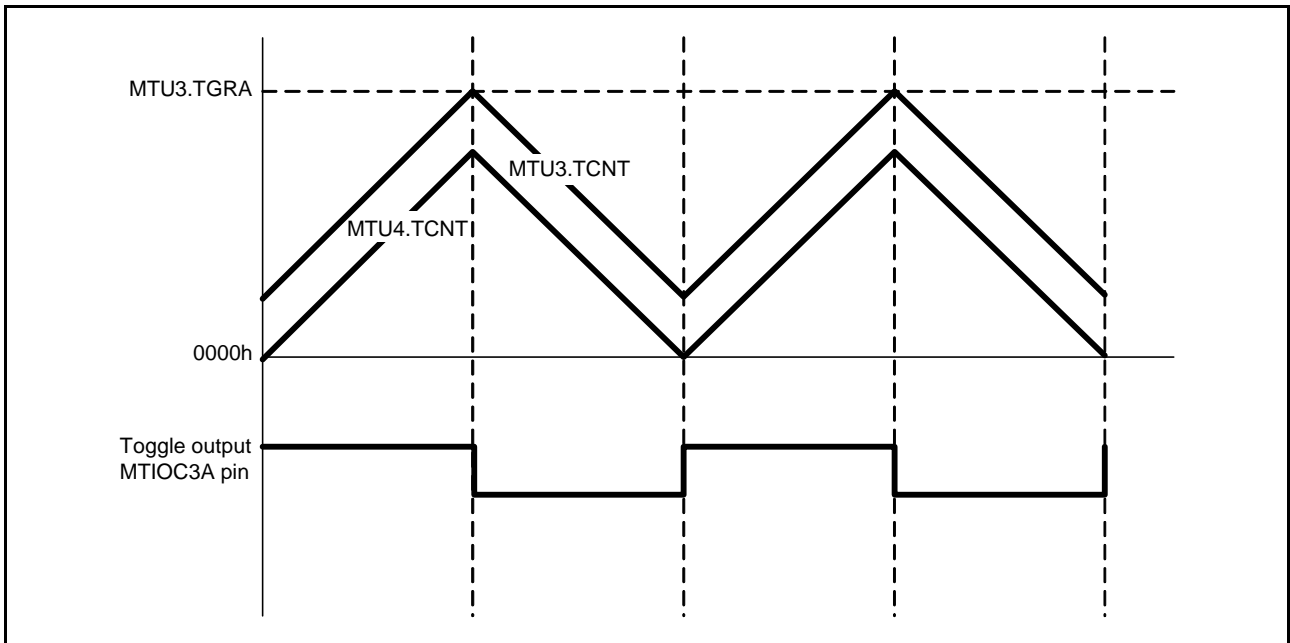


Figure 20.54 Example of Toggle Output Waveform Synchronized with PWM Output

(m) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTS can be cleared by another channel source when a mode for synchronization with another channel is specified by TSYR and synchronous clearing is selected with the MTU3.TCR.CCLR[2:0] bits.

Figure 20.55 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

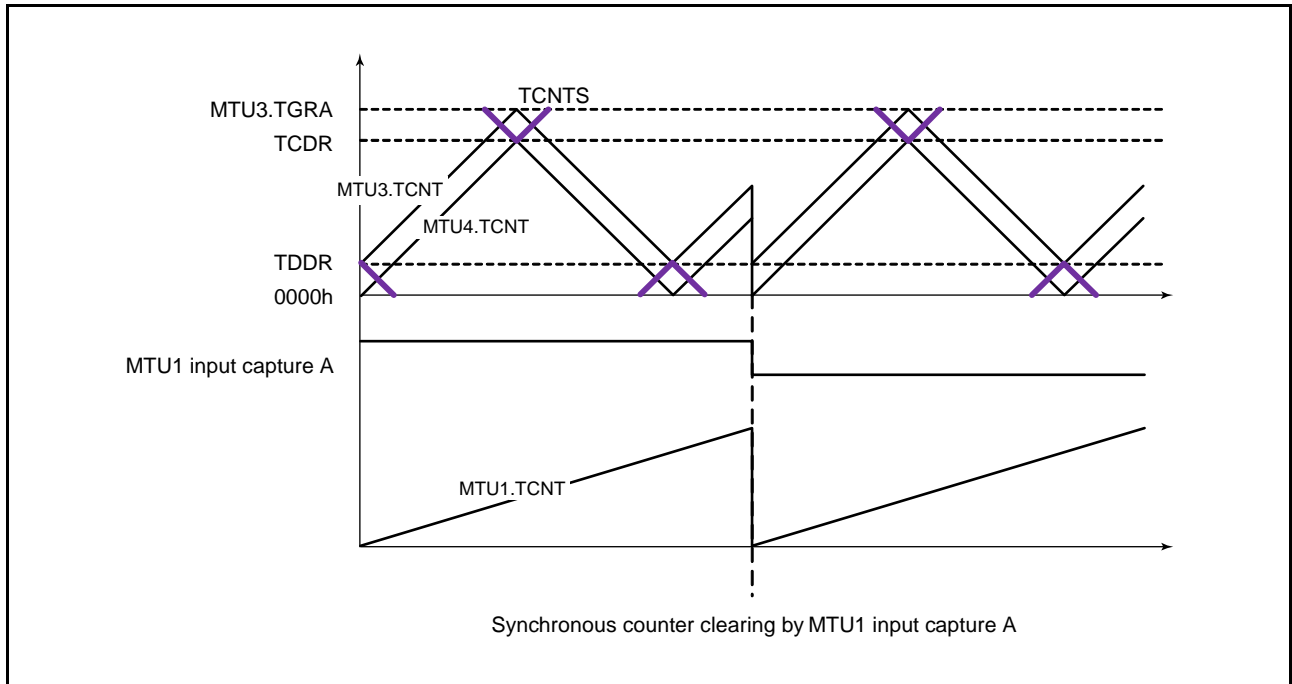


Figure 20.55 Counter Clearing Synchronized with Another Channel

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the TWCR.WRE bit to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval (Tb2 interval) at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression through setting TWCR.WRE bit to 1 is applicable only when synchronous clearing occurs in the Tb2 interval as indicated by (10) or (11) in Figure 20.56. When synchronous clearing occurs outside that interval, the initial value specified by the TOCR1.OLSN bit and TOCR1.OLSP bit is output. Even in the Tb2 interval, if synchronous clearing occurs in the initial output period (indicated by (1) in Figure 20.56) 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.

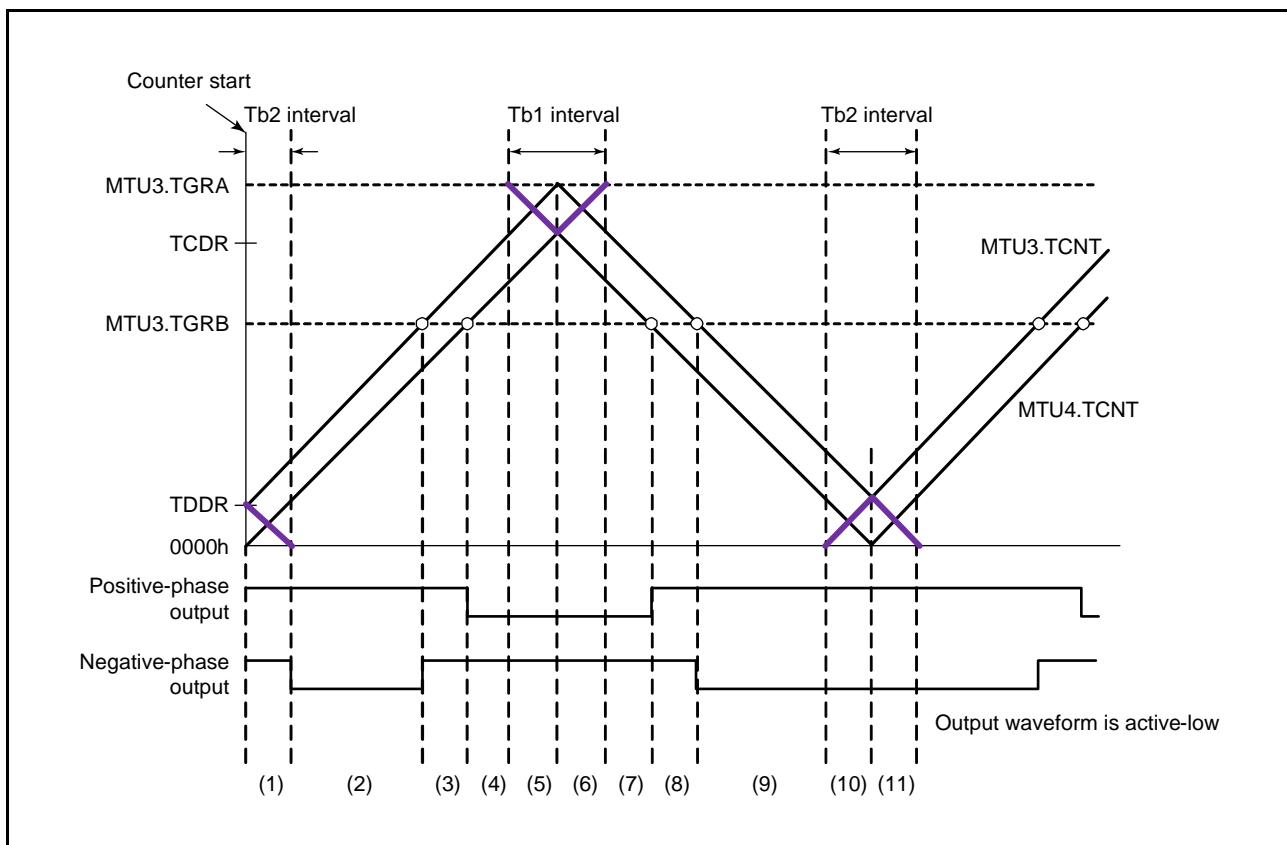
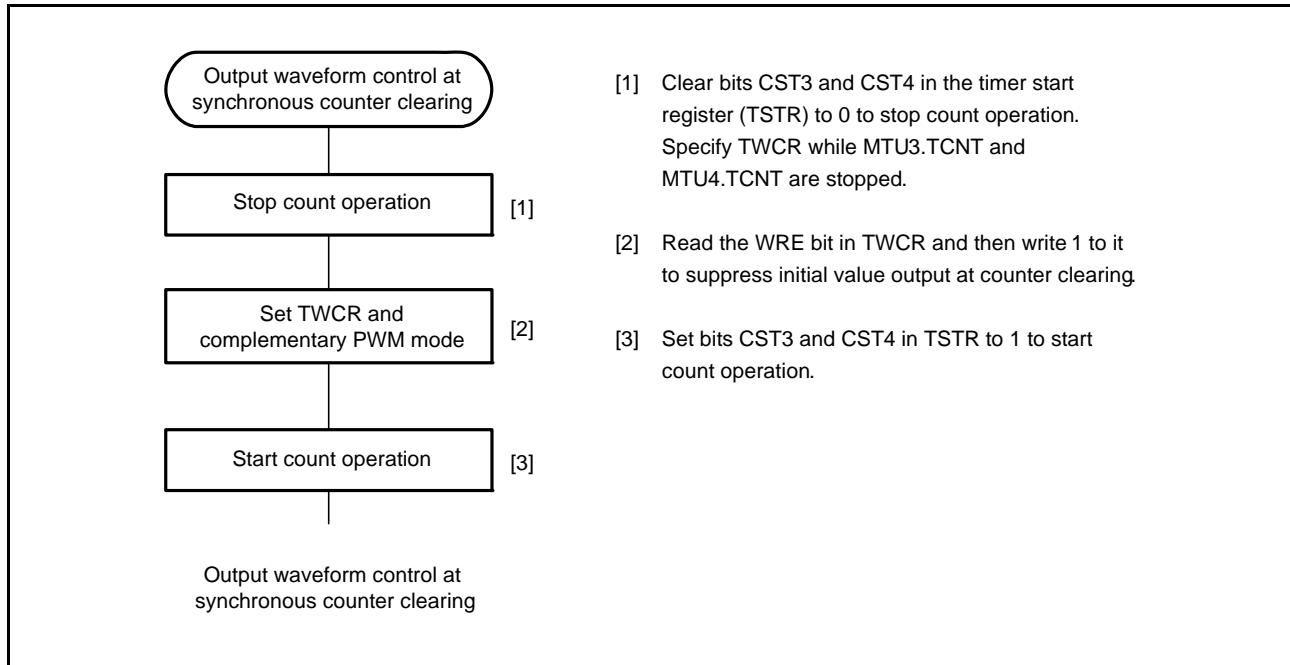


Figure 20.56 Timing for Synchronous Counter Clearing

- Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

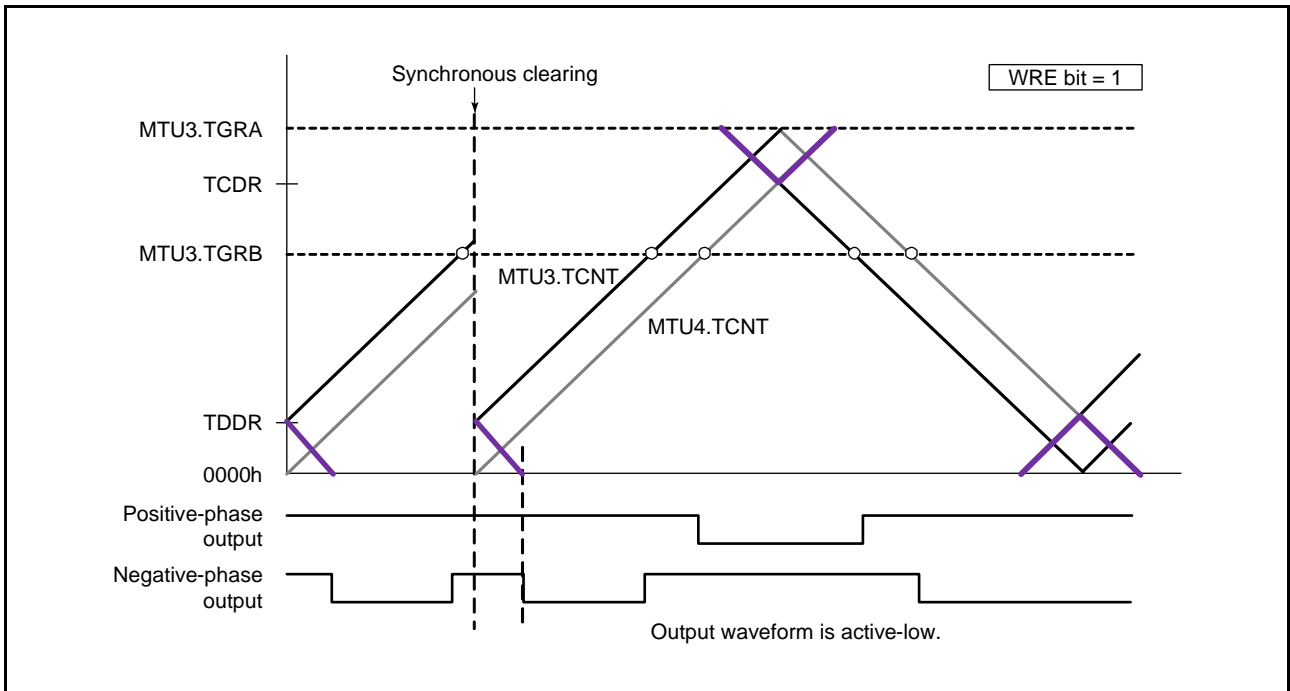
An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in Figure 20.57.



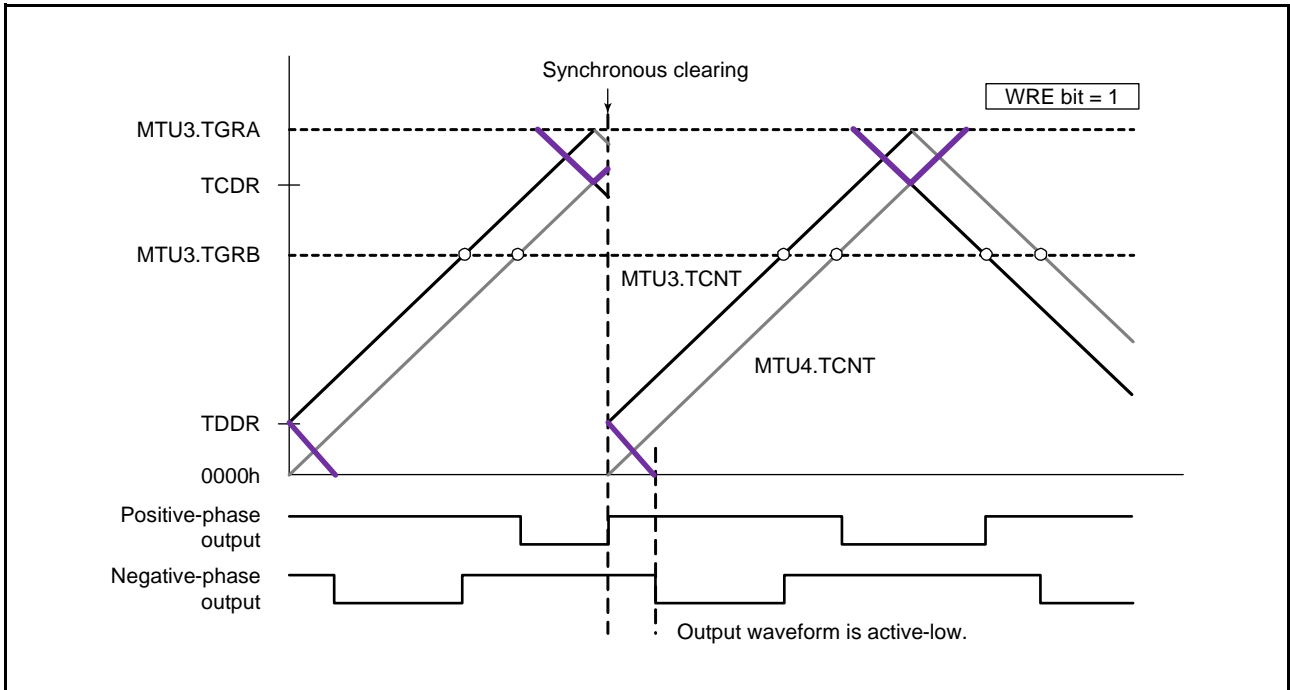
**Figure 20.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode**

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 20.58 to Figure 20.61 show examples of output waveform control in which the MTU operates in complementary PWM mode and synchronous counter clearing is generated while the TWCR.WRE bit is set to 1. In the examples shown in Figure 20.58 to Figure 20.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 20.56, respectively.

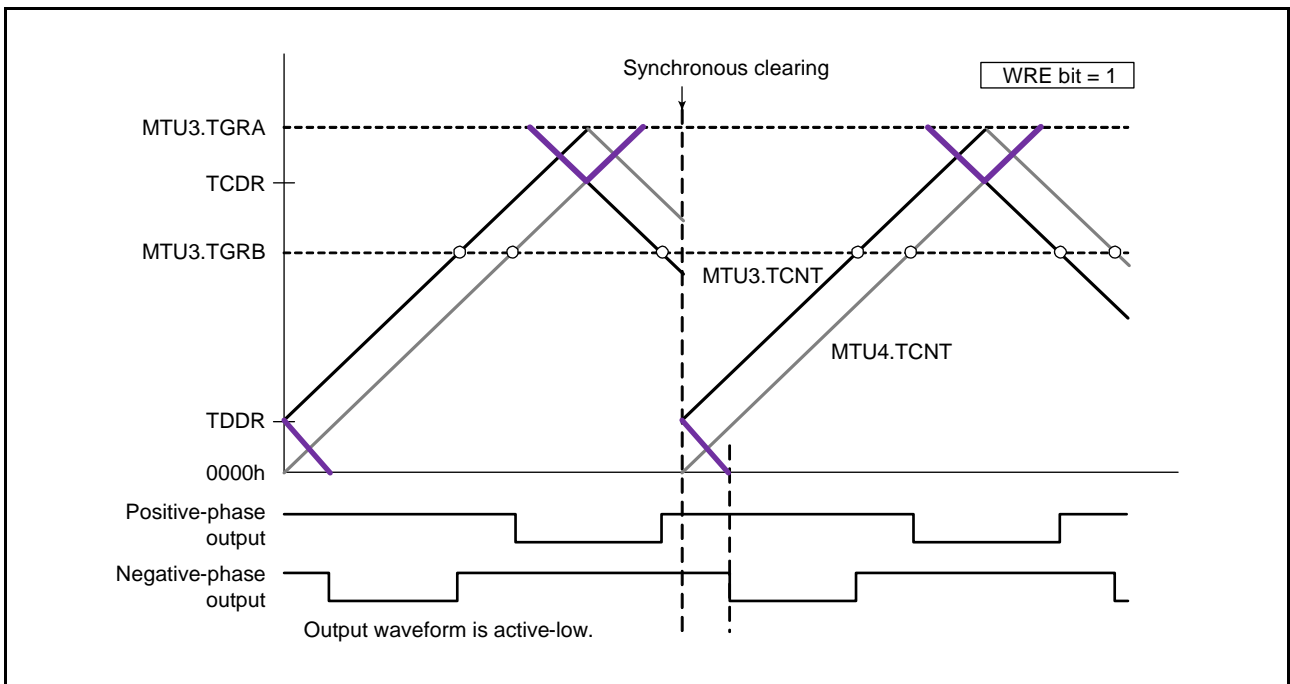


**Figure 20.58 Example of Synchronous Clearing in Dead Time during Up-Counting**  
 (Timing (3) in Figure 20.56; TWCR.WRE Bit is 1)

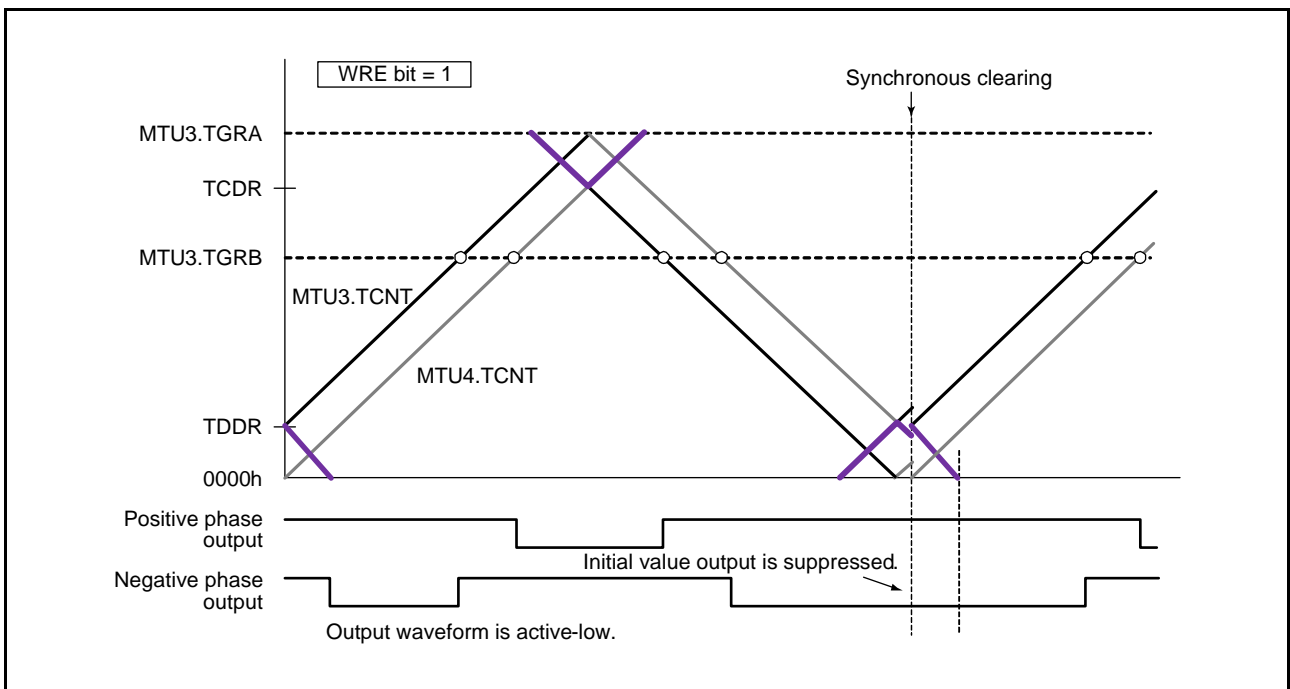


**Figure 20.59 Example of Synchronous Clearing in Interval Tb at Crest**  
 (Timing (6) in Figure 20.56; TWCR.WRE Bit is 1)





**Figure 20.60** Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 20.56; TWCR.WRE Bit is 1)



**Figure 20.61** Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 20.56; TWCR.WRE Bit 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 TWCR.CCE bit is set.

Figure 20.62 shows 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 TOCR1.PSYE bit to 1.

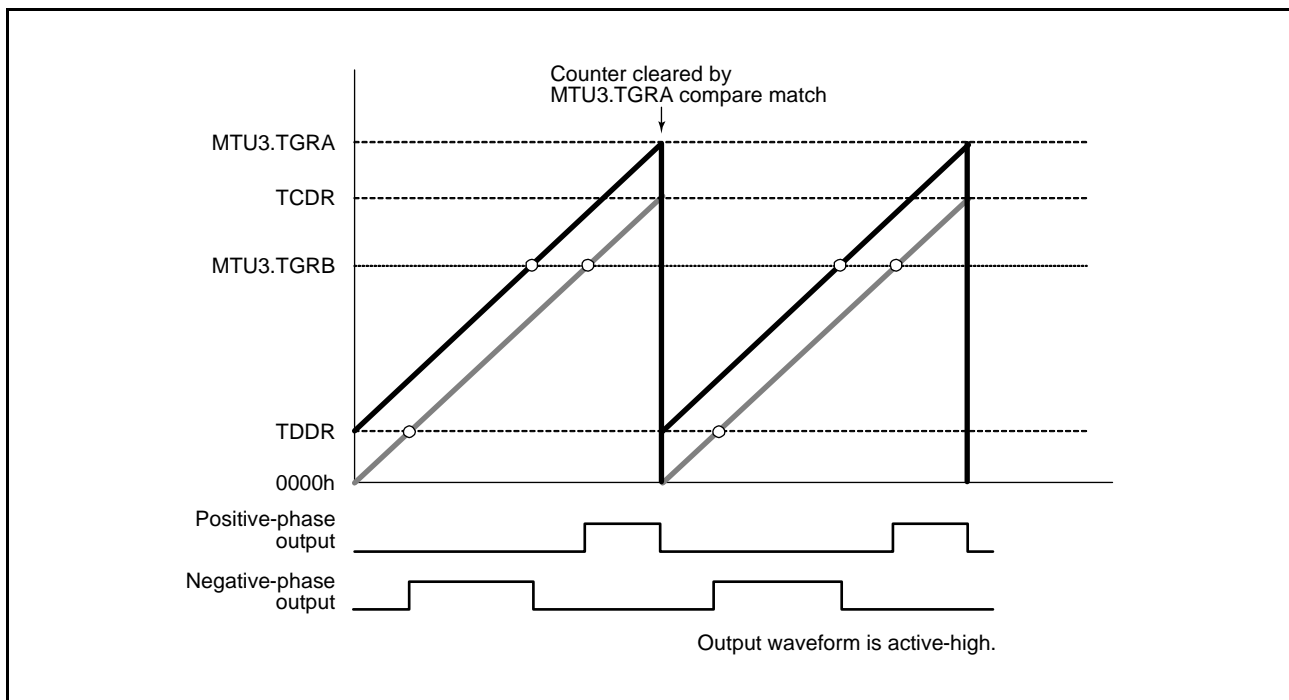


Figure 20.62 Example of Counter Clearing Operation by MTU3.TGRA Compare Match

(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 20.63 to Figure 20.66 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 TGCR.FB bit to 0. In this case, the external signals indicating the magnetic pole position should be input to timer input pins MTIOC0A, MTIOC0B, and MTIOC0C in MTU0. When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.

When the TGCR.FB bit is 1, the output on/off state is switched when the TGCR.UF bit, TGCR.VF bit, or TGCR.WF bit is cleared to 0 or set to 1.

The driving waveforms are output from the 6-phase PWM output pins for complementary PWM mode.

With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the TGCR.N bit or TGCR.P bit to 1. When the TGCR.N bit or TGCR.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 TOCR1.OLSN bit and TOCR1.OLSP bit regardless of the setting of the TGCR.N bit and TGCR.P bit.

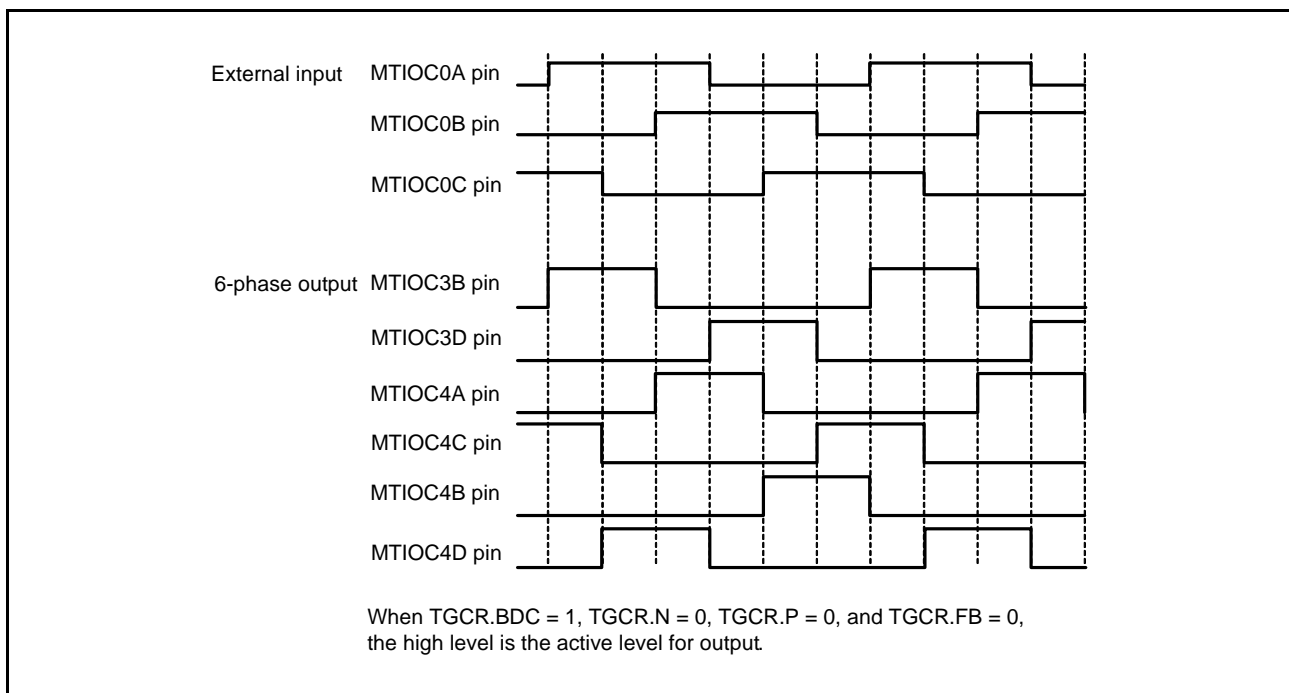


Figure 20.63 Example of Output Phase Switching by External Input (1)

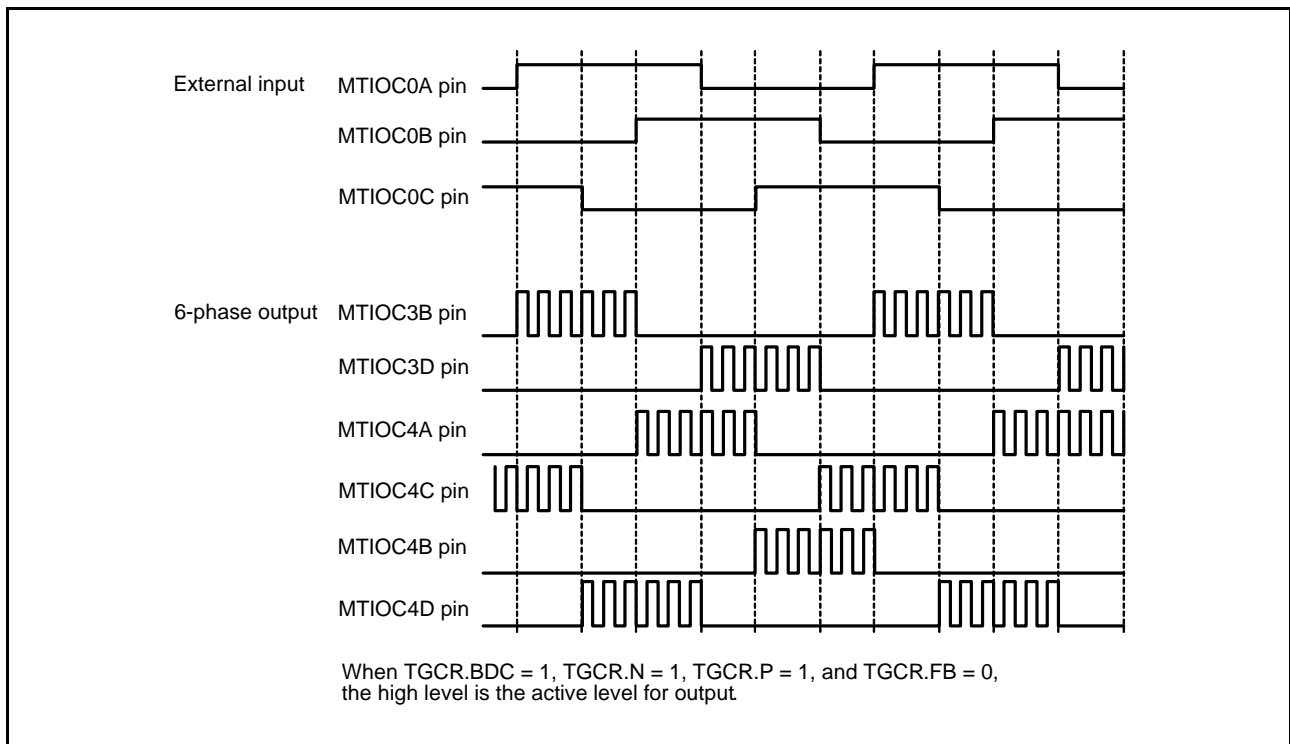


Figure 20.64 Example of Output Phase Switching by External Input (2)

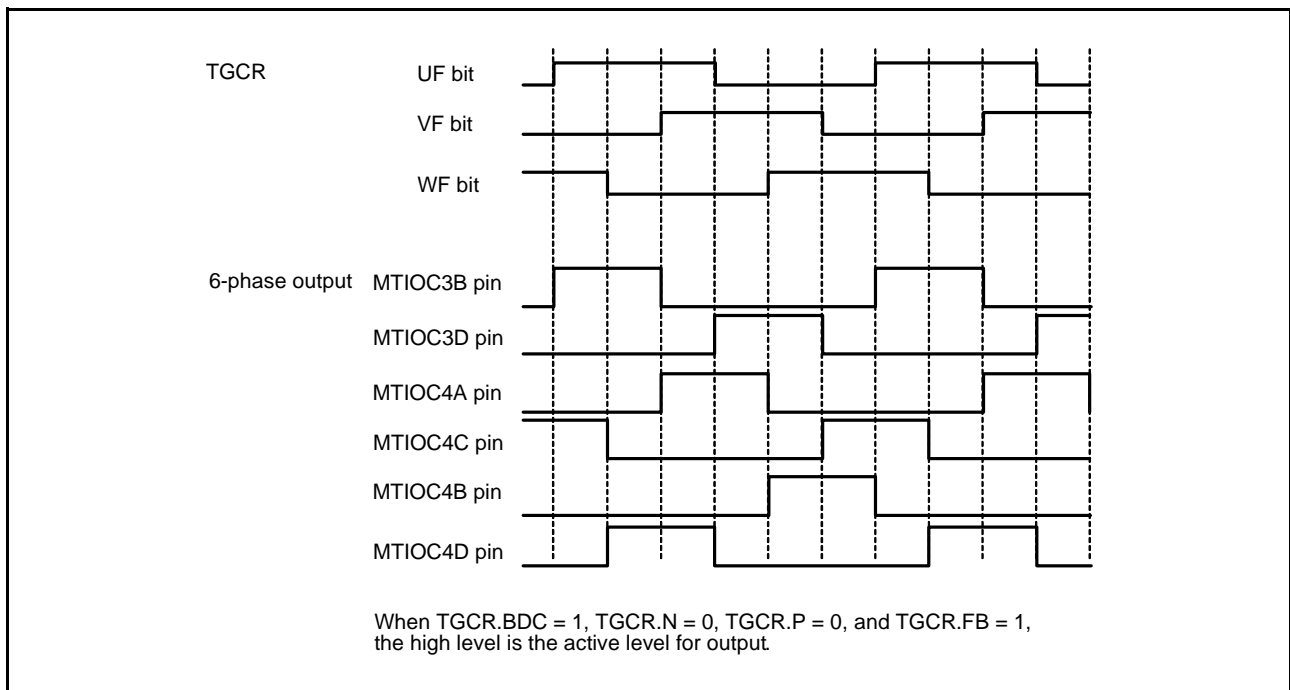
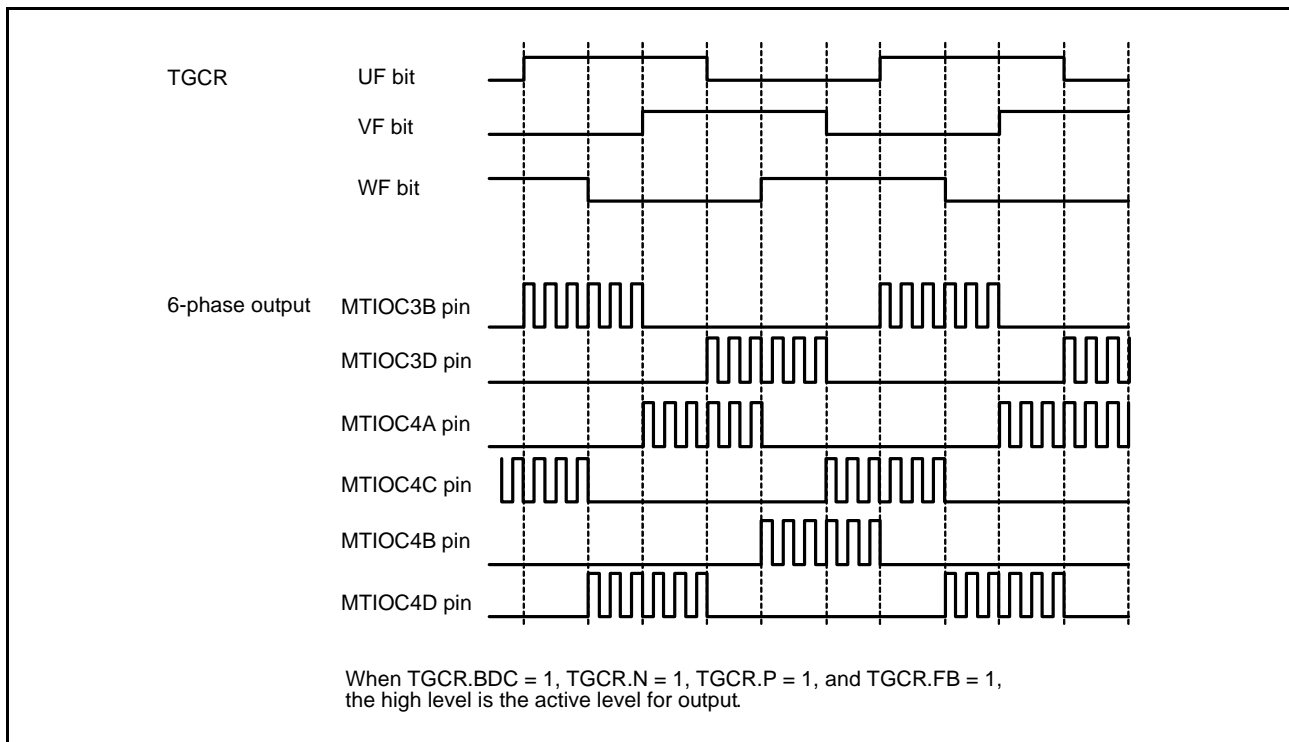


Figure 20.65 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)



**Figure 20.66 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 TIER.TTGE bit to 1. To issue an A/D converter start request at an MTU4.TCNT underflow (trough), set the MTU4.TIER.TTGE2 bit 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 setting 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 set 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 20.3.9, A/D Converter Start Request Delaying Function.

The timer interrupt skipping set 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 TITCR.T3AEN and TITCR.T4VEN bits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Operation Setting Procedure

Figure 20.67 shows an example of the interrupt skipping operation setting procedure. Figure 20.68 shows the periods during which interrupt skipping count can be changed.

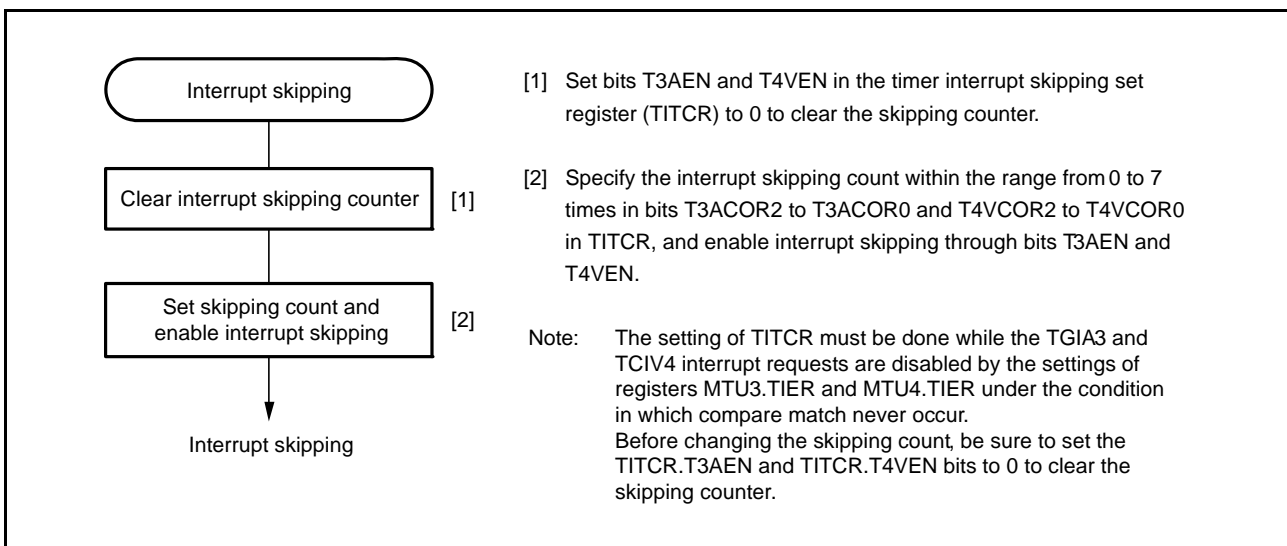


Figure 20.67 Example of Interrupt Skipping Operation Setting Procedure

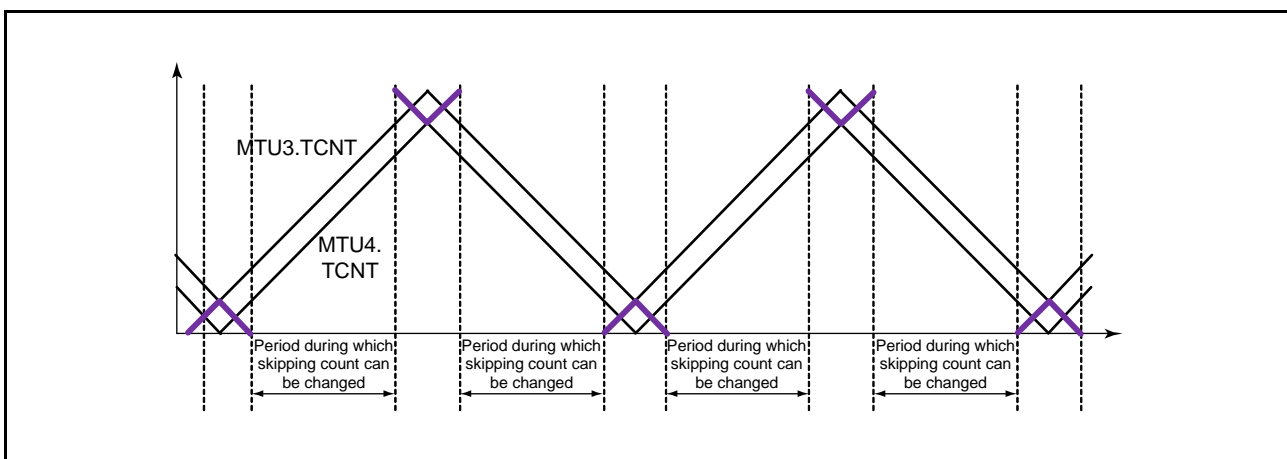


Figure 20.68 Periods during which Interrupt Skipping Count Can be Changed

(b) Example of Interrupt Skipping Operation

Figure 20.69 shows an example of MTU3.TGIA interrupt skipping in which the interrupt skipping count is set to three by the TITCR.T3ACOR bit and the TITCR.T3AEN bit is set to 1.

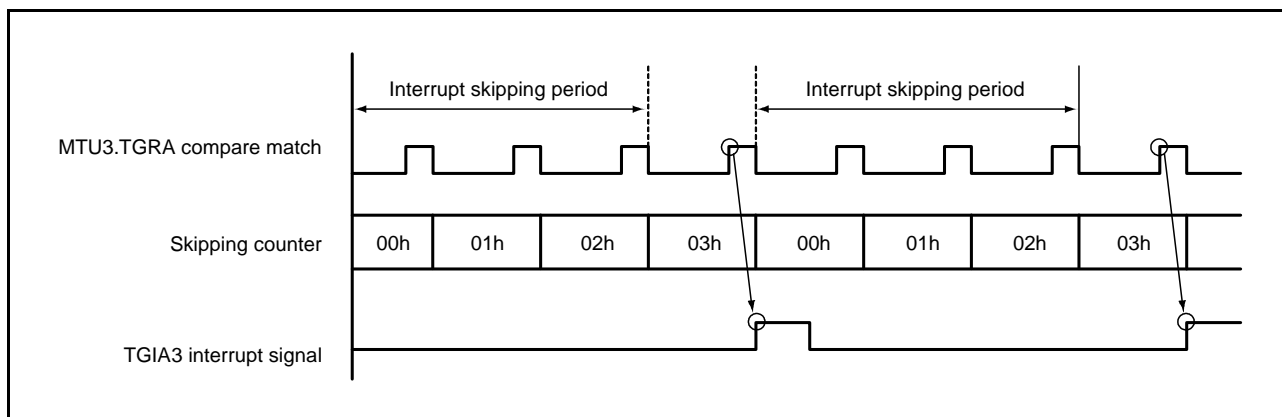


Figure 20.69 Example of Interrupt Skipping Operation

(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 20.70 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 20.71 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b). While this setting is valid, if data is written to the buffer register within the buffer transfer-enabled period, the data is transferred immediately from the buffer register to the temporary register. If data is written to the buffer register outside the buffer transfer-enabled period, the data is transferred from the buffer register to the temporary register at the timing when the next buffer transfer-enabled period starts.

Note that the buffer transfer-enabled period depends on the TITCR.T3AEN bit and TITCR.T4VEN bit settings. Figure 20.72 shows the relationship between the TITCR.T3AEN bit and TITCR.T4VEN bit settings 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 setting bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the TBTER.BTE1 bit to 0). If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.

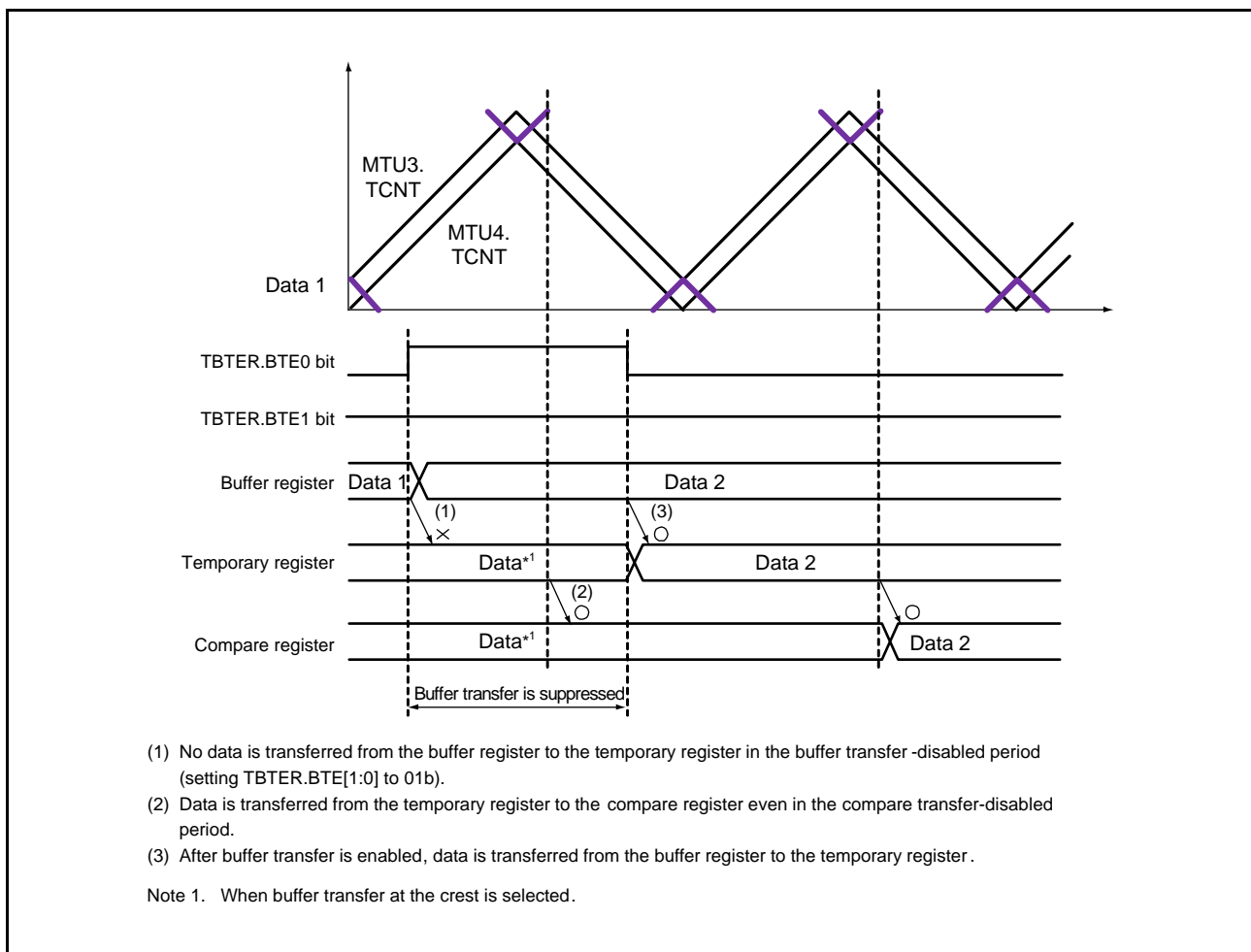
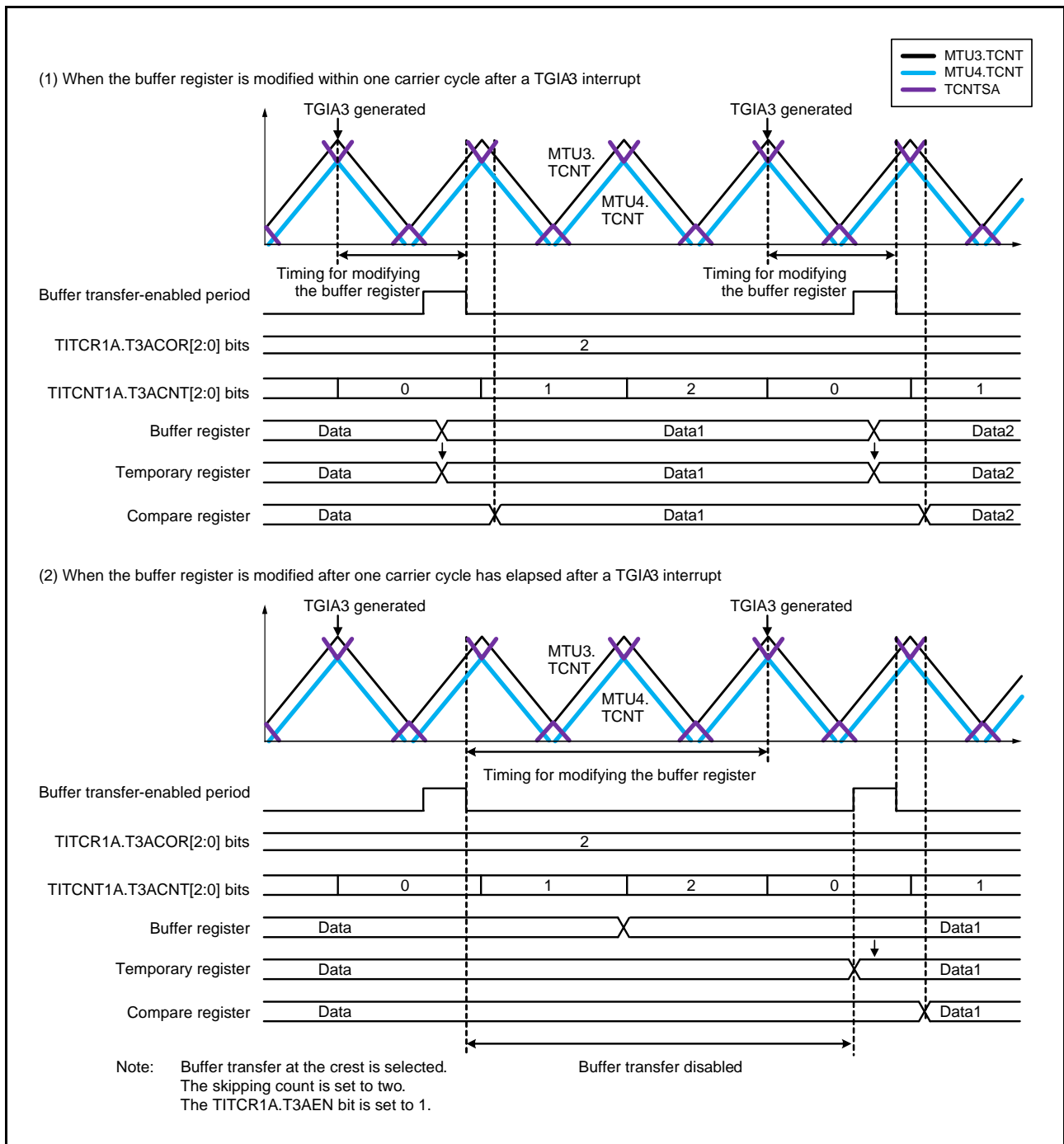
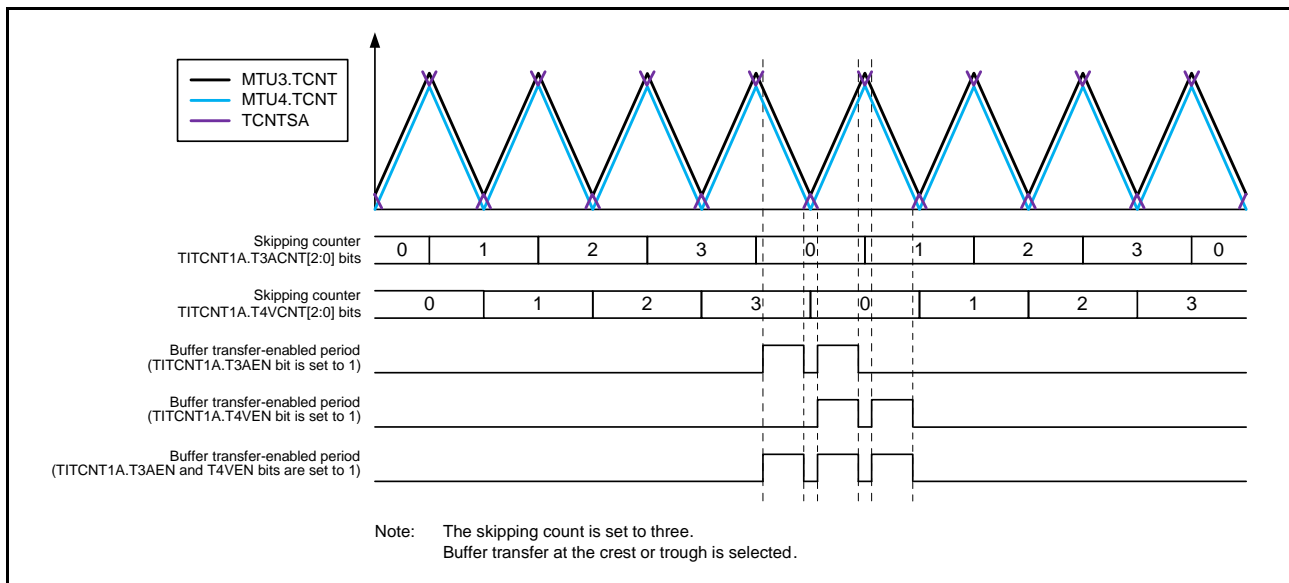


Figure 20.70 Example of Operation When Buffer Transfer is Disabled (TBTER.BTE[1:0] = 01b)





**Figure 20.71** Example of Operation When Buffer Transfer is Linked with Interrupt Skipping (TBTER.BTE[1:0] = 10b)



**Figure 20.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period**

**(4) Complementary PWM Mode Output Protection Functions**

The MTU provides the following protection functions for complementary PWM mode output.

**(a) Register and Counter Miswrite Prevention Function**

Access from the CPU to the mode registers, control registers, compare registers can be enabled or disabled by setting the TRWER.RWE bit. 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**

The PWM output pins of MTU0, MTU3, and MTU4 can be set to the high-impedance state automatically. Refer to section 21, Port Output Enable 2 (POE2a), for details.

### 20.3.9 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 TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and ITB4VE bit.

#### (1) Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 20.73 shows an example of procedure for specifying the A/D converter start request delaying function.

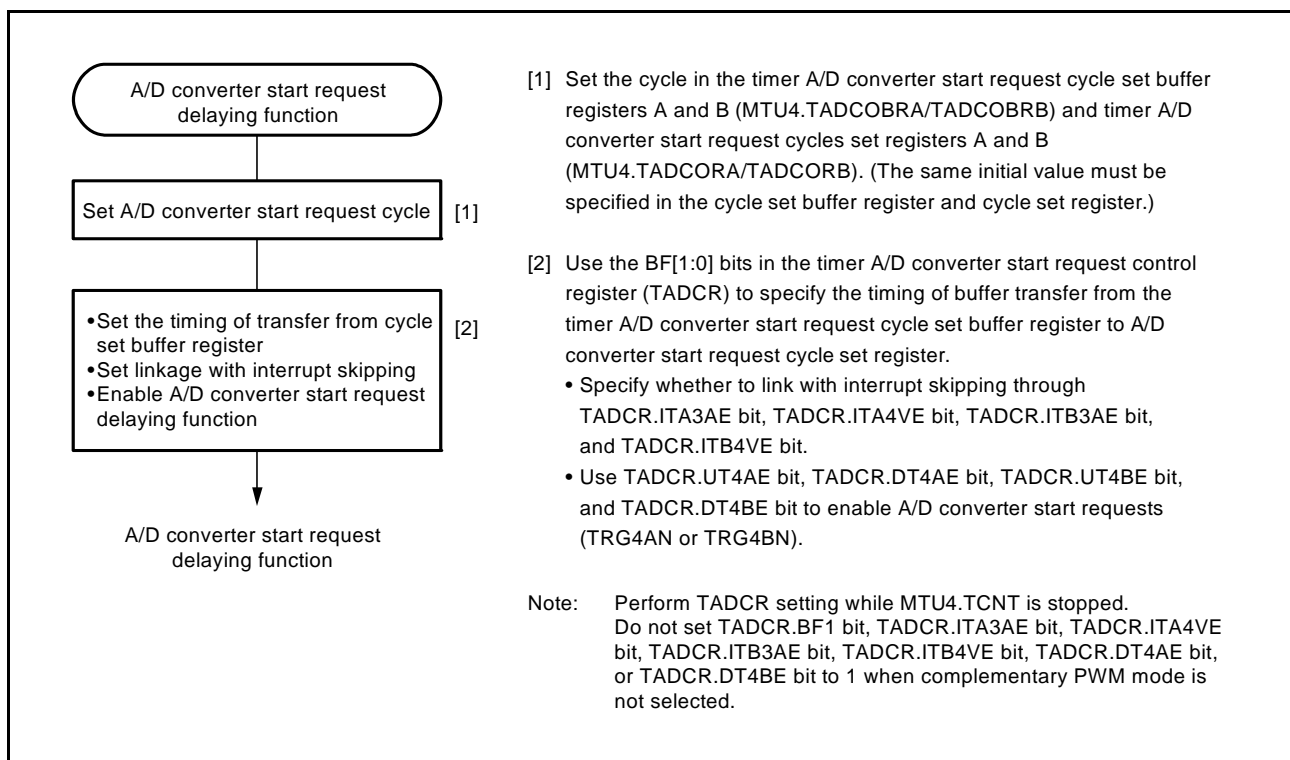
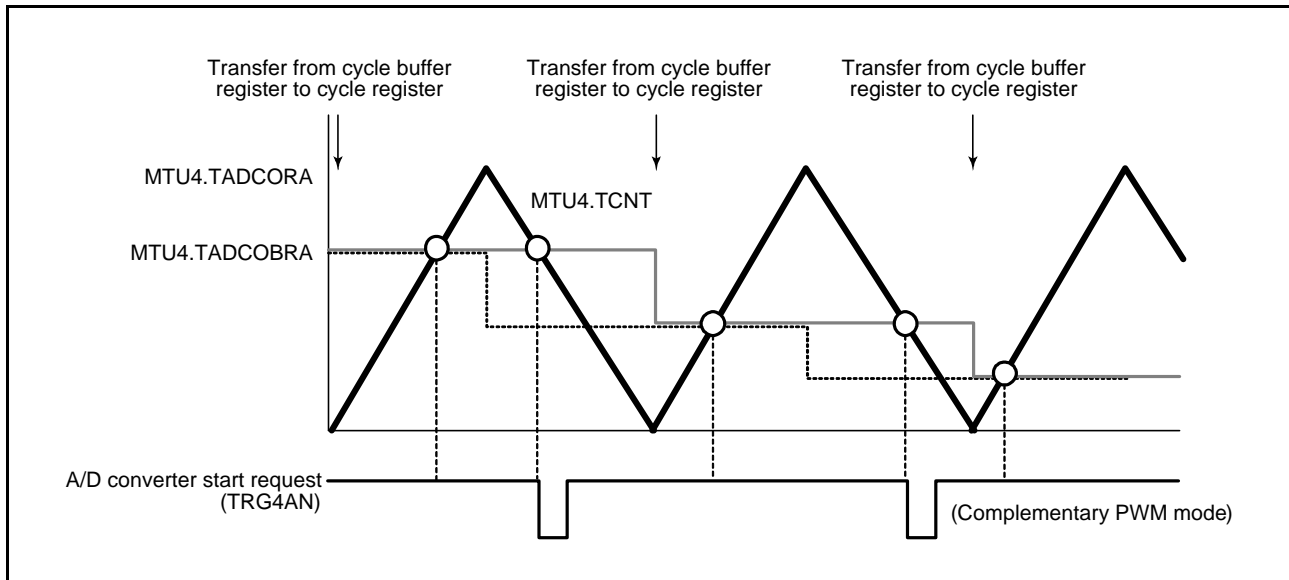


Figure 20.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

## (2) Basic Example of A/D Converter Start Request Delaying Function Operation

Figure 20.74 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 20.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation**

## (3) Buffer Transfer

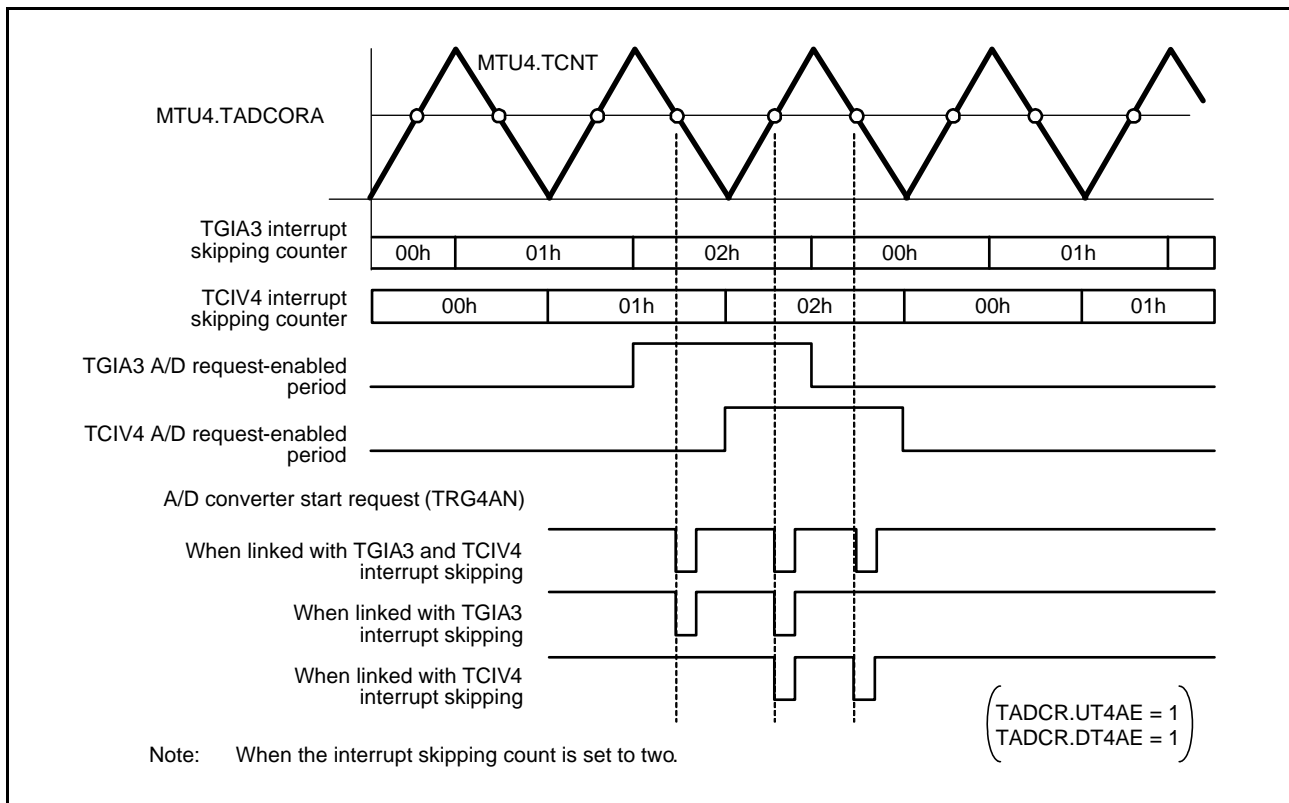
The data in the timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB) is updated by writing data to the timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the MTU4.TADCR.BF[1:0] bits.

## (4) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

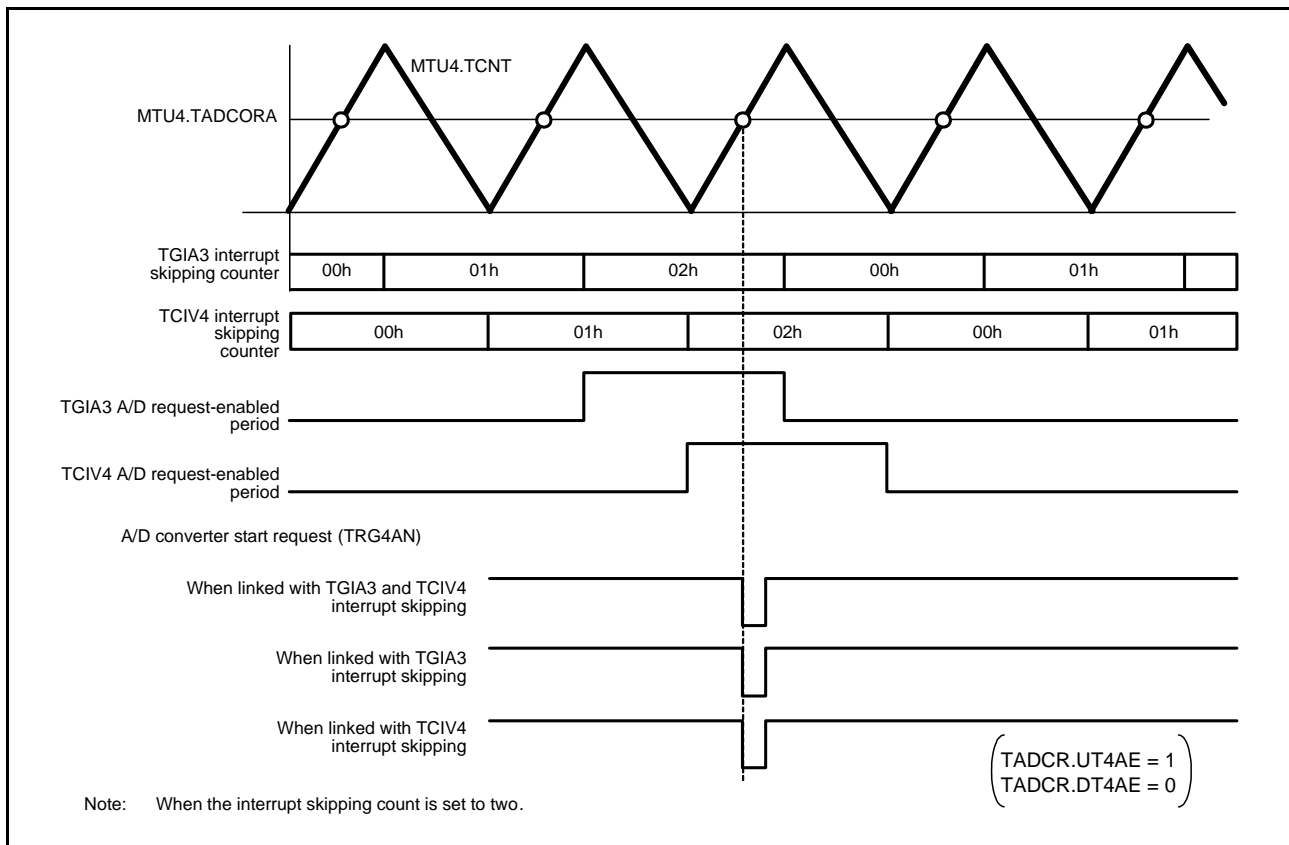
A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and TADCR.ITB4VE bit. Figure 20.75 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and down-counting and A/D converter start requests are linked with interrupt skipping.

Figure 20.76 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 TITCR.T3AEN bit and TITCR.T4VEN bit are cleared to 0 or the skipping count setting 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 TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and TADCR.ITB4VE bit to 0). Note that TRG4ABN (TRG4AN or TRG4BN) is output as the A/D converter start request signal in this case.



**Figure 20.75 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 20.76 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)**

### 20.3.10 External Pulse Width Measurement

Up to three external pulse widths can be measured in MTU5.

When the IOC[4:0] bits in MTU5.TIORU, TIORV, and TIORW are set for pulse width measurement, the pulse width of the signal input to the MTIC5U, MTIC5V, and MTIC5W pins is measured. TCNTU, TCNTV, and TCNTW count up while the level specified by the IOC[4:0] bits is input.

Figure 20.77 shows an example of setting external pulse width measurement, and Figure 20.78 an example of external pulse width measurement.

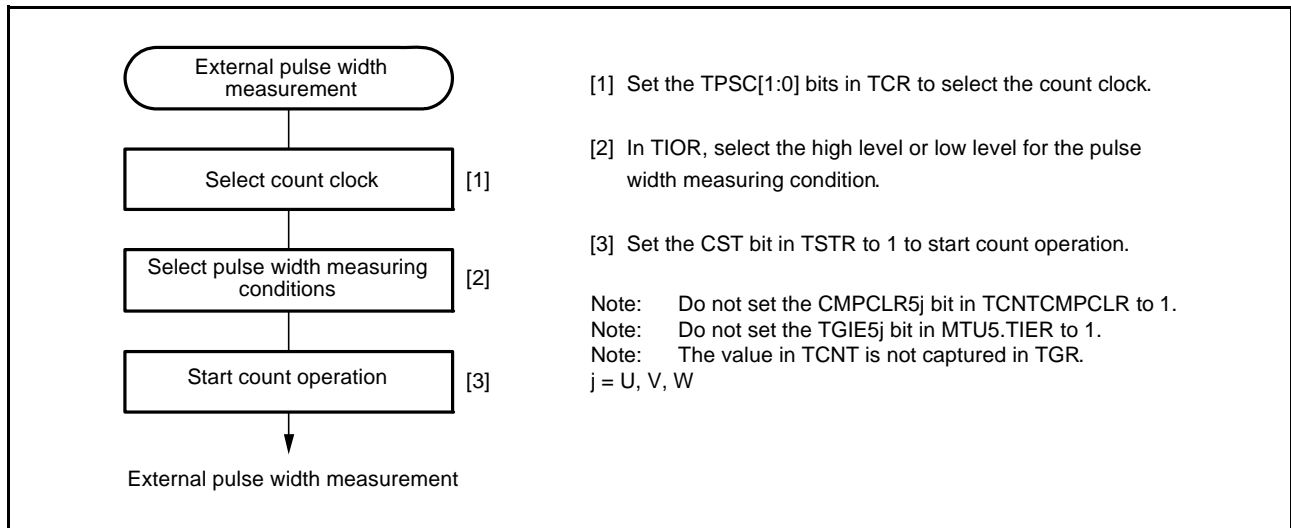


Figure 20.77 Example of External Pulse Width Measurement Setting Procedure

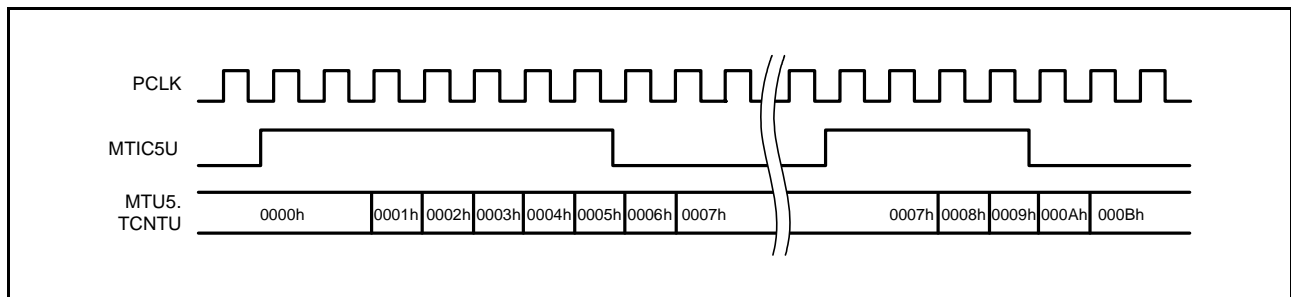


Figure 20.78 Example of External Pulse Width Measurement (Measuring High Pulse Width)

### 20.3.11 Dead Time Compensation

The motor control circuit is configured so that a delay in the dead time (delay between complementary PWM output and inverter output) is fed back to MTU5 (Figure 20.79). The MTU5 external pulse measurement function allows the delay in the dead time to be measured and reflected in the duty ratio, which can be used as dead time compensation for the PWM output waveform in complementary PWM operation when MTU and MTU7 are used (Figure 20.80). Figure 20.81 shows the procedure for setting dead time compensation using MTU5. For details on MTU5 operation at this time, refer to section (2), TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode Operation.

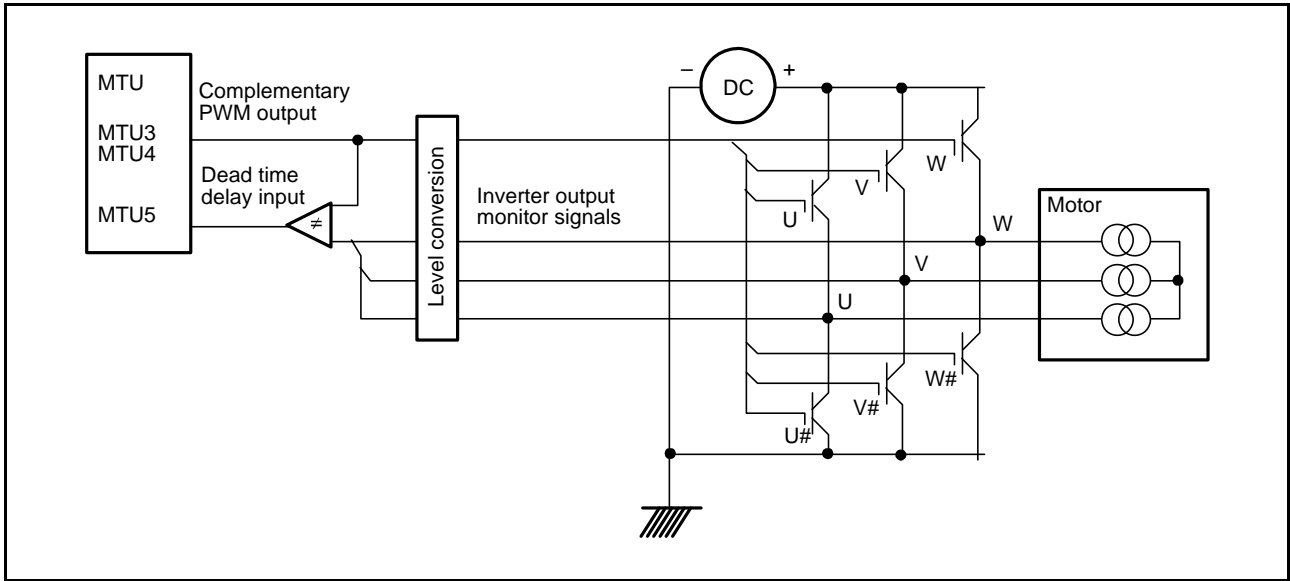


Figure 20.79 Example of Motor Control Circuit Configuration

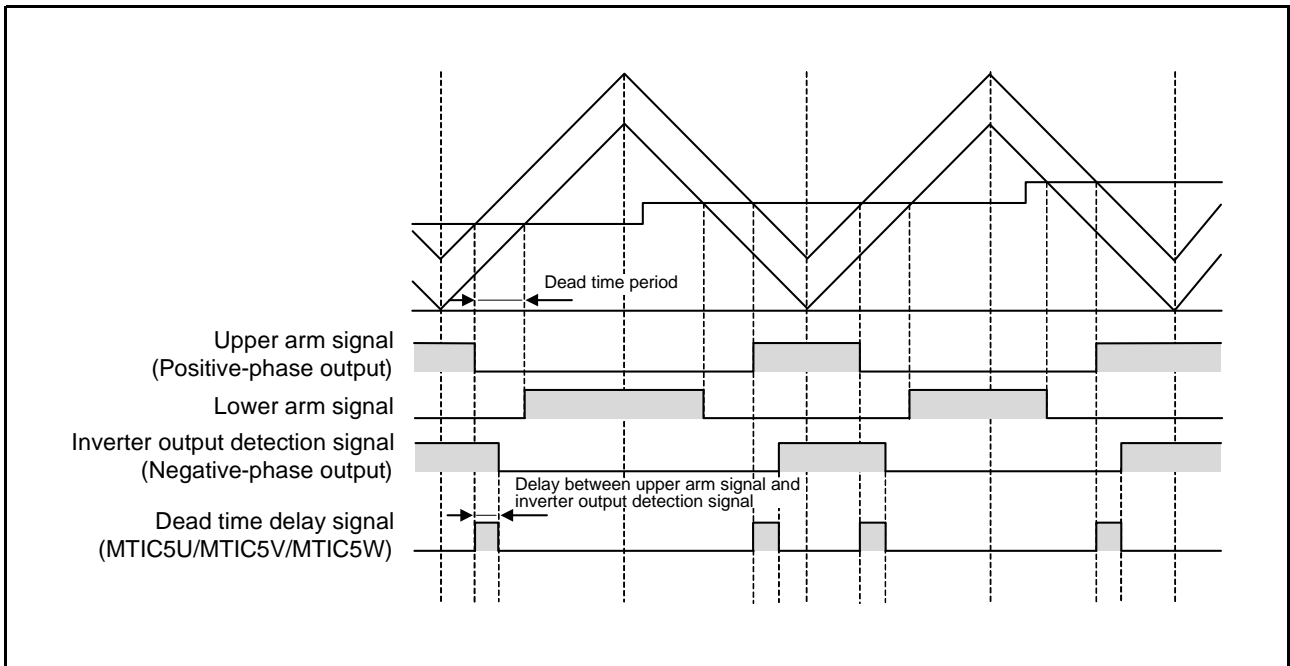


Figure 20.80 Delay in Dead Time in Complementary PWM Mode Operation

## (1) Example of Dead Time Compensation Setting Procedure

Figure 20.81 shows an example of dead time compensation setting procedure by using three counters in MTU5.

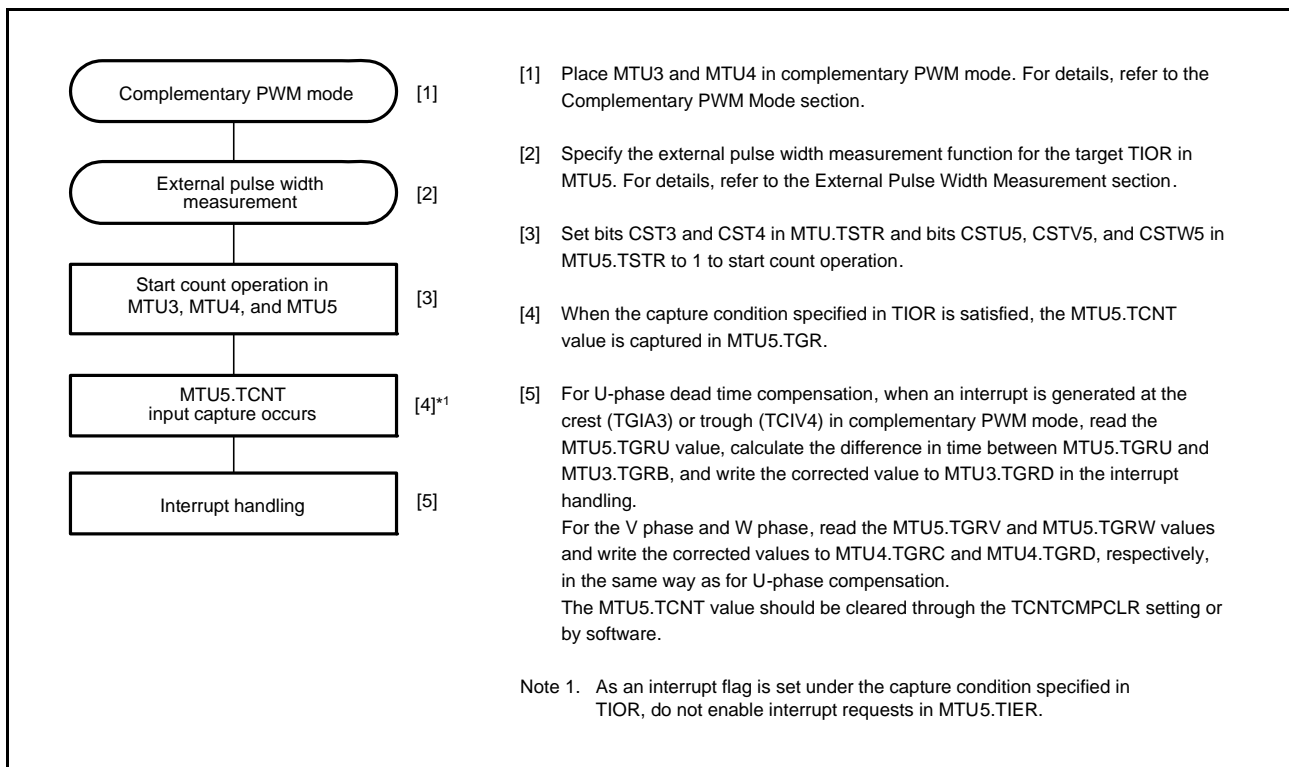


Figure 20.81 Example of Dead Time Compensation Setting Procedure



(2) TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode Operation

The MTU5 external pulse width measurement function can be used to transfer the value in TCNTU, TCNTV, and TCNTW to TGRU, TGRV, and TGRW at the crest, trough, or crest and trough during complementary PWM mode operation. The transfer timing should be set in TIORU, TIORV, and TIORW. When the TCNTCNPCLR.CMPCLR5U, CMPCLR5V, and CMPCLR5W bits are set to 1, TCNTU, TCNTV, and TCNTW are cleared to 0 at the transfer timing for TGRU, TGRV, and TGRW.

Figure 20.82 shows an operation example in which TCNTU is used as a free-running counter without being cleared, and the value is captured in TGRU at the crest or trough in complementary PWM mode.

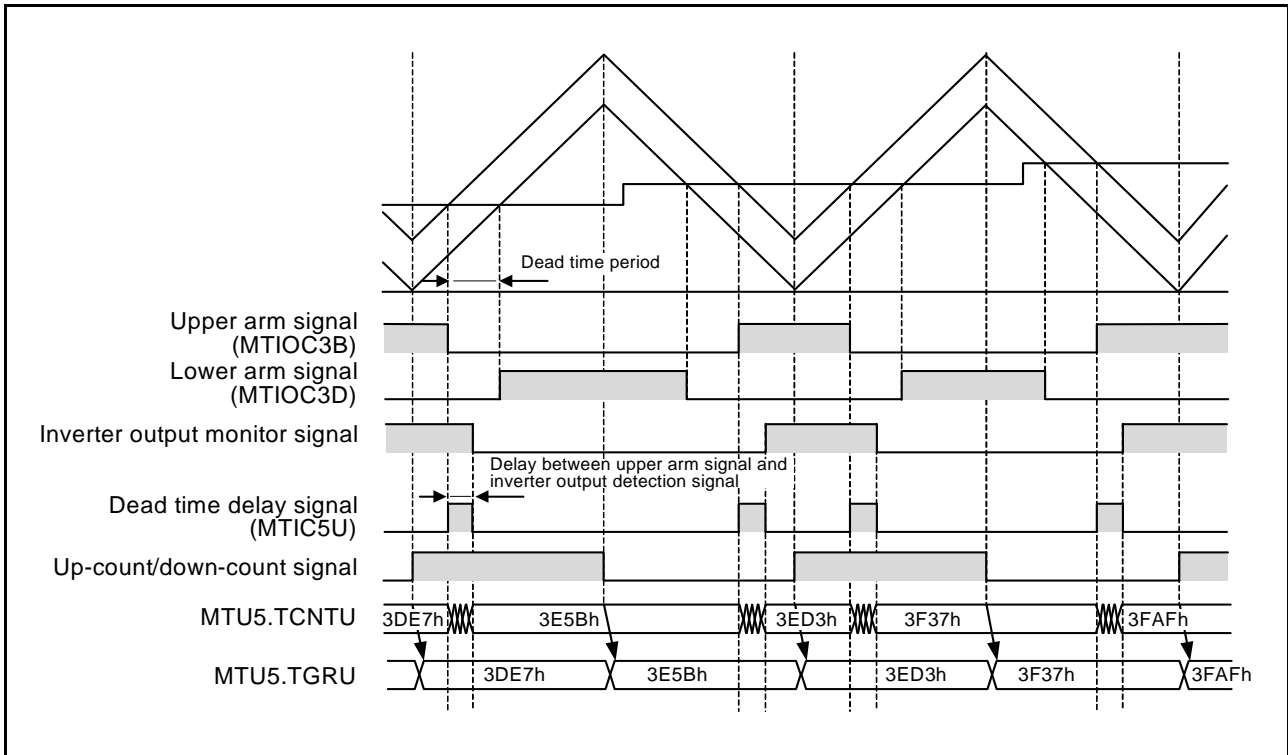


Figure 20.82 MTU5.TCNT Capture at Crest and/or Trough in Complementary PWM Mode Operation

### 20.3.12 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 20.83 shows the timing of noise filtering.

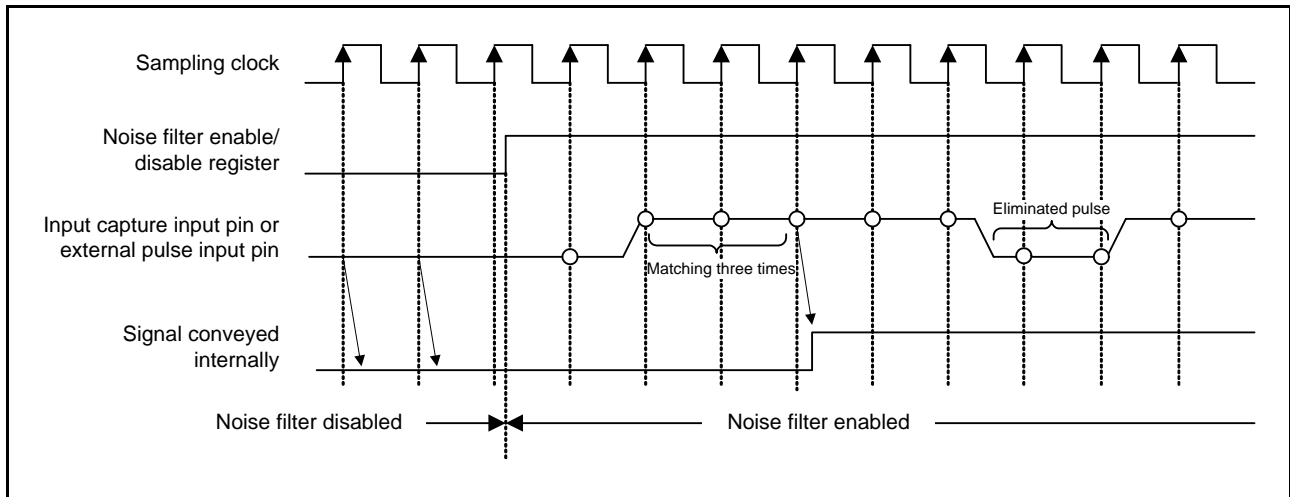


Figure 20.83 Timing of Noise Filtering

## 20.4 Interrupt Sources

### 20.4.1 Interrupt Sources and Priorities

There are three interrupt sources; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is detected, an interrupt is requested if the corresponding enable/disable bit in TIER is set to 1. Relative channel priorities can be changed by the interrupt controller; however the priority within a channel is fixed. For details, refer to section 14, Interrupt Controller (ICUb).

Table 20.57 lists the MTU interrupt sources.

**Table 20.57 MTU Interrupt Sources (1)**

Channel	Name	Interrupt Source	DTC Activation	Priority
MTU0	TGIA0	MTU0.TGRA input capture/compare match	Possible	High ↑
	TGIB0	MTU0.TGRB input capture/compare match	Possible	
	TGIC0	MTU0.TGRC input capture/compare match	Possible	
	TGID0	MTU0.TGRD input capture/compare match	Possible	
	TCIV0	MTU0.TCNT overflow	Not possible	
	TGIE0	MTU0.TGRE compare match	Not possible	
	TGIF0	MTU0.TGRF compare match	Not possible	
MTU1	TGIA1	MTU1.TGRA input capture/compare match	Possible	↑
	TGIB1	MTU1.TGRB input capture/compare match	Possible	
	TCIV1	MTU1.TCNT overflow	Not possible	
	TCIU1	MTU1.TCNT underflow	Not possible	
MTU2	TGIA2	MTU2.TGRA input capture/compare match	Possible	↑
	TGIB2	MTU2.TGRB input capture/compare match	Possible	
	TCIV2	MTU2.TCNT overflow	Not possible	
	TCIU2	MTU2.TCNT underflow	Not possible	
MTU3	TGIA3	MTU3.TGRA input capture/compare match	Possible	↑
	TGIB3	MTU3.TGRB input capture/compare match	Possible	
	TGIC3	MTU3.TGRC input capture/compare match	Possible	
	TGID3	MTU3.TGRD input capture/compare match	Possible	
	TCIV3	MTU3.TCNT overflow	Not possible	
MTU4	TGIA4	MTU4.TGRA input capture/compare match	Possible	↑
	TGIB4	MTU4.TGRB input capture/compare match	Possible	
	TGIC4	MTU4.TGRC input capture/compare match	Possible	
	TGID4	MTU4.TGRD input capture/compare match	Possible	
	TCIV4	MTU4.TCNT overflow/underflow	Possible	
MTU5	TGIU5	MTU5.TGRU input capture/compare match	Possible	Low
	TGIV5	MTU5.TGRV input capture/compare match	Possible	
	TGIW5	MTU5.TGRW input capture/compare match	Possible	

Note: This table lists the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

### (1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TIER.TGIE bit 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 TIER.TGIE bit 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 TIER.TCIEU bit is set to 1 when a TCNT underflow occurs on a channel. The MTU has two underflow interrupts (one each for MTU1 and MTU2).

## 20.4.2 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, refer to section 16, 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.

## 20.4.3 A/D Converter Activation

The A/D converter can be activated by one of the following five methods in the MTU. Table 20.58 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 MTU4.TIER.TTGE2 bit 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 TIER.TTGE bit is set to 1
- When the MTU4.TCNT count reaches the trough (MTU4.TCNT = 0000h) during complementary PWM mode operation while the MTU4.TIER.TTGE2 bit 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

An input capture or compare match between MTU0.TCNT, MTU0.TGRA, and MTU0.TGRB activates the A/D converter. 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 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 TADCR.UT4AE bit, TADCR.DT4AE bit, TADCR.UT4BE bit, or TADCR.DT4BE bit is set to 1. For details, refer to section 20.3.9, 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 20.58 Interrupt Sources and A/D Converter Start Request Signals**

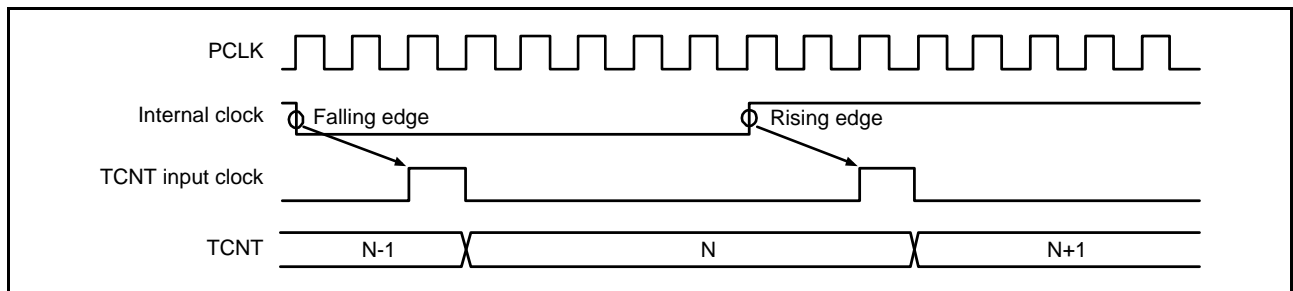
Target Registers	A/D Start Request Source	A/D Converter Start Request Signal
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRGAN
MTU1.TGRA and MTU1.TCNT		
MTU2.TGRA and MTU2.TCNT		
MTU3.TGRA and MTU3.TCNT		
MTU4.TGRA and MTU4.TCNT		
MTU4.TCNT	MTU4.TCNT trough in complementary PWM mode	
MTU0.TGRA 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		TRG4AN
TADCORB and MTU4.TCNT		TRG4BN
TADCORA and MTU4.TCNT or TADCORB and MTU4.TCNT		TRG4ABN

## 20.5 Operation Timing

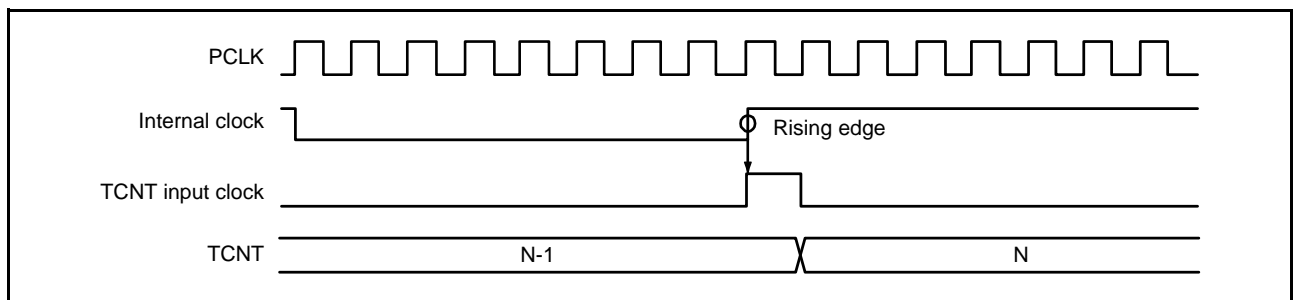
### 20.5.1 Input/Output Timing

#### (1) TCNT Count Timing

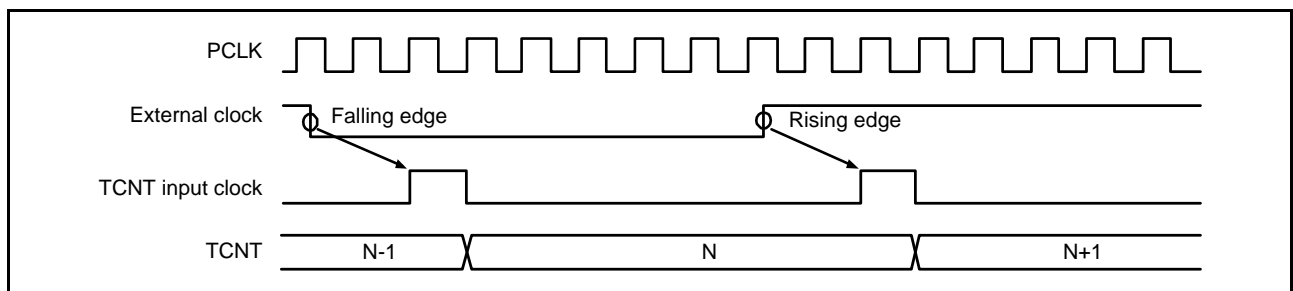
Figure 20.84 and Figure 20.85 show the TCNT count timing for TGI interrupt in internal clock operation, Figure 20.86 shows the TCNT count timing in external clock operation (normal mode), and Figure 20.87 shows the TCNT count timing in external clock operation (phase counting mode).



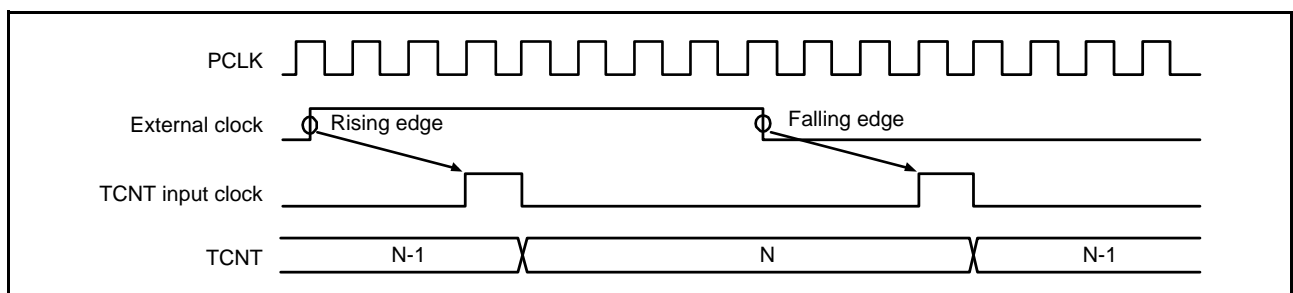
**Figure 20.84** Count Timing in Internal Clock Operation (MTU0 to MTU4)



**Figure 20.85** Count Timing in Internal Clock Operation (MTU5)



**Figure 20.86** Count Timing in External Clock Operation (MTU0 to MTU4)



**Figure 20.87** 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 20.88 shows the output compare output timing (normal mode or PWM mode) and Figure 20.89 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

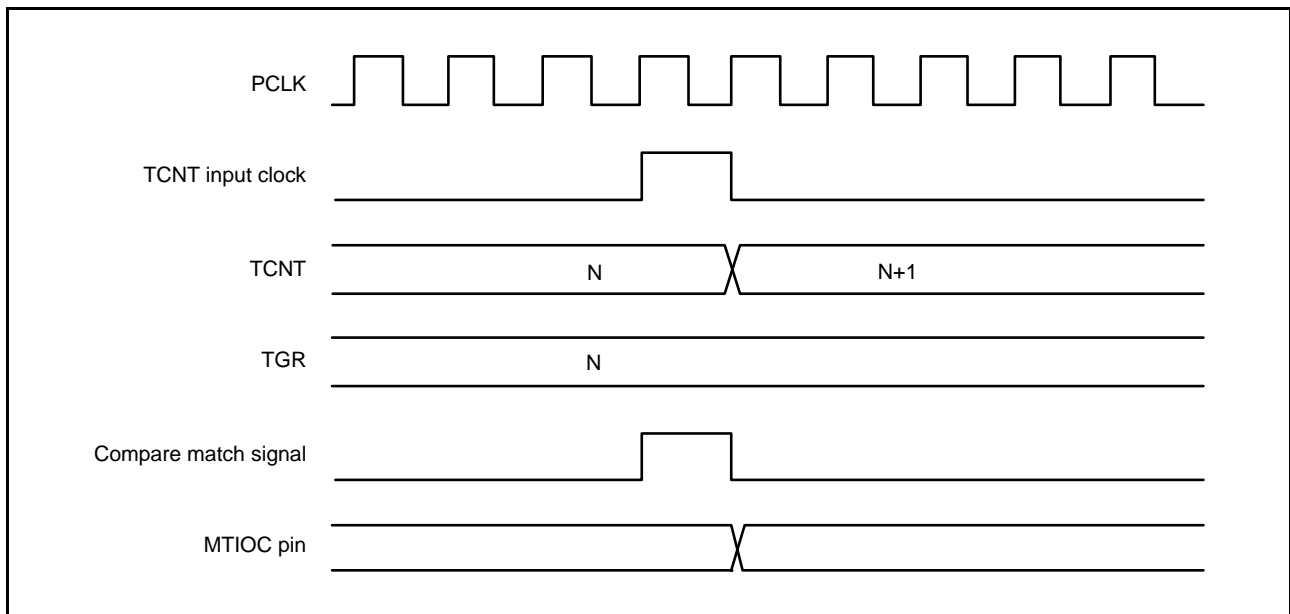


Figure 20.88 Output Compare Output Timing (Normal Mode or PWM Mode)

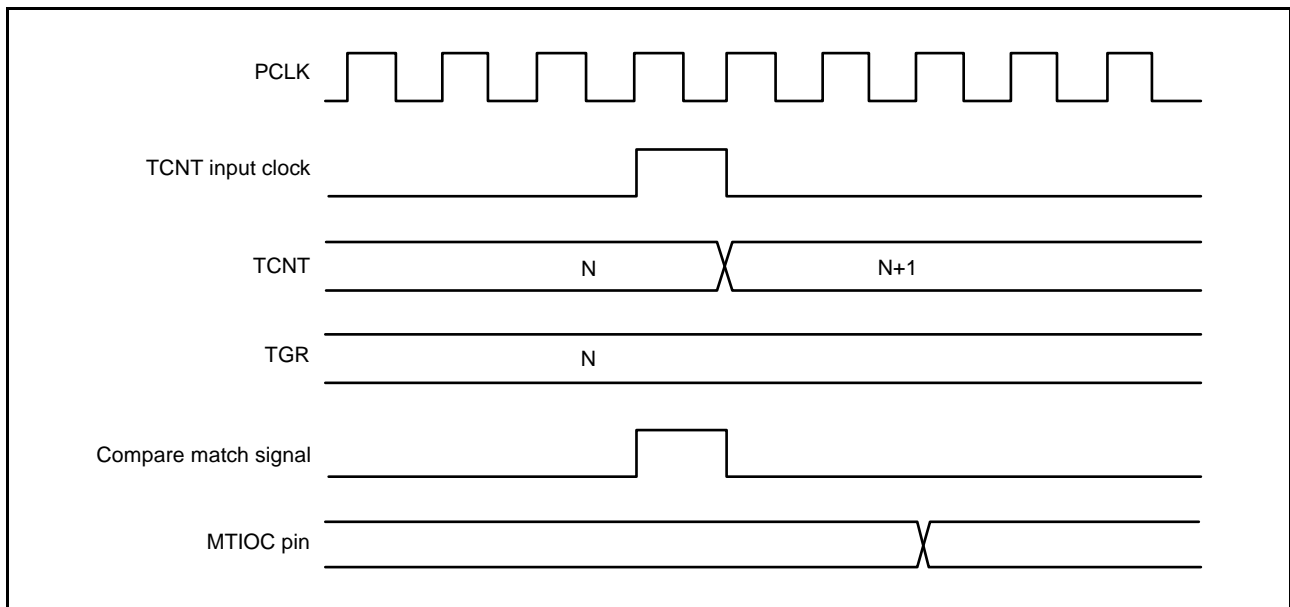


Figure 20.89 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode)

(3) Input Capture Signal Timing

Figure 20.90 shows the input capture signal timing.

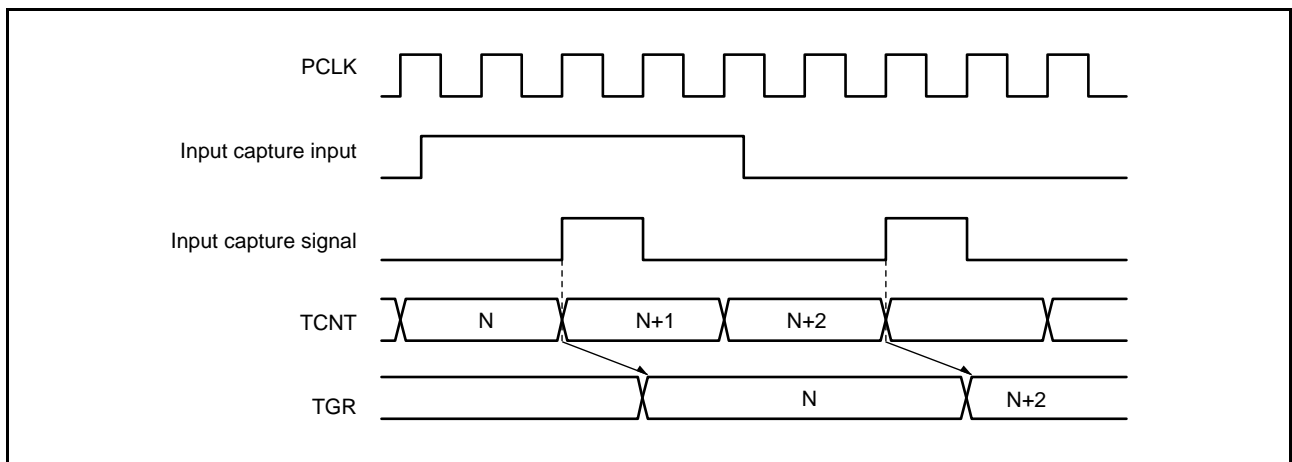


Figure 20.90 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 20.91 and Figure 20.92 show the timing when counter clearing on compare match is specified, and Figure 20.93 shows the timing when counter clearing on input capture is specified.

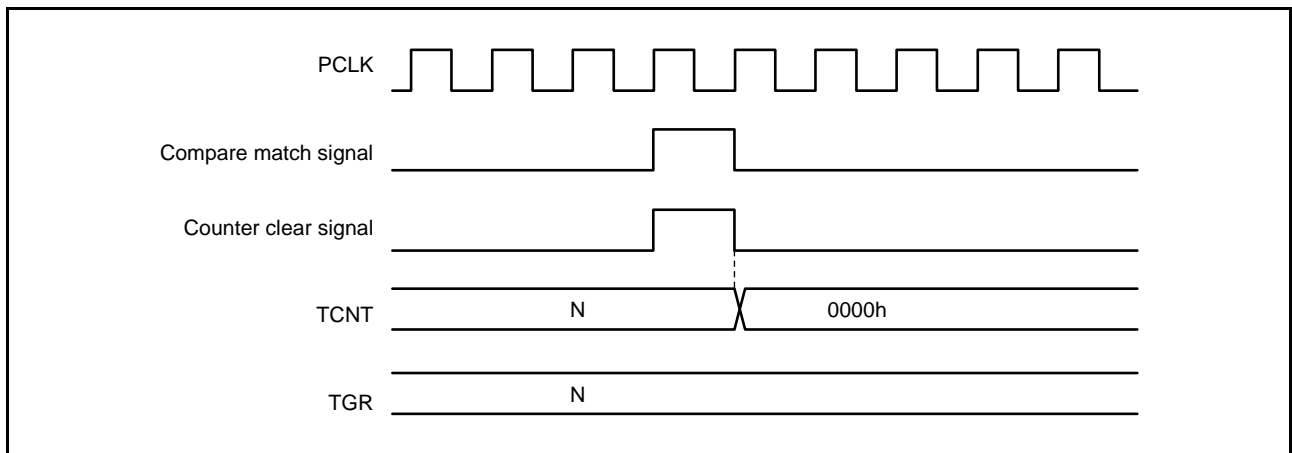


Figure 20.91 Counter Clear Timing (Compare Match) (MTU0 to MTU4)

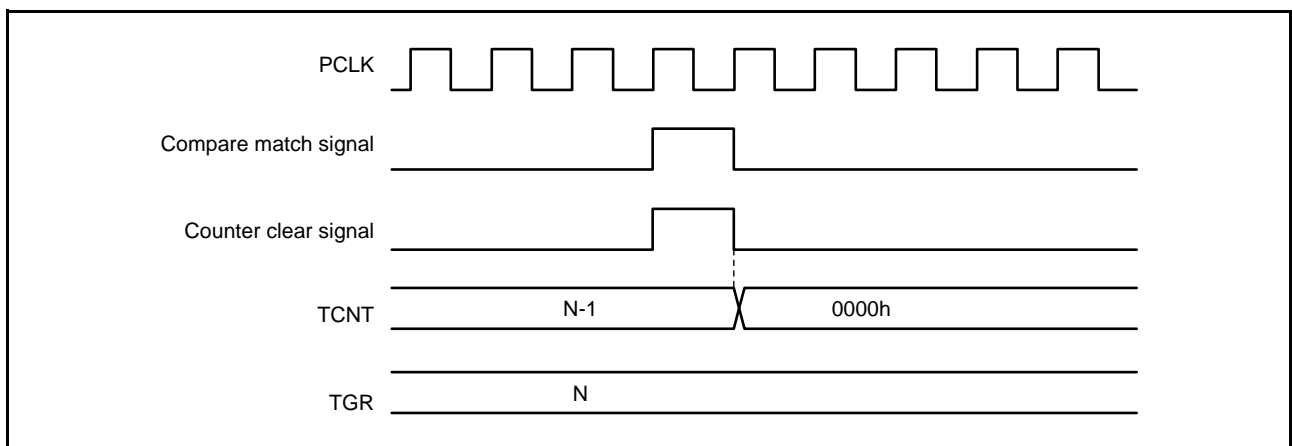


Figure 20.92 Counter Clear Timing (Compare Match) (MTU5)



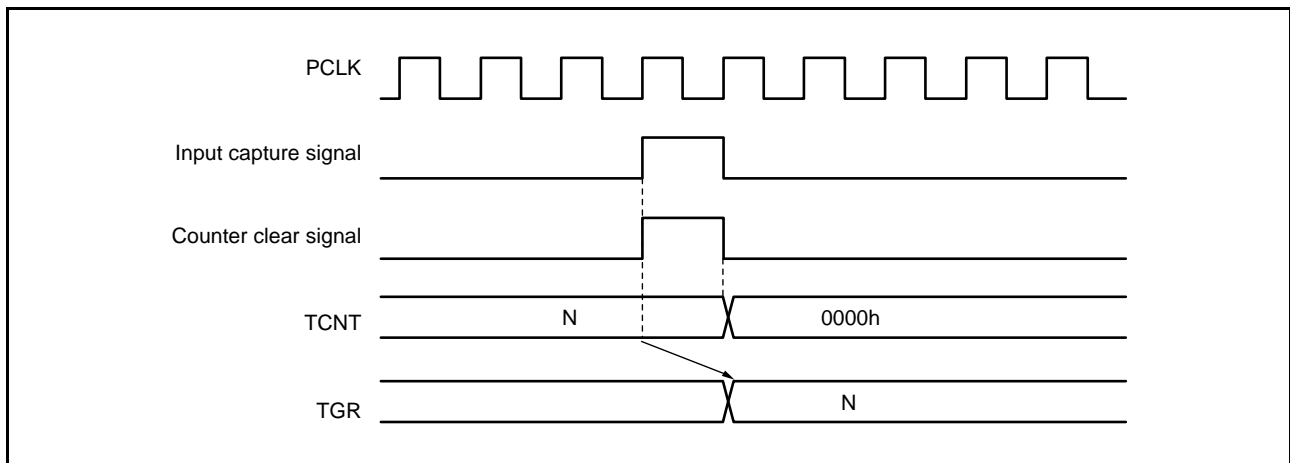


Figure 20.93 Counter Clear Timing (Input Capture) (MTU0 to MTU5)

(5) Buffer Operation Timing

Figure 20.94 to Figure 20.96 show the timing in buffer operation.

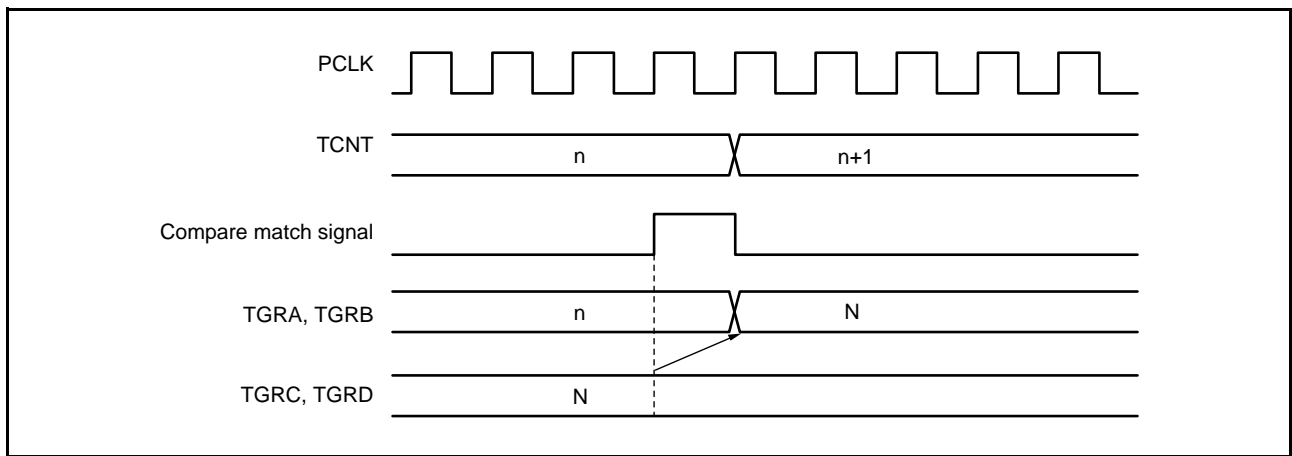


Figure 20.94 Buffer Operation Timing (Compare Match)

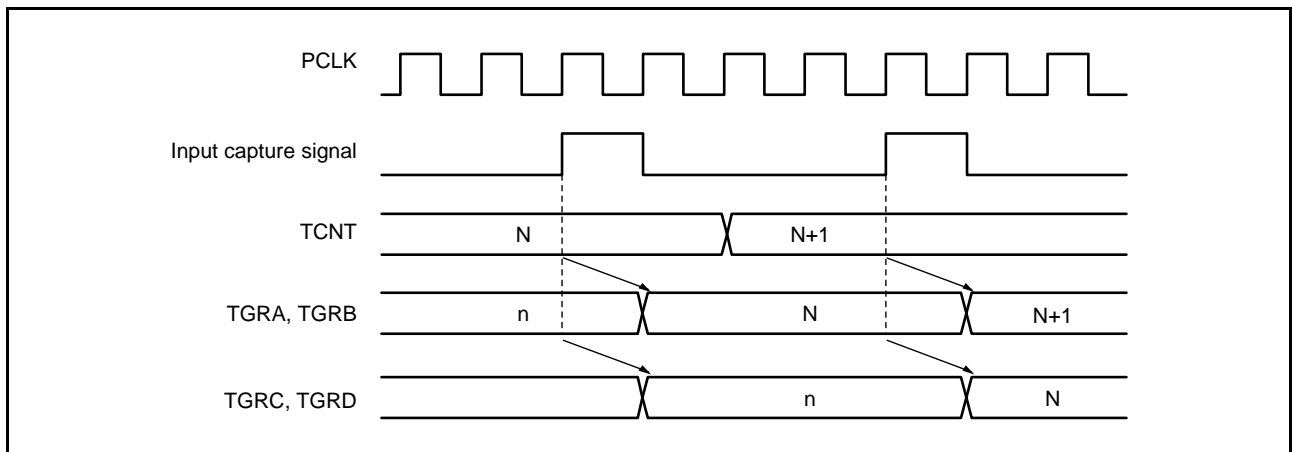


Figure 20.95 Buffer Operation Timing (Input Capture)

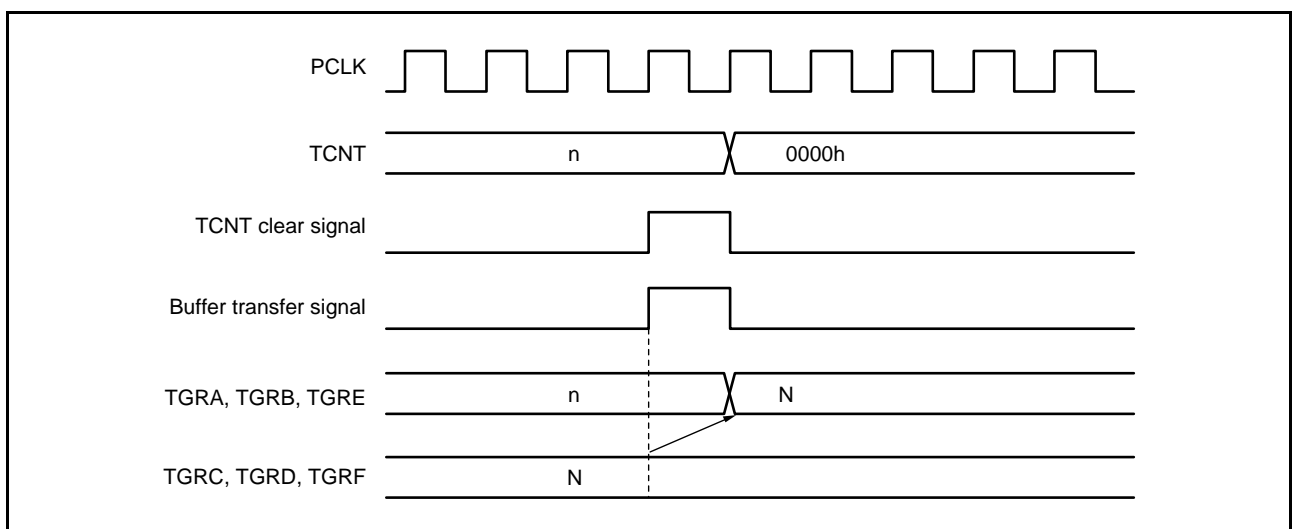


Figure 20.96 Buffer Operation Timing (When TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 20.97 to Figure 20.99 show the buffer transfer timing in complementary PWM mode.

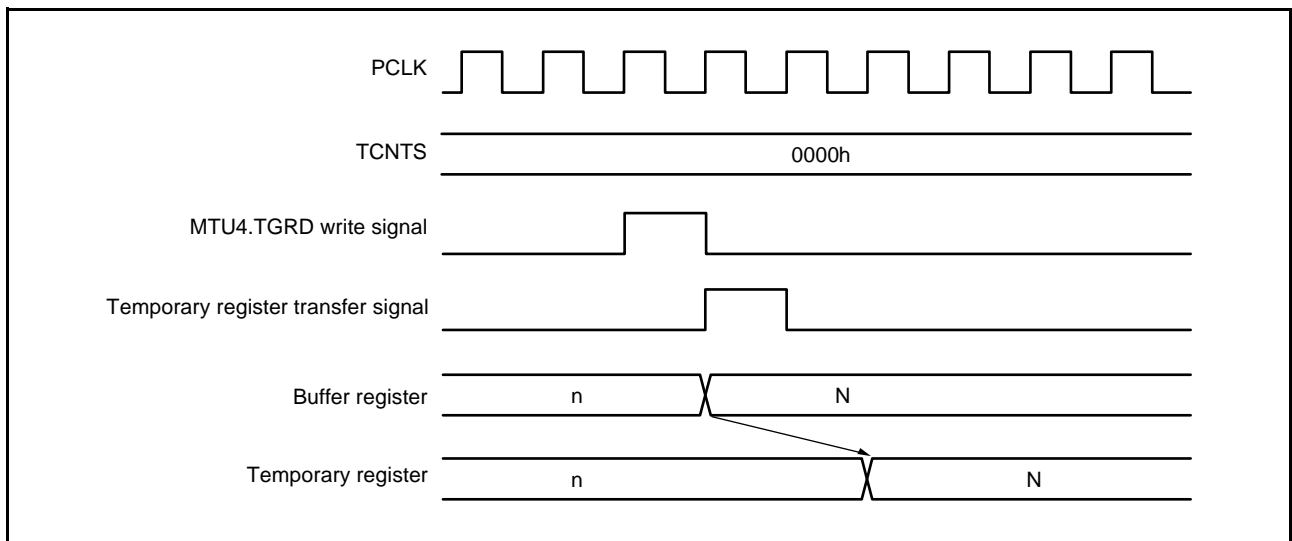


Figure 20.97 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

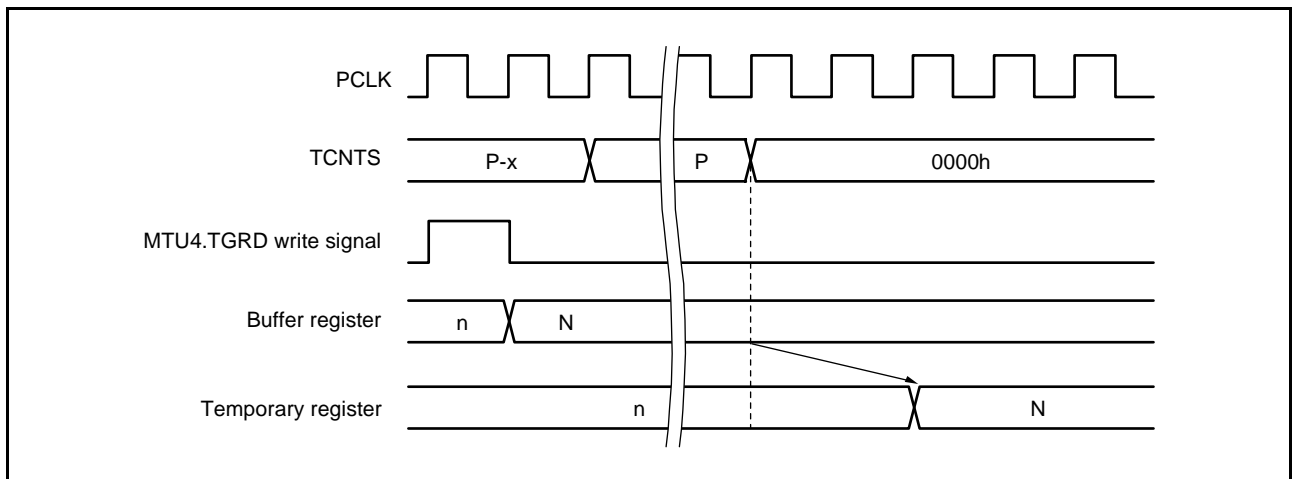


Figure 20.98 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

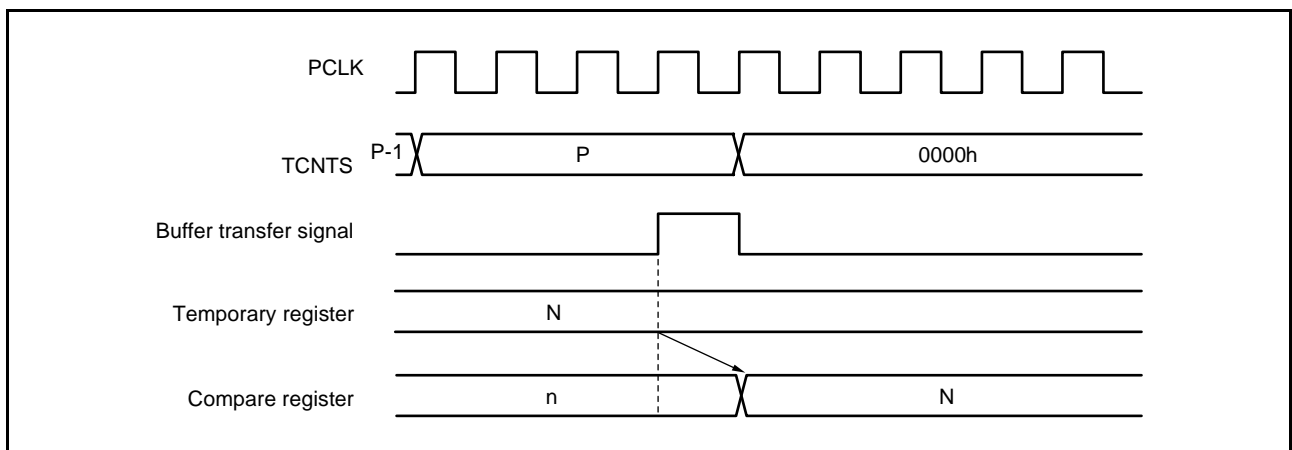
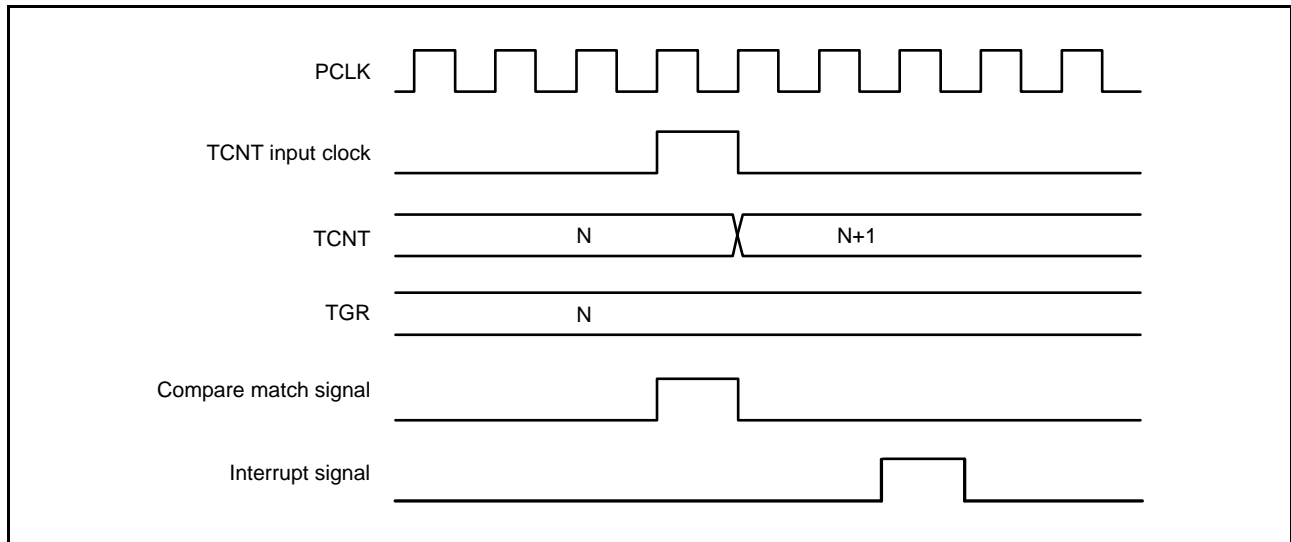


Figure 20.99 Transfer Timing from Temporary Register to Compare Register

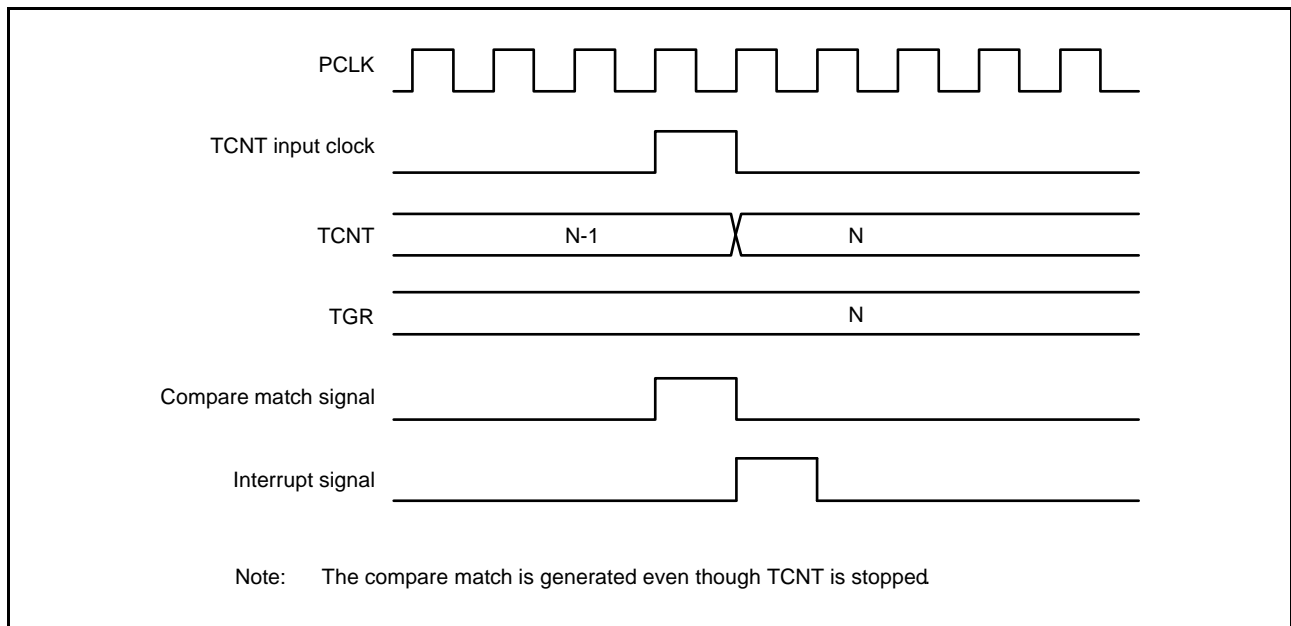
### 20.5.2 Interrupt Signal Timing

#### (1) Timing for TGI Interrupt by Compare Match

Figure 20.100 and Figure 20.101 show the TGI interrupt request signal timing on compare match.



**Figure 20.100 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4)**



**Figure 20.101 TGI Interrupt Timing (Compare Match) (MTU5)**

(2) Timing for TGI Interrupt by Input Capture

Figure 20.102 and Figure 20.103 show TGI interrupt request signal timing on input capture.

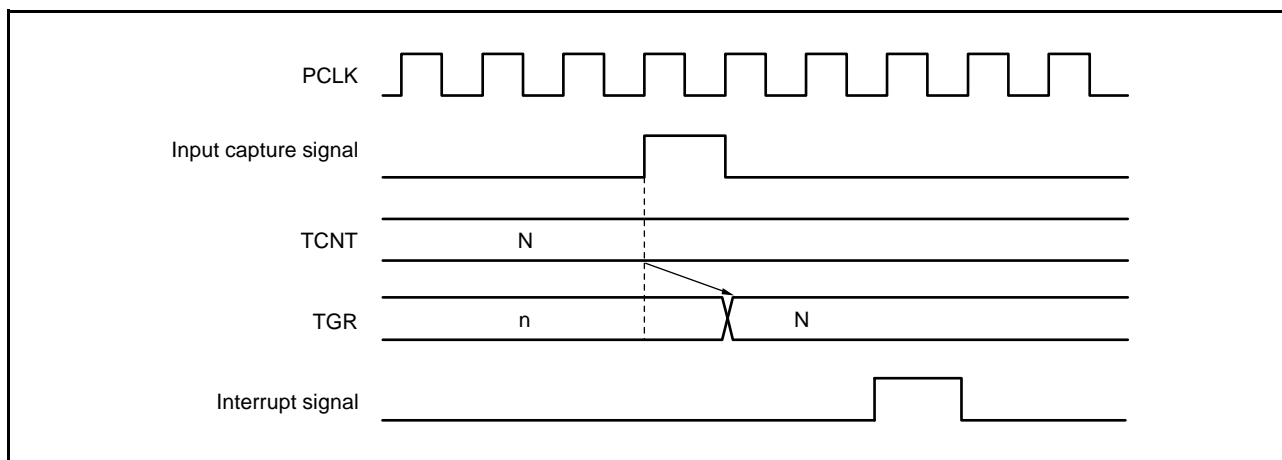


Figure 20.102 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4)

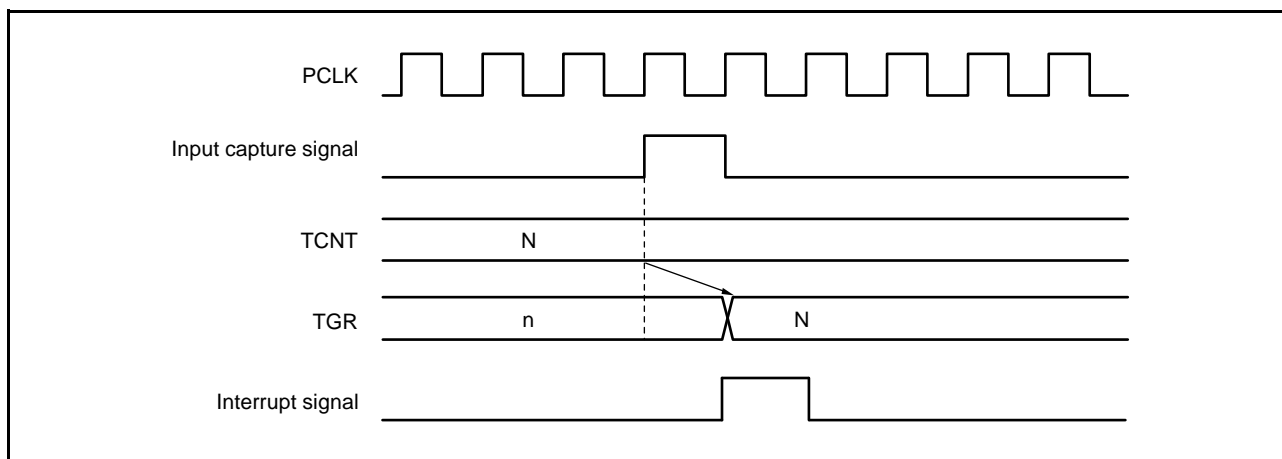


Figure 20.103 TGI Interrupt Timing (Input Capture) (MTU5)

(3) TCIV and TCIU Interrupt Timing

Figure 20.104 shows the TCIV interrupt request signal timing on overflow.

Figure 20.105 shows the TCIU interrupt request signal timing on underflow.

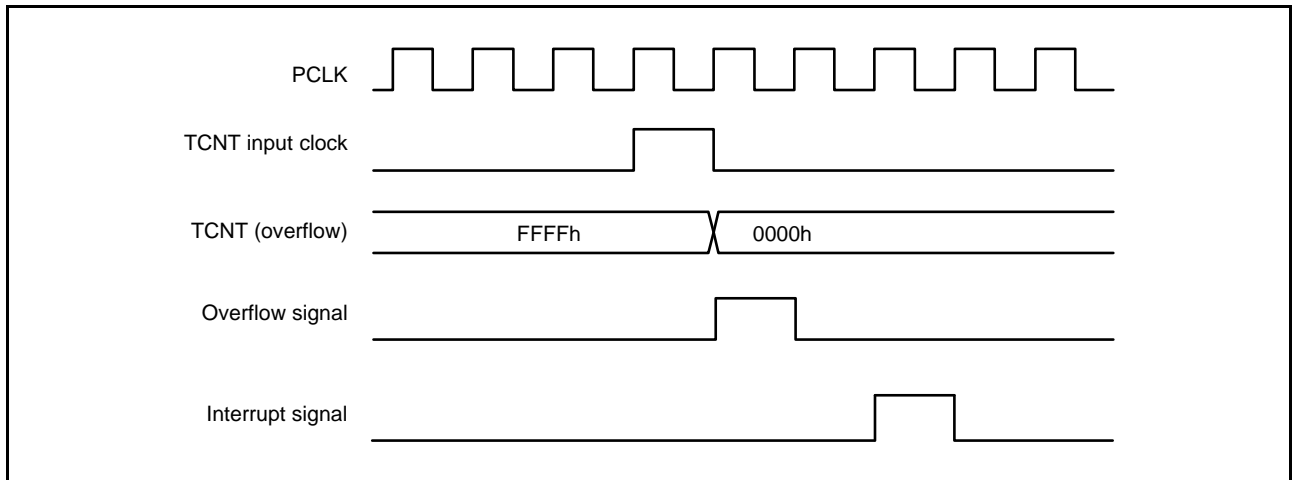


Figure 20.104 TCIV Interrupt Timing

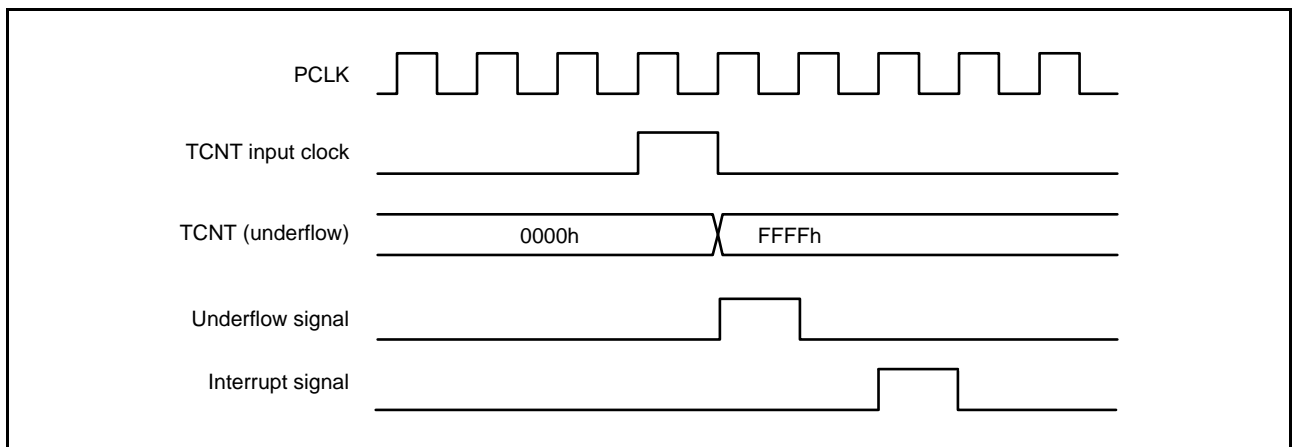


Figure 20.105 TCIU Interrupt Timing

## 20.6 Usage Notes

### 20.6.1 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 releasing the module clock stop mode. For details, refer to section 11, Low Power Consumption.

### 20.6.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 PCLK clock for single-edge detection, and at least 2.5 PCLK clock 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 clock, and the pulse width must be at least 2.5 PCLK clock. Figure 20.106 shows the input clock conditions in phase counting mode.

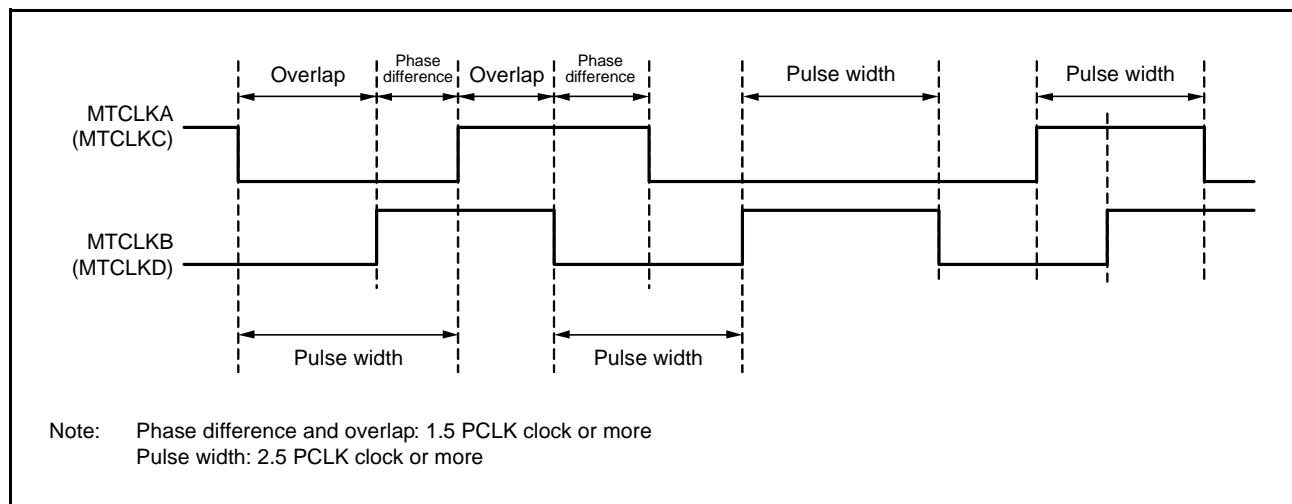


Figure 20.106 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

### 20.6.3 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 count clock frequency set by TCR.TPSC[2:0] bits

N: TGR setting

### 20.6.4 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 20.107 shows the timing in this case.

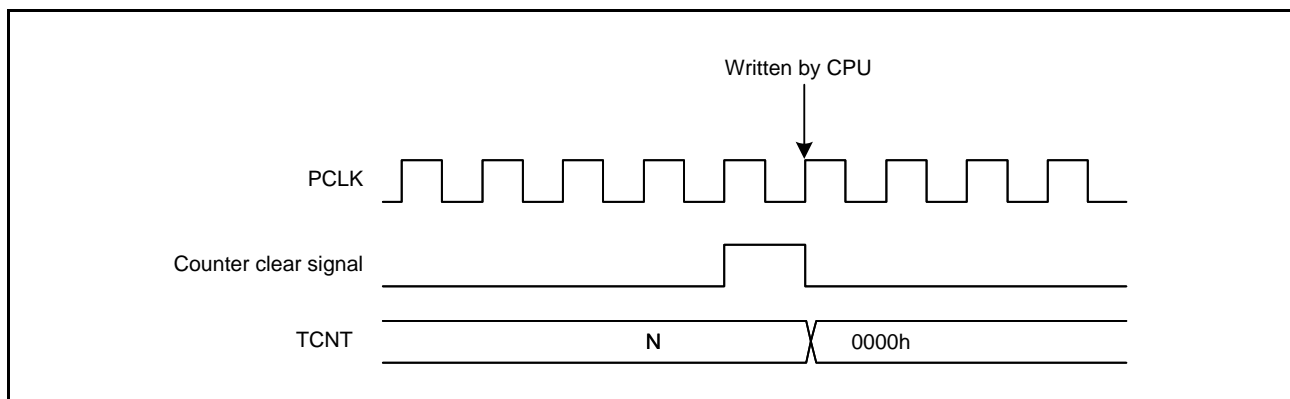


Figure 20.107 Contention between TCNT Write and Counter Clear Operations



### 20.6.5 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 20.108 shows the timing in this case.

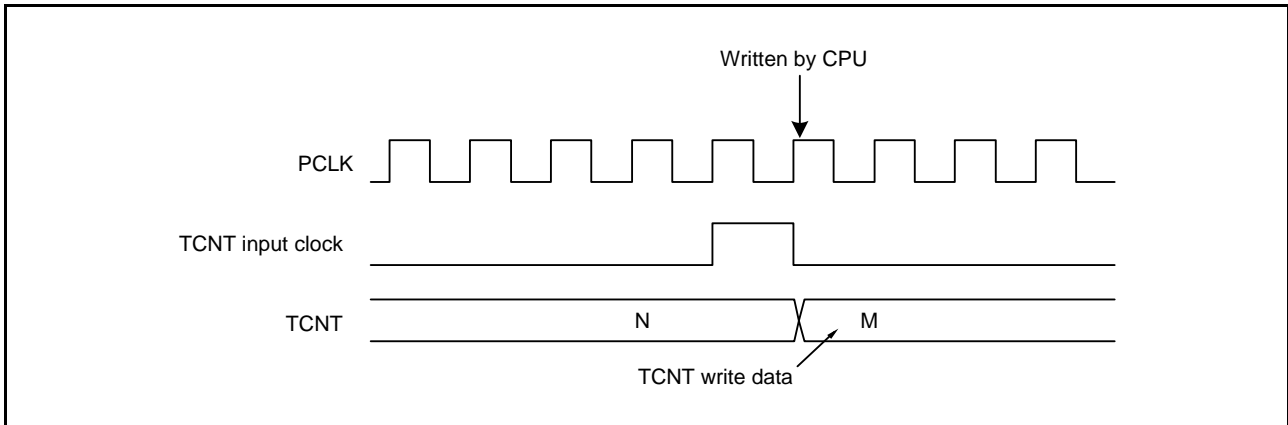


Figure 20.108 Contention between TCNT Write and Increment Operations

### 20.6.6 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 20.109 shows the timing in this case.

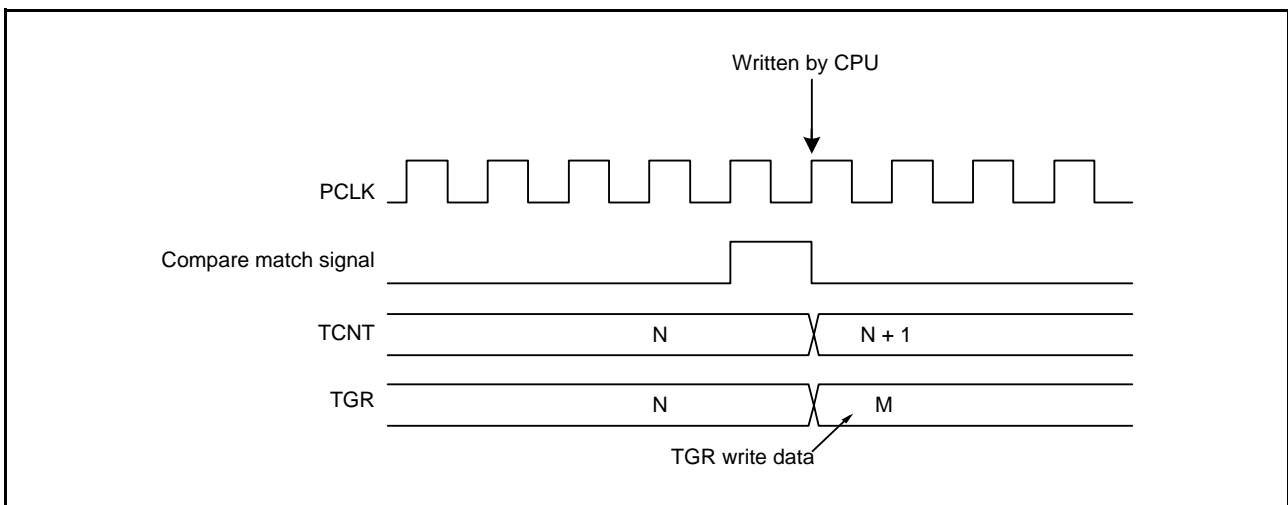


Figure 20.109 Contention between TGR Write Operation and Compare Match

### 20.6.7 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 20.110 shows the timing in this case.

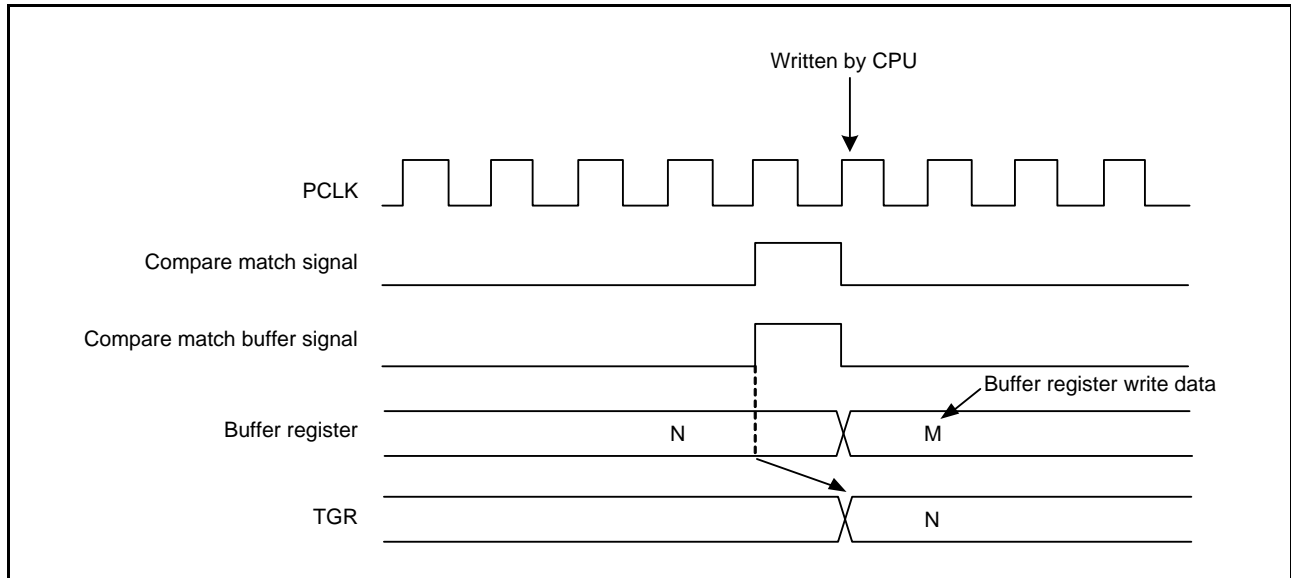


Figure 20.110 Contention between Buffer Register Write Operation and Compare Match

### 20.6.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the timer buffer operation transfer mode register (TBTM), if TCNT clearing occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 20.111 shows the timing in this case.

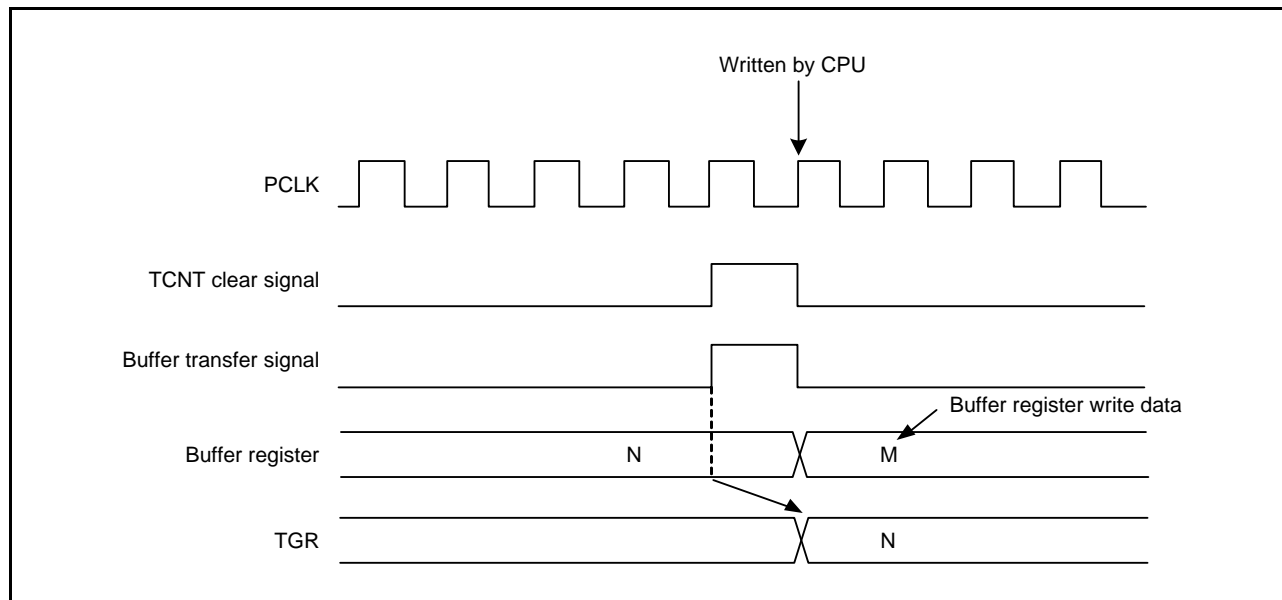


Figure 20.111 Contention between Buffer Register Write and TCNT Clear Operations

### 20.6.9 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 20.112 shows the timing in this case.

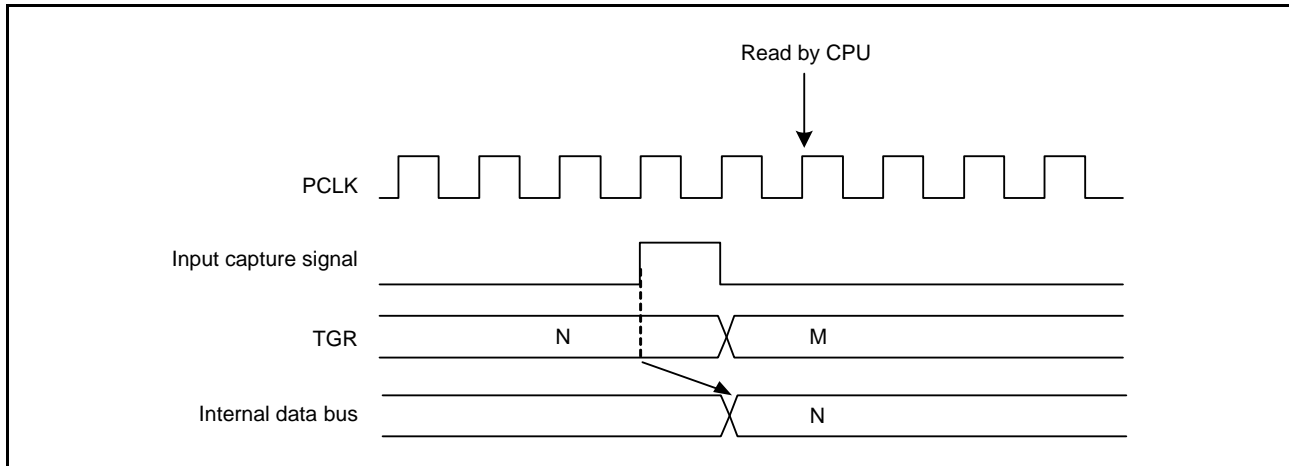


Figure 20.112 Contention between TGR Read Operation and Input Capture (MTU0 to MTU5)

### 20.6.10 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 20.113 and Figure 20.114 show the timing in this case.

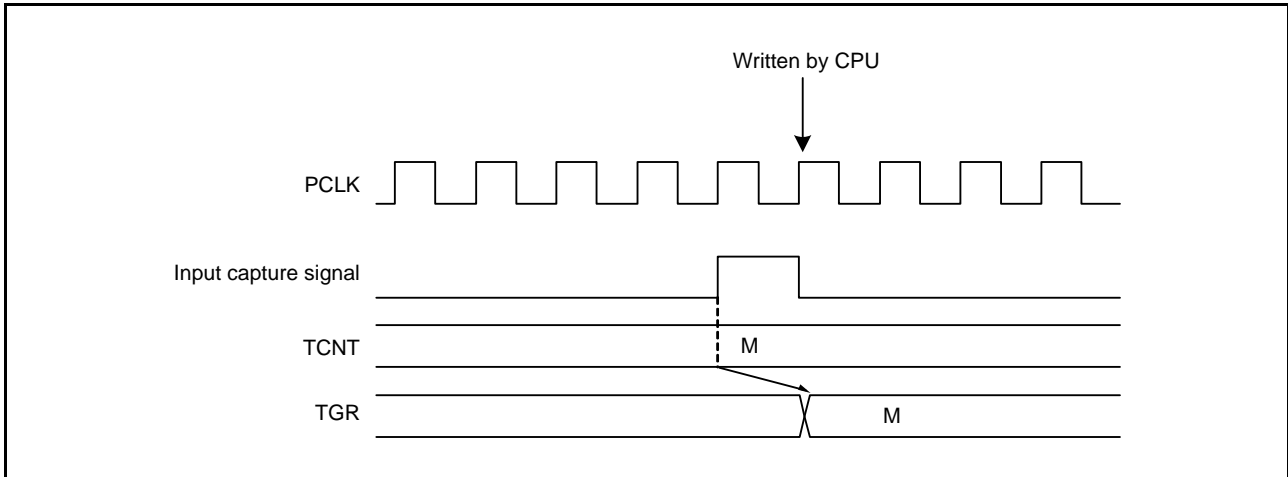


Figure 20.113 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4)

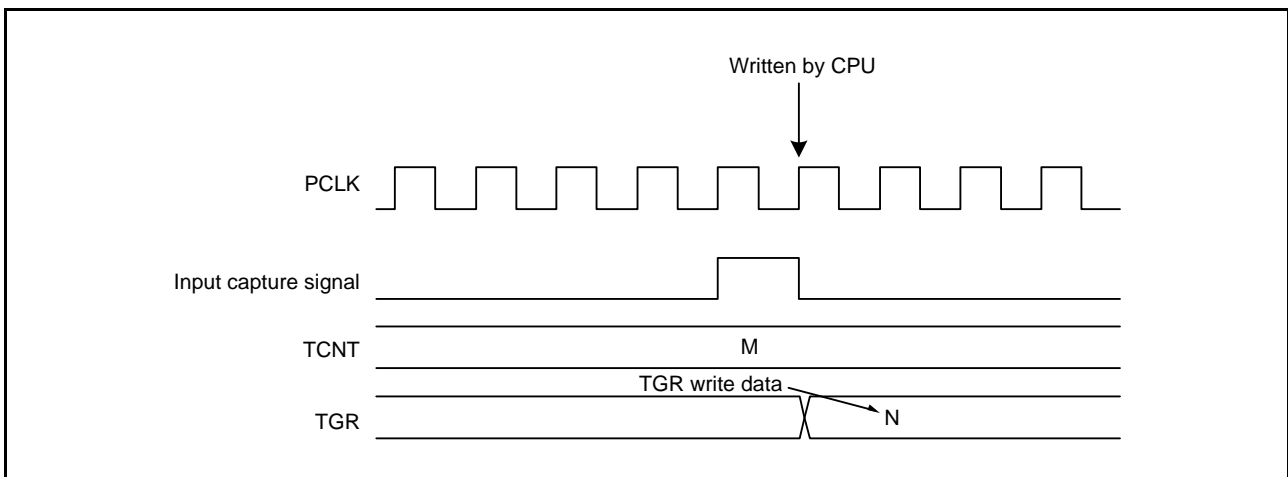


Figure 20.114 Contention between TGR Write Operation and Input Capture (MTU5)

### 20.6.11 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 20.115 shows the timing in this case.

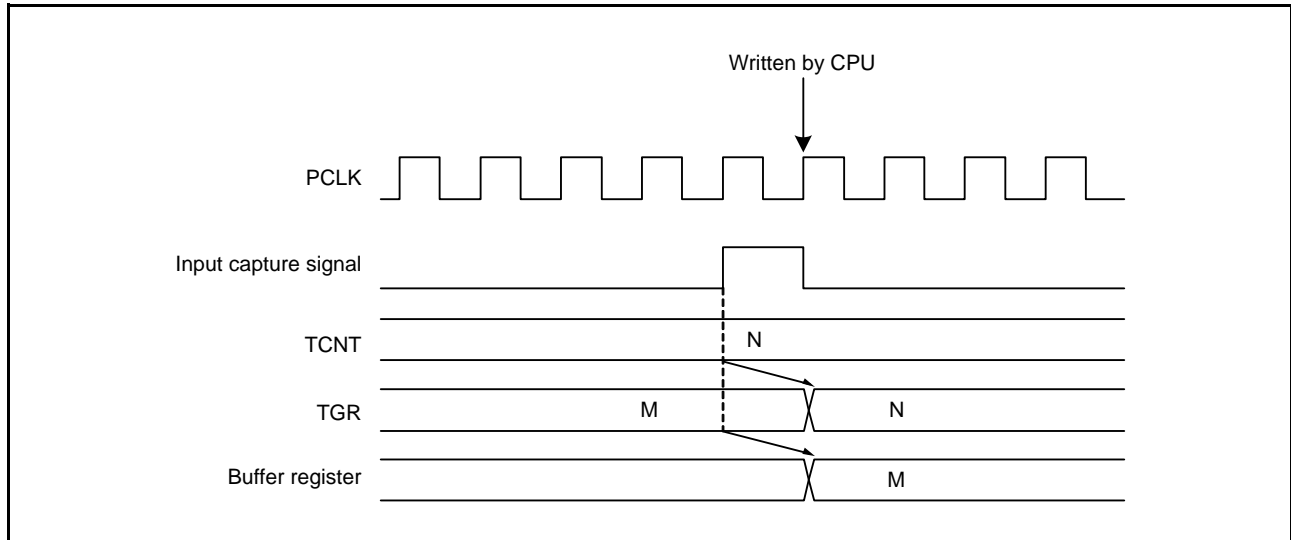


Figure 20.115 Contention between Buffer Register Write Operation and Input Capture

### 20.6.12 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

With timer counters MTU1.TCNT and MTU2.TCNT in a cascade, when a contention occurs between MTU1.TCNT counting (an MTU2.TCNT overflow/underflow) and the MTU2.TCNT write operation 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 20.116 shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

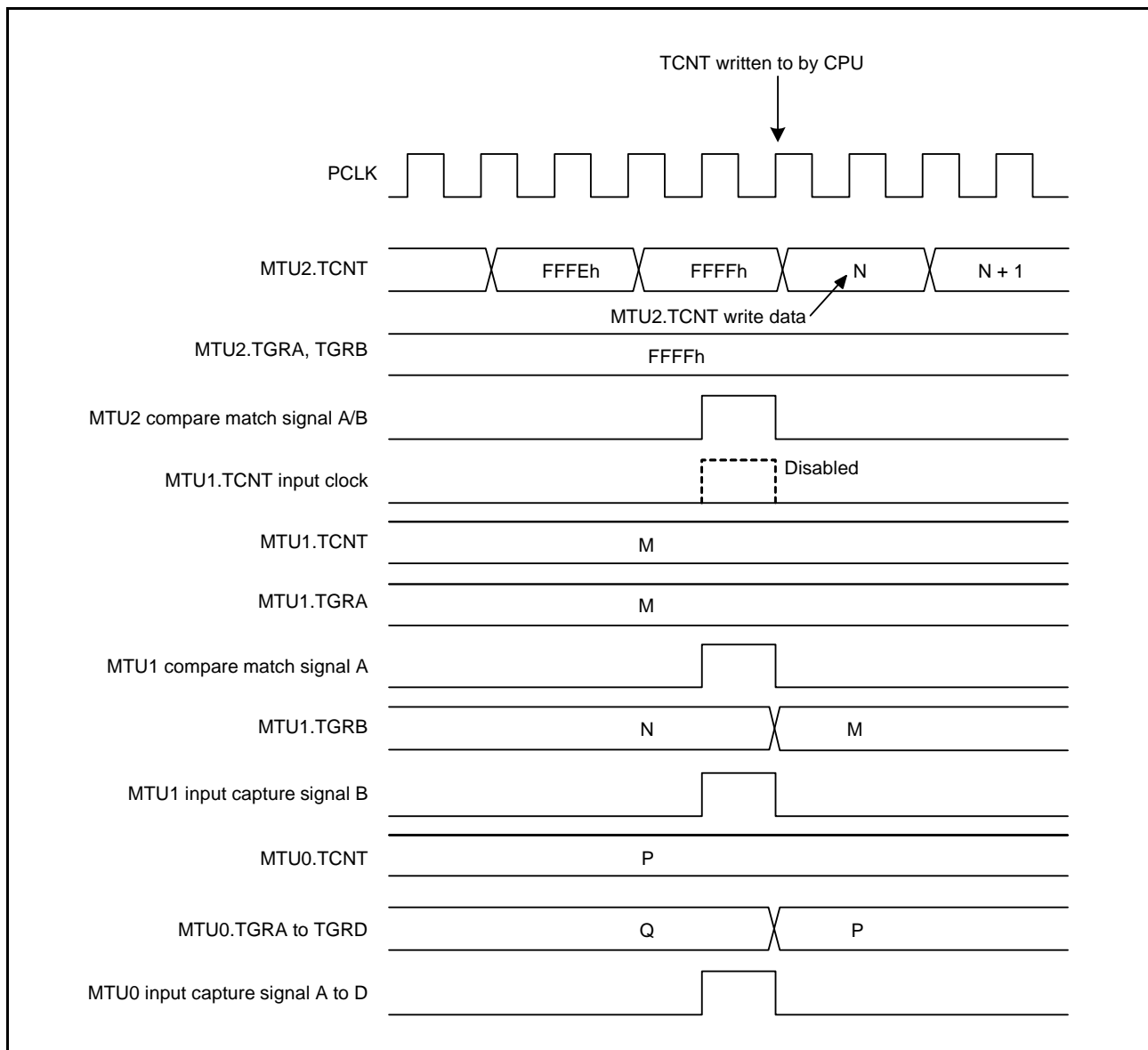


Figure 20.116 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

### 20.6.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT is stopped in complementary PWM mode, 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 20.117 shows this operation.

When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT.

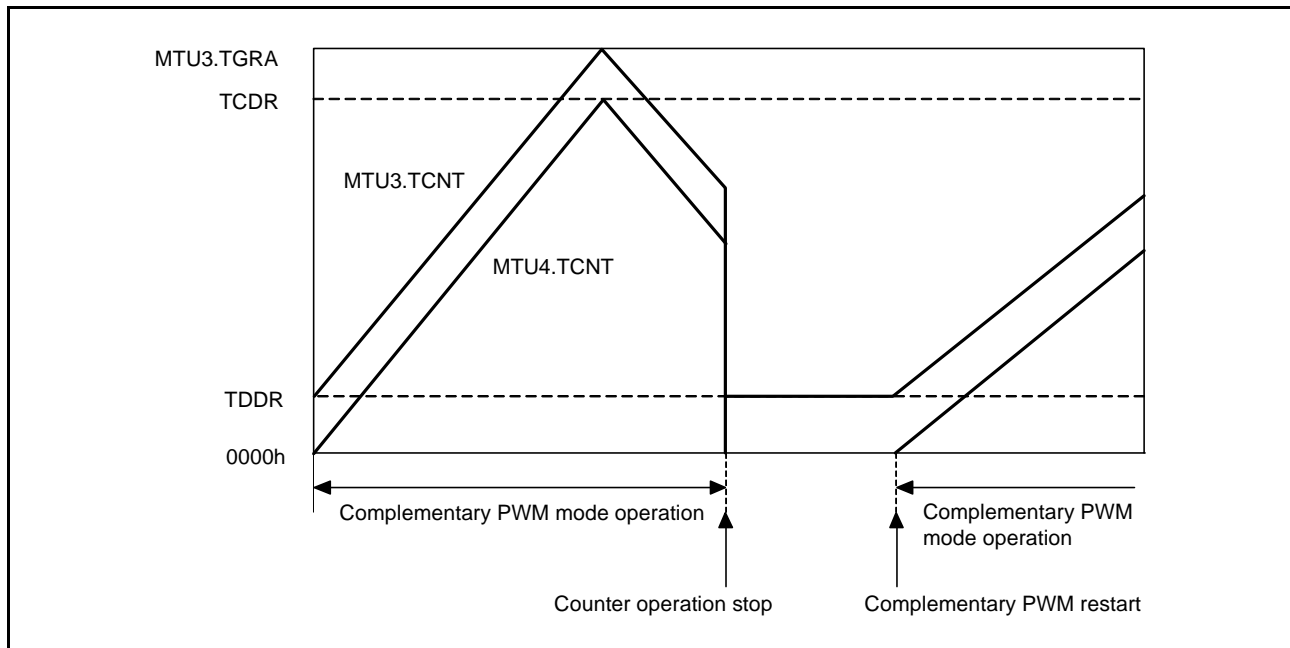


Figure 20.117 Counter Value when Stopped in Complementary PWM Mode (MTU3 and MTU4 Operation)

### 20.6.14 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 MTU4.TMDR.BFA bit and MTU4.TMDR.BFB bit should be set to 0. Setting the MTU4.TMDR.BFA bit to 1 disables MTIOC4C pin waveform output. Setting the MTU4.TMDR.BFB bit 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 MTU3.TMDR.BFA bit 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.



### 20.6.15 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the MTU4.TMDR.BFA bit and MTU4.TMDR.BFB bit to 0. Setting the MTU4.TMDR.BFA bit to 1 disables MTIOC4C pin waveform output. Setting the MTU4.TMDR.BFB bit 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 MTU3.TMDR.BFA bit and MTU3.TMDR.BFB bit. For example, if the MTU3.TMDR.BFA bit 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 20.118 shows an example of MTU3.TGR, MTU4.TGR, MTIOC3m, and MTIOC4m operation with the MTU3.TMDR.BFA bit and MTU3.TMDR.BFB bit set to 1 and the MTU4.TMDR.BFA bit and MTU4.TMDR.BFB bit set to 0. (m = A to D)

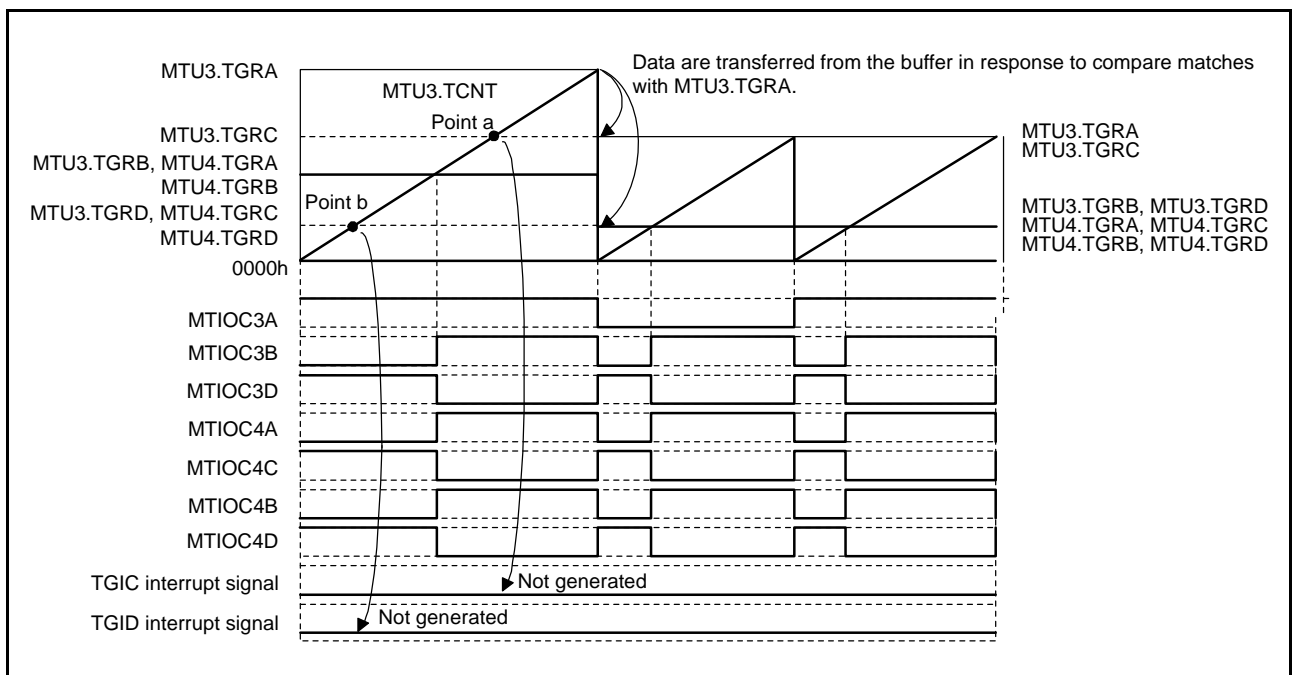


Figure 20.118 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

### 20.6.16 Overflow Flags in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT start counting when the TSTR.CST3 bit 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 20.119 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.

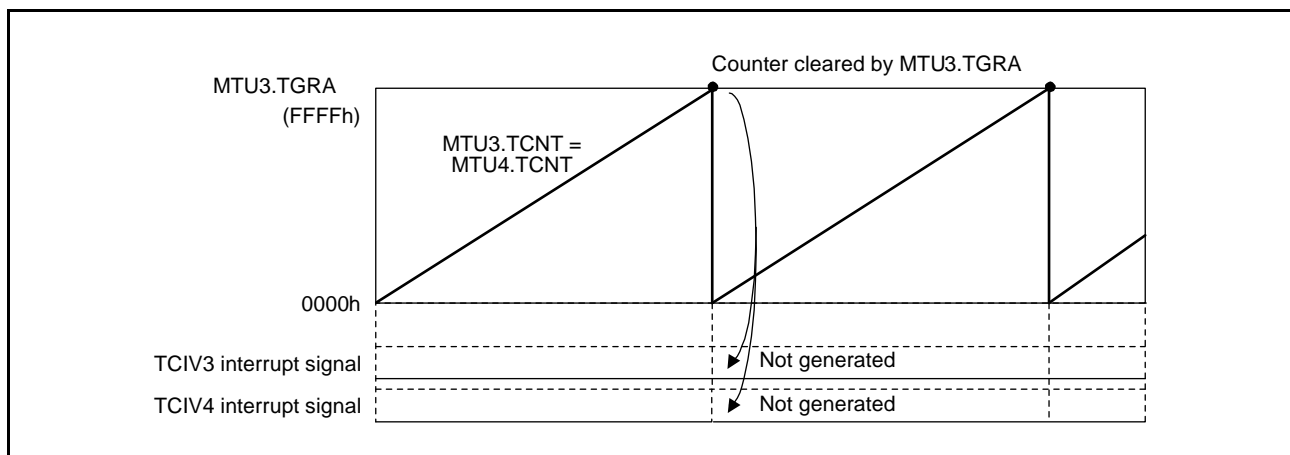


Figure 20.119 Overflow Flags in Reset-Synchronized PWM Mode

### 20.6.17 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. If an overflow and counter clearing due to an input capture occur simultaneously, an input capture interrupt signal is output and an overflow interrupt signal is not output.

Figure 20.120 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

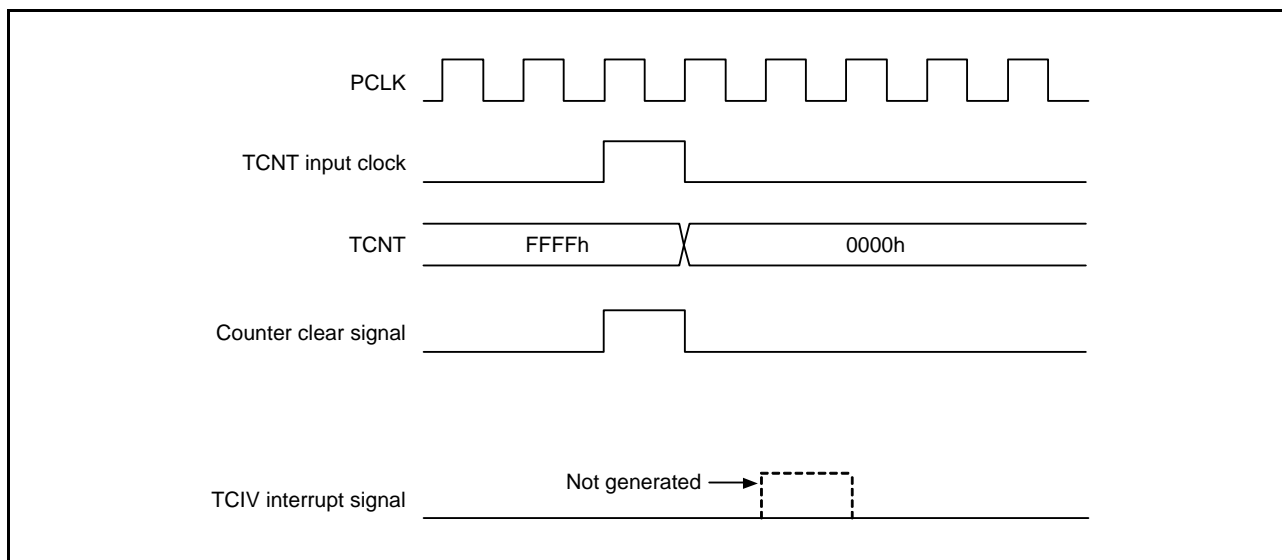


Figure 20.120 Contention between Overflow and Counter Clearing

### 20.6.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT up-count or down-count 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 20.121 shows the operation timing when there is contention between TCNT write operation and overflow.

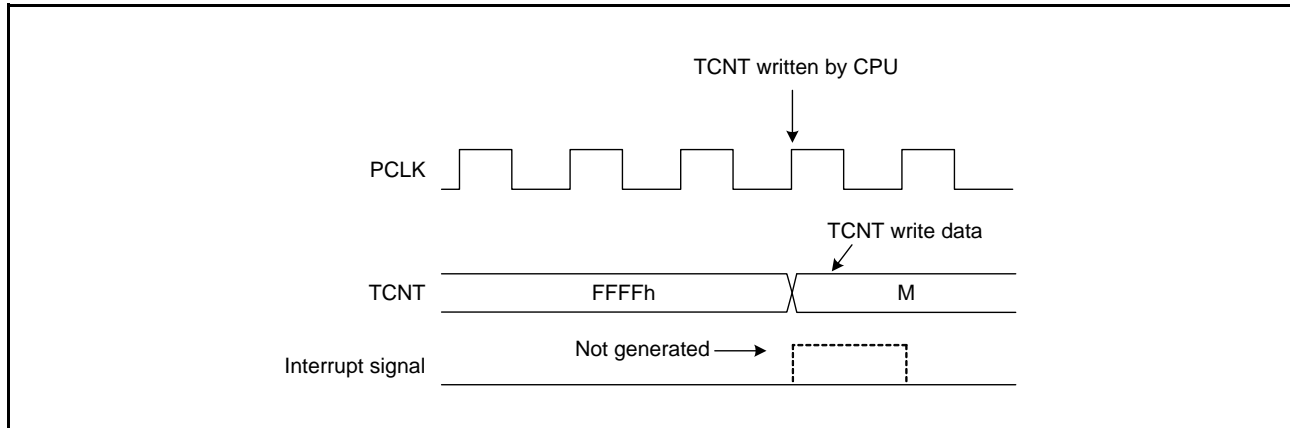


Figure 20.121 Contention between TCNT Write Operation and Overflow

### 20.6.19 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.

### 20.6.20 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 TOCR1.OLSP bit and TOCR1.OLSN bit to set the levels for PWM waveform output. Also, when either of these modes is in use, set TIOR to 00h. The negative-phase output level when the TDER.TDER bit is set to 0 (no dead time is generated) in complementary PWM mode is the inverse of the positive-phase output level according to the TOCR1.OLSP bit setting, not the TOCR1.OLSN bit setting.

### 20.6.21 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 DTC is not possible.

Accordingly, disable interrupts, etc. before making the settings for the module stop state.

### 20.6.22 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 function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input capture input. This function can be used to read the 32-bit counter such that MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, refer to section 20.2.8, Timer Input Capture Control Register (TICCR).

### 20.6.23 Notes When Complementary PWM Mode Output Protection Functions are Not Used

The complementary PWM mode output protection functions are initially enabled. Refer to section 21, Port Output Enable 2 (POE2a), for details.

### 20.6.24 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 (interrupt requests enabled), 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.

### 20.6.25 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 inverse-phase output pins beyond the period for active-level output.

Condition 1: In portion (10) of the initial output inhibition period in Figure 20.122, 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 20.123, 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).

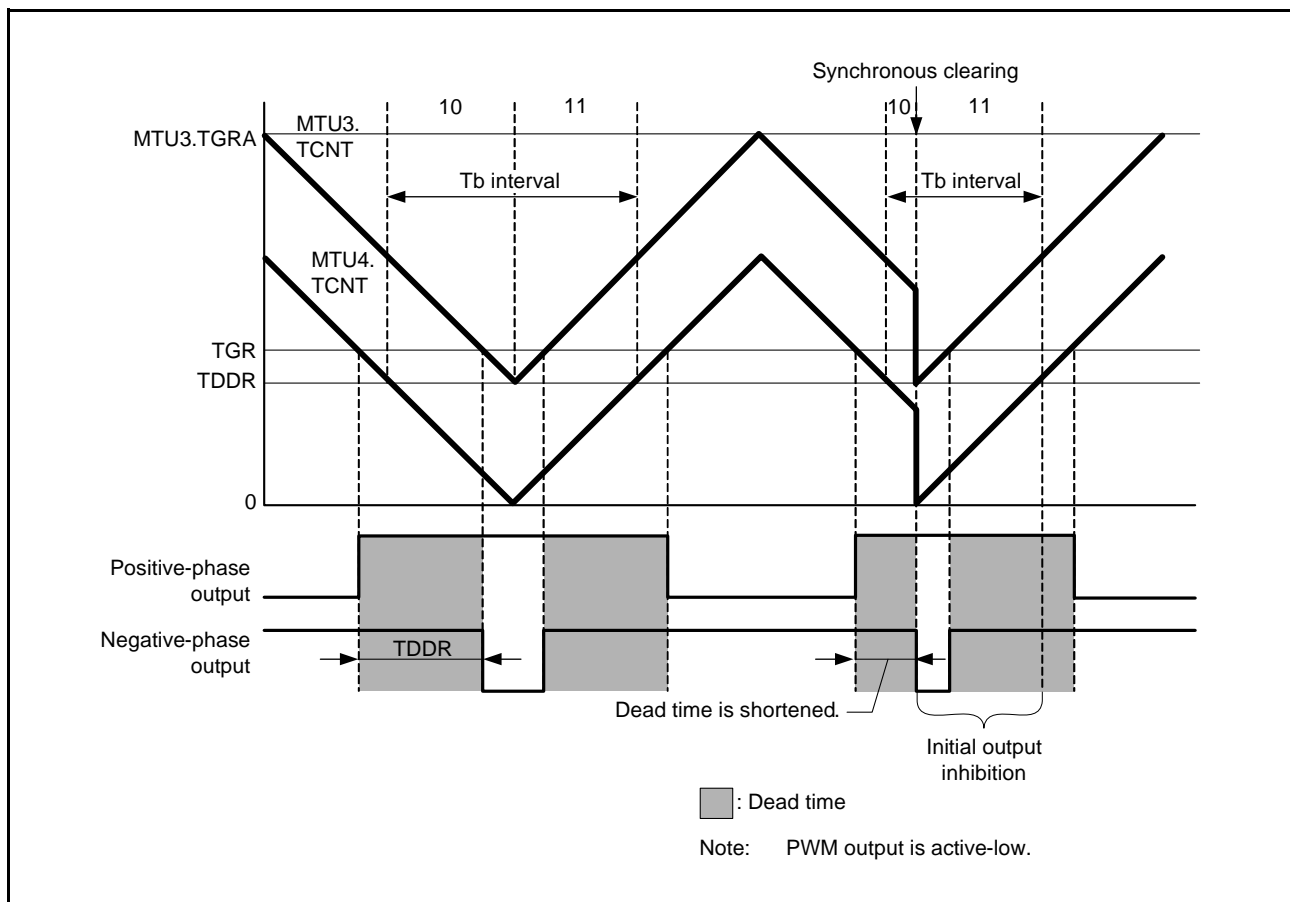


Figure 20.122 Example of Synchronous Clearing (When Condition 1 Applies)

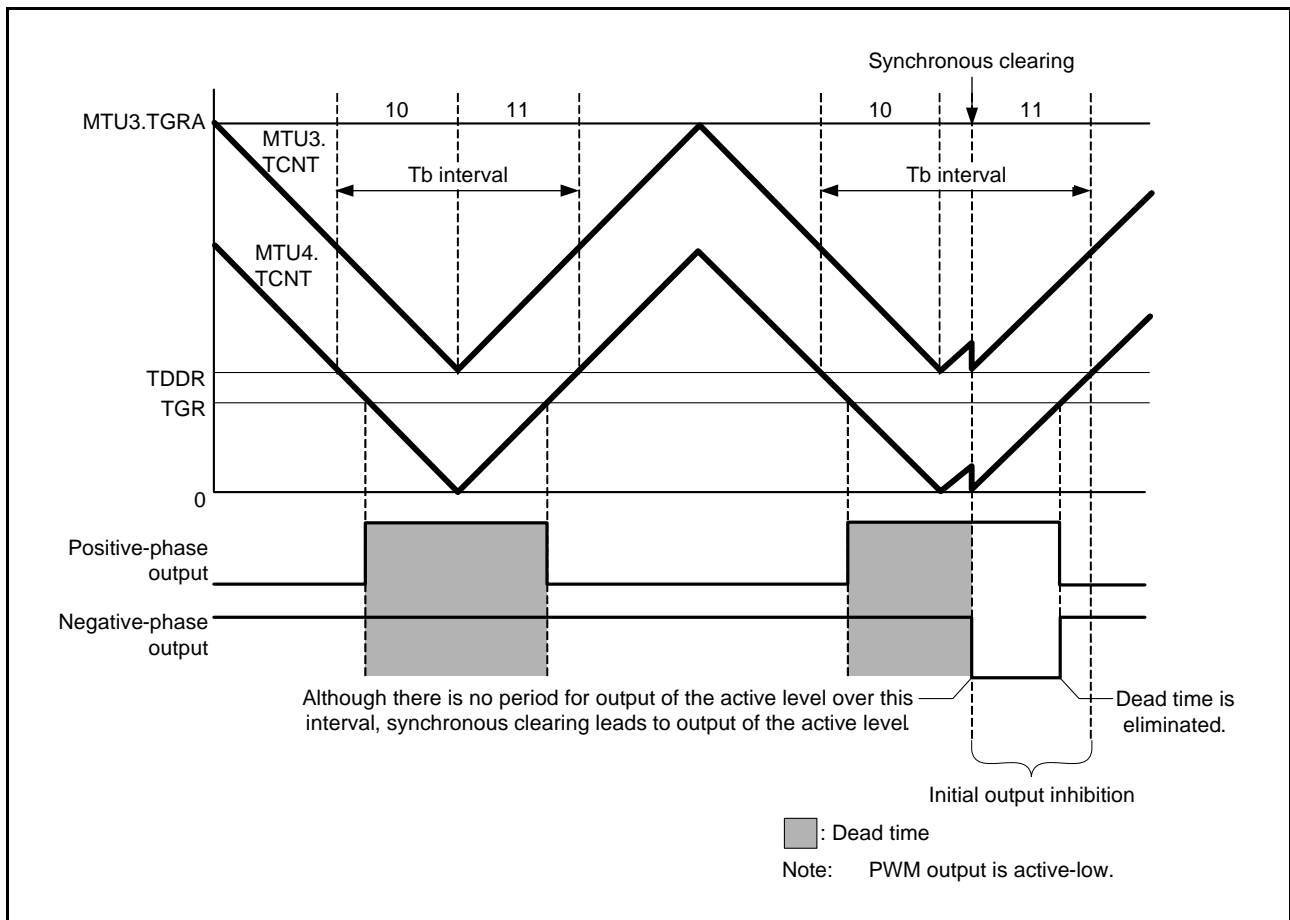


Figure 20.123 Example of Synchronous Clearing (When Condition 2 Applies)

### 20.6.26 Continuous Output of Interrupt Signal in Response to a Compare Match

When TGR is set to 0000h, PCLK/1 clock is set as the count clock, and compare match is set as the trigger for clearing of the count 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 20.124 shows the timing for continuous output of the interrupt signal in response to a compare match.

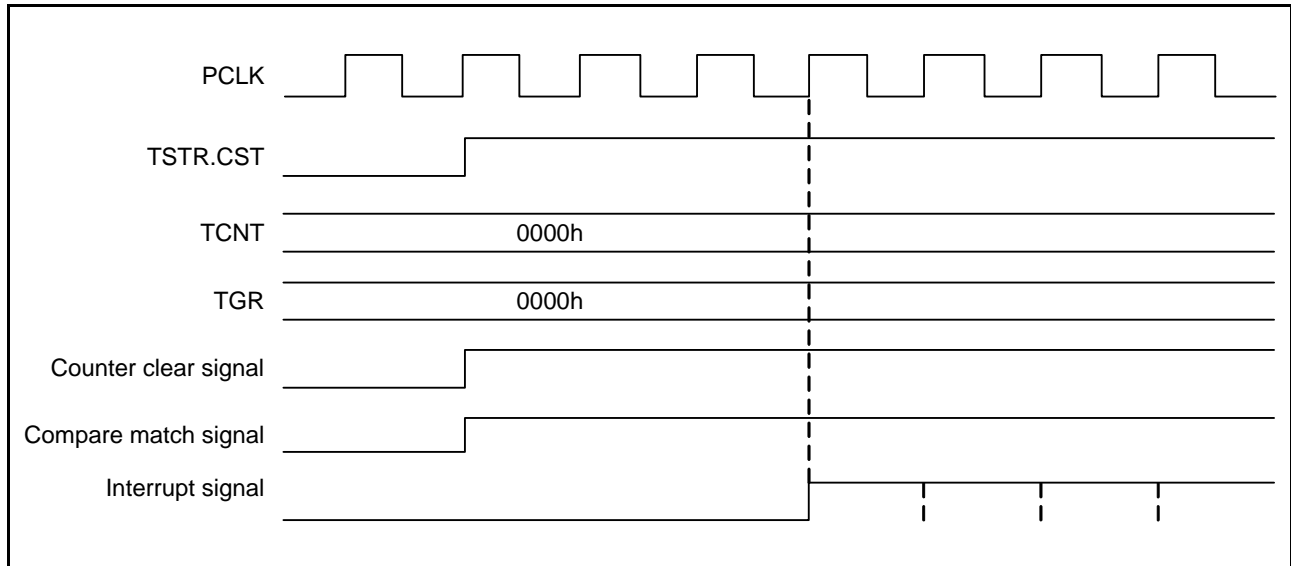


Figure 20.124 Continuous Output of Interrupt Signal in Response to a Compare Match



## 20.7 MTU Output Pin Initialization

### 20.7.1 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.

### 20.7.2 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 20.59.

Note that the following notations are used for operating modes.

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

**Table 20.59 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)

### 20.7.3 Overview of Pin Initialization Procedures and Mode Transitions in Case of Error during Operation

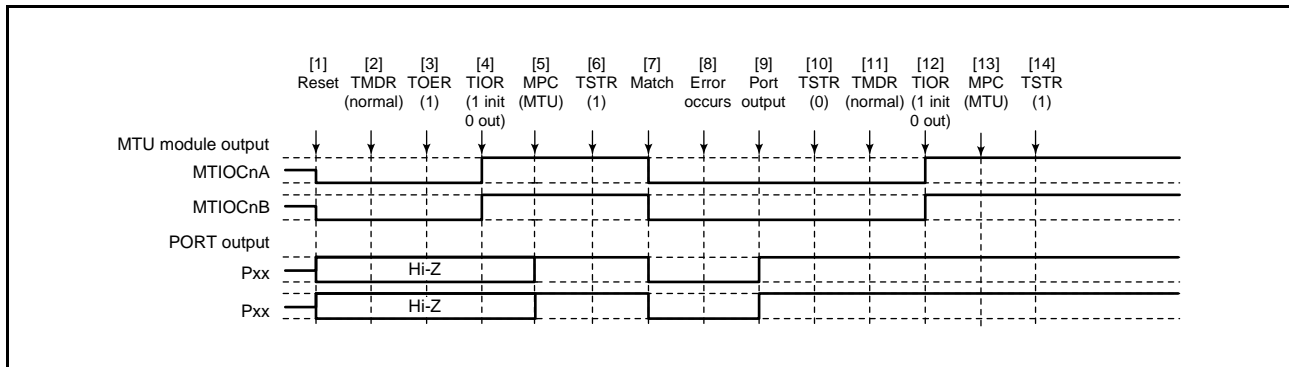
- When making a transition to a mode (Normal, PWM1, PWM2, or PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of TIOR setting.
- In PWM mode 1, a waveform is not output to the MTIOCnB and MTIOCnD (n = 3, 4) pins. If there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports.
- In PWM mode 2, a waveform is not output to the cycle register pins. If there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports.
- In normal mode or PWM 2 mode, if the TGRC and TGRD operate as buffer registers, a waveform is not output to the pins. If there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register and there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, temporarily disable output in MTU3 and MTU4 with the timer output master enable register (TOER). If there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports. Switch to normal mode, perform initialization with the TIOR register, and restore the TIOR register to its initial value. 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 unless otherwise specified.

Pin initialization procedures are described below for the numbered combinations in Table 20.59. The active level is assumed to be low.

## (1) Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 20.125 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

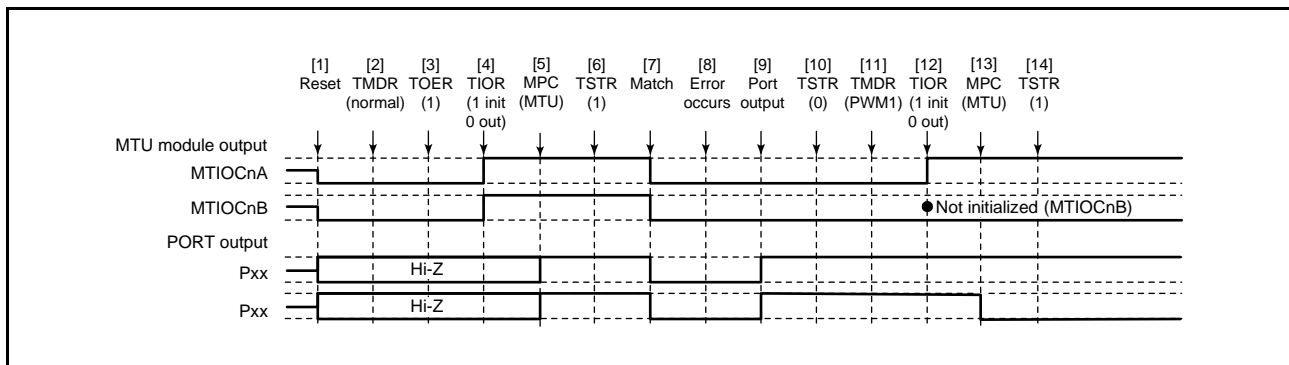


**Figure 20.125 Error Occurrence in Normal Mode, Recovery in Normal Mode**

- [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 20.126 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.



**Figure 20.126 Error Occurrence in Normal Mode, Recovery in PWM Mode 1**

[1] to [10] are the same as in Figure 20.125.

[11] Set PWM mode 1.

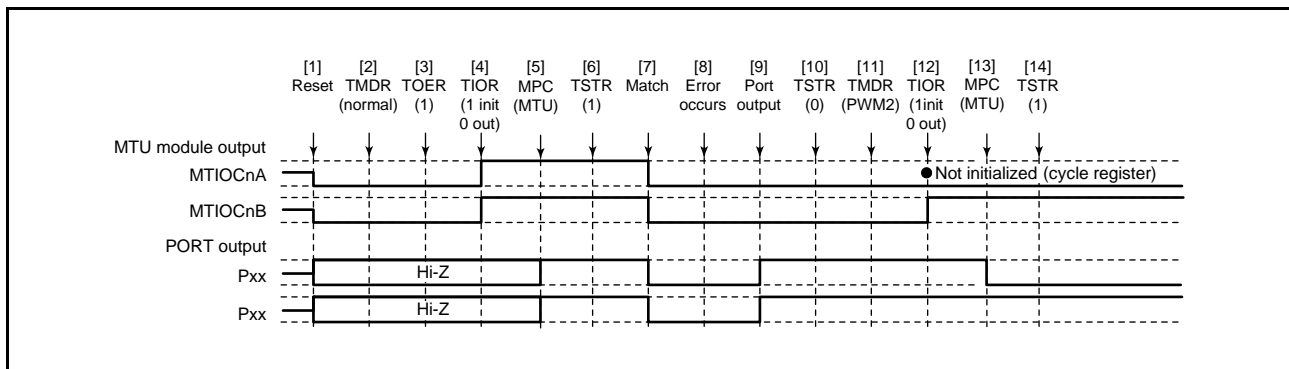
[12] Set the TIOR register to initialize pins, i.e. so that the MTIOCnB (or MTIOCnD) does not produce a waveform in PWM mode 1. If a particular level should be output, set the port direction register (PDR) and the port output data register (PODR) so that the pins of the I/O port operate as general outputs.

[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 20.127 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.



**Figure 20.127 Error Occurrence in Normal Mode, Recovery in PWM Mode 2**

[1] to [10] are the same as in Figure 20.125.

[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.)

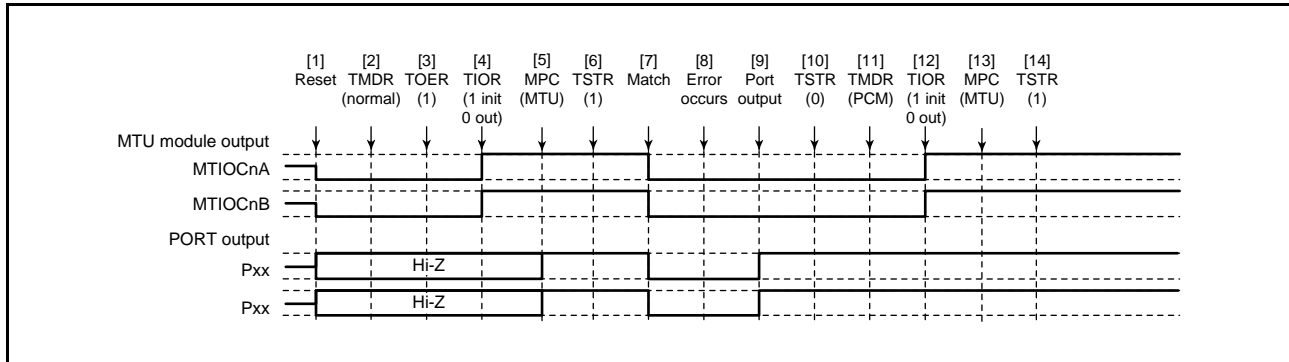
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[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 20.128 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.



**Figure 20.128 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode**

[1] to [10] are the same as in Figure 20.125.

[11] Set the phase counting mode.

[12] Initialize the pins with TIOR.

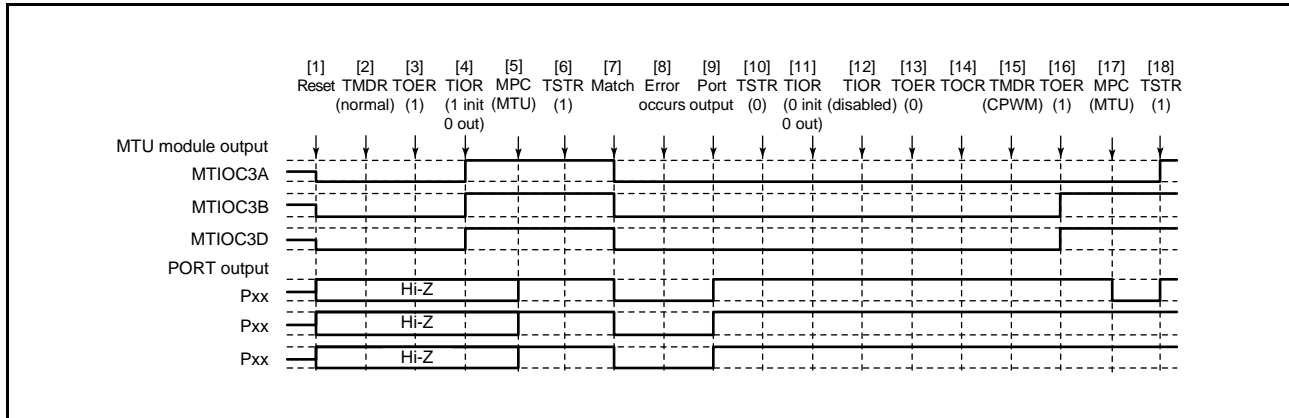
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[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 20.129 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.



**Figure 20.129 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode**

[1] to [10] are the same as in Figure 20.125.

[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.

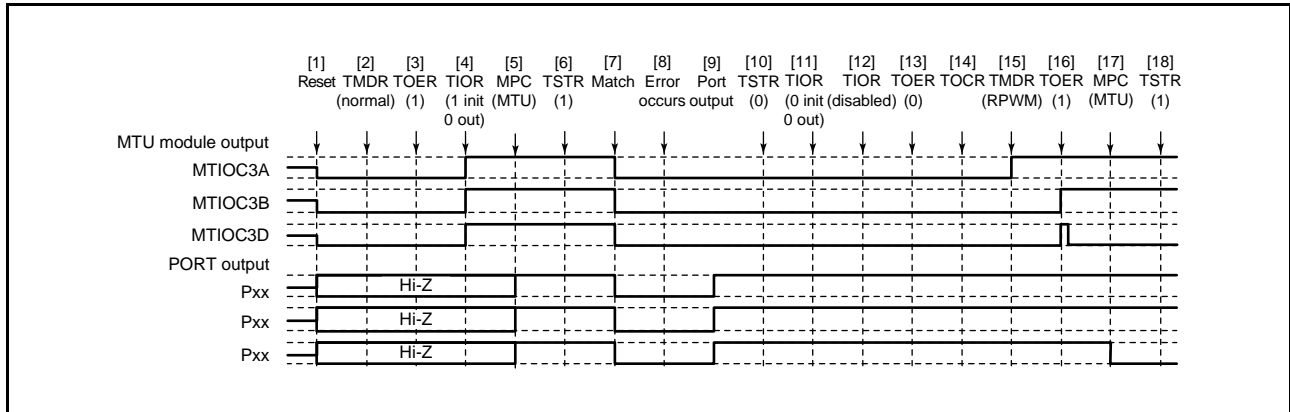
[16] Enable output in MTU3 and MTU4 with TOER.

[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 20.130 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.



**Figure 20.130 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode**

[1] to [13] are the same as in Figure 20.125.

[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.

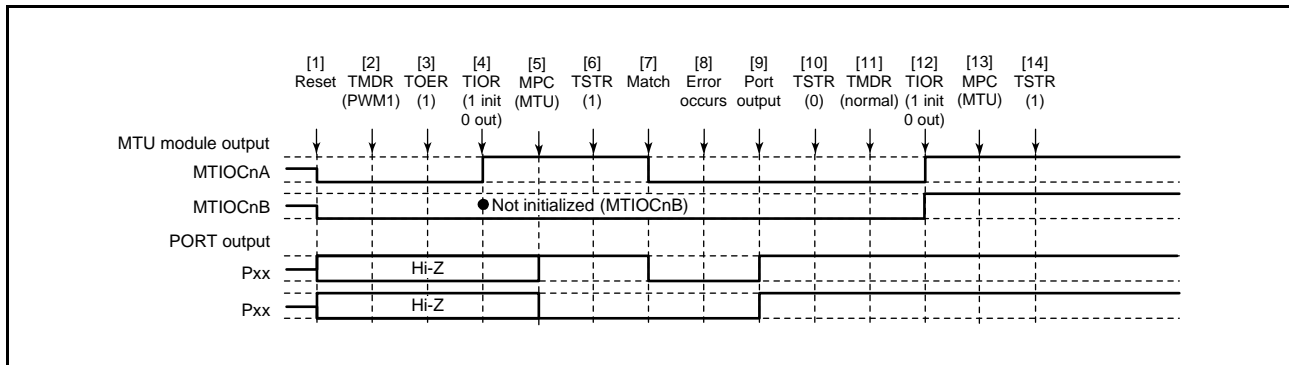
[17] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[18] Restart operation by setting TSTR.



## (7) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 20.131 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

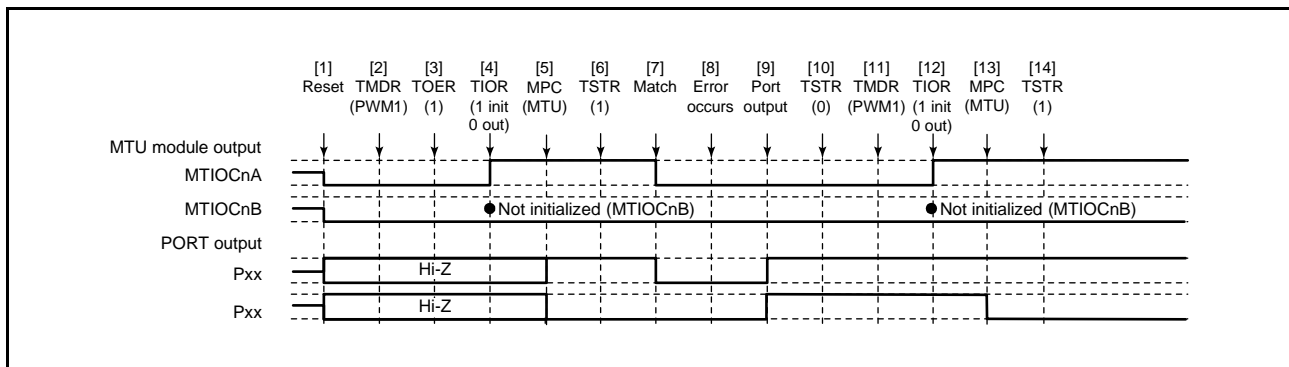


**Figure 20.131 Error Occurrence in PWM Mode 1, Recovery in Normal Mode**

- [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 20.132 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.



**Figure 20.132 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1**

[1] to [10] are the same as in Figure 20.131.

[11] This step is not necessary when restarting in PWM mode 1.

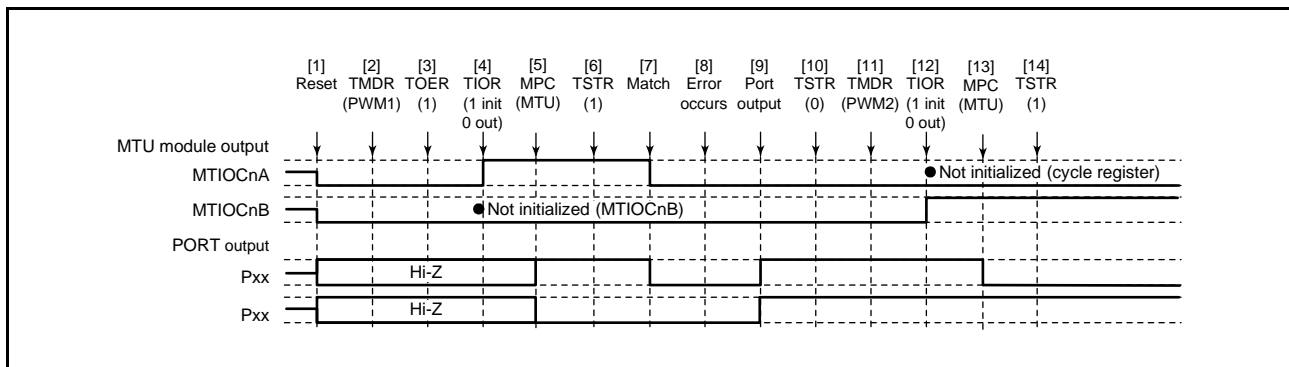
[12] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[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 20.133 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.



**Figure 20.133 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2**

[1] to [10] are the same as in Figure 20.131.

[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 20.134 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

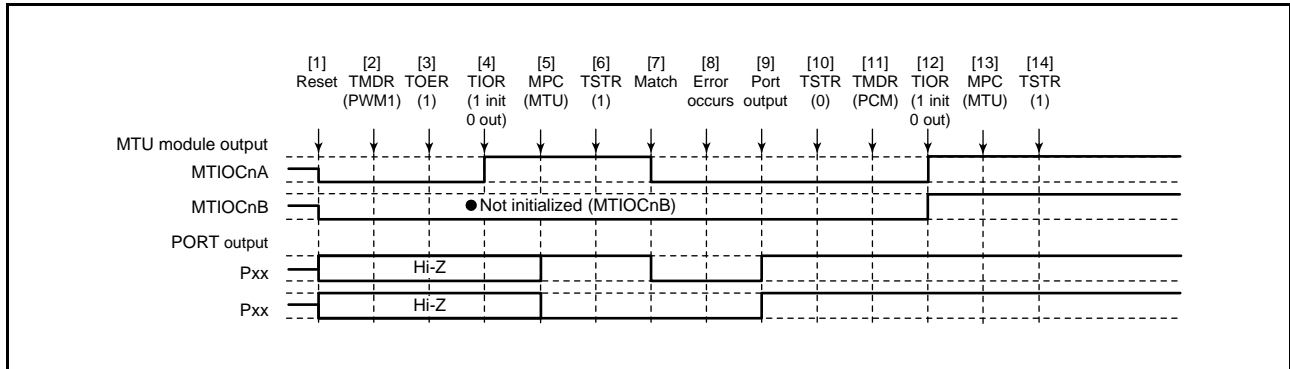


Figure 20.134 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

[1] to [10] are the same as in Figure 20.131.

[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 20.135 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

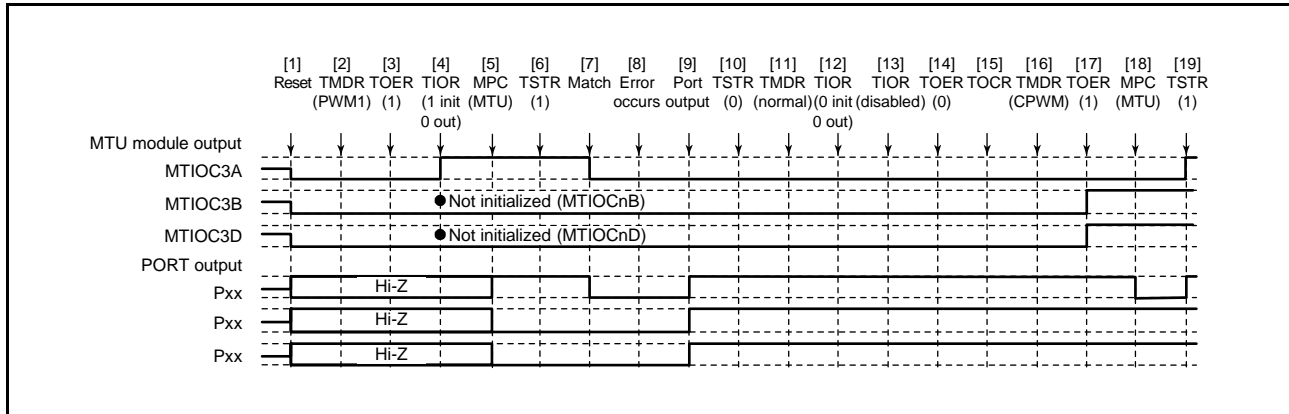


Figure 20.135 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 20.131.

[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 20.136 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

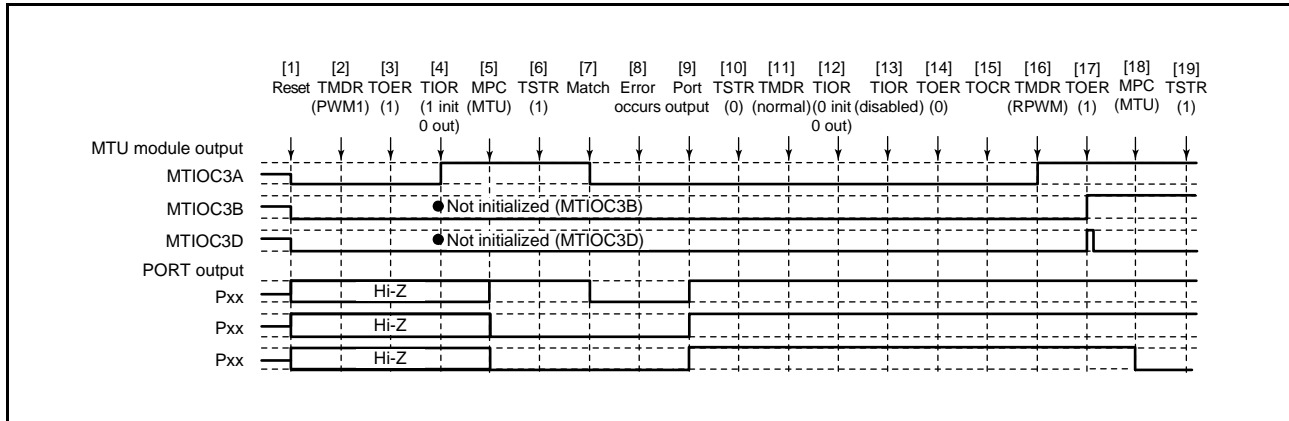


Figure 20.136 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

[1] to [14] are the same as in Figure 20.135.

[15] Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.

[16] Set reset-synchronized PWM mode.

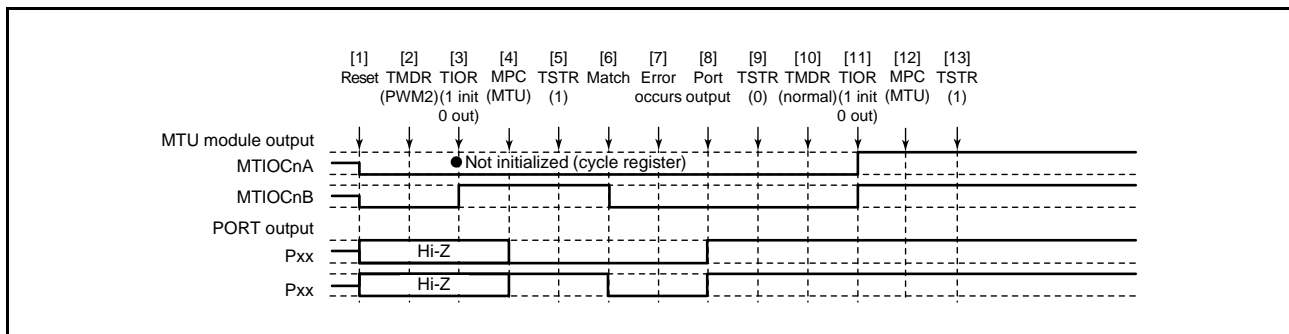
[17] Enable output in MTU3 and MTU4 with TOER.

[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 20.137 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.



**Figure 20.137 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 20.138 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

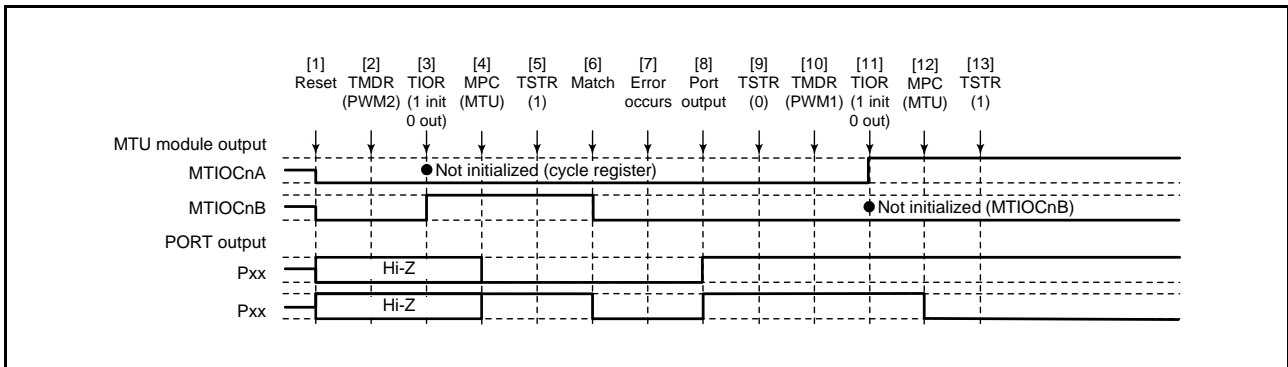


Figure 20.138 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

[1] to [9] are the same as in Figure 20.137.

[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 20.139 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

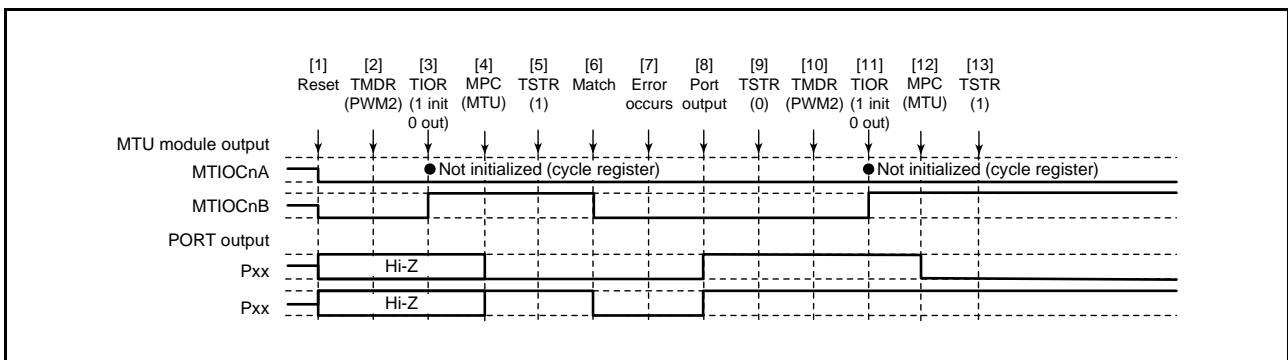


Figure 20.139 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

[1] to [9] are the same as in Figure 20.137.

[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 20.140 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

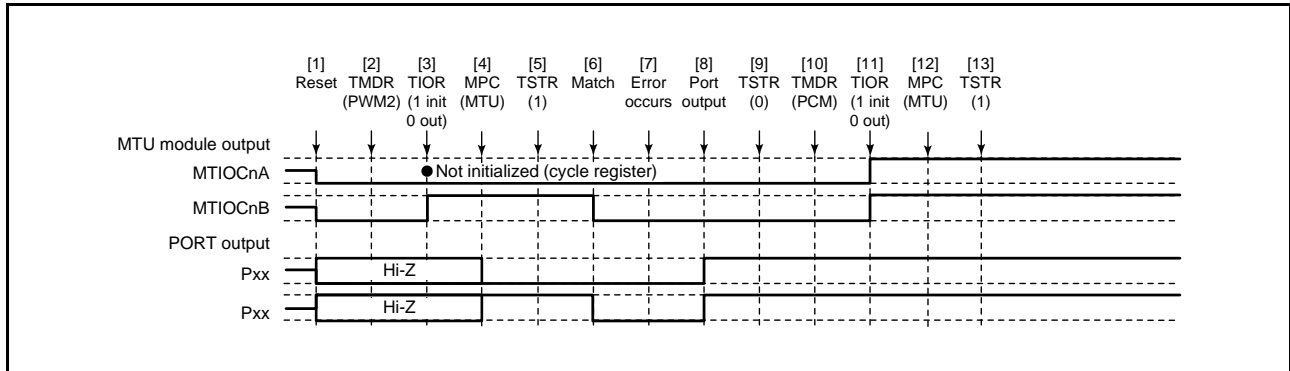


Figure 20.140 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

[1] to [9] are the same as in Figure 20.137.

[10] Set the phase counting mode.

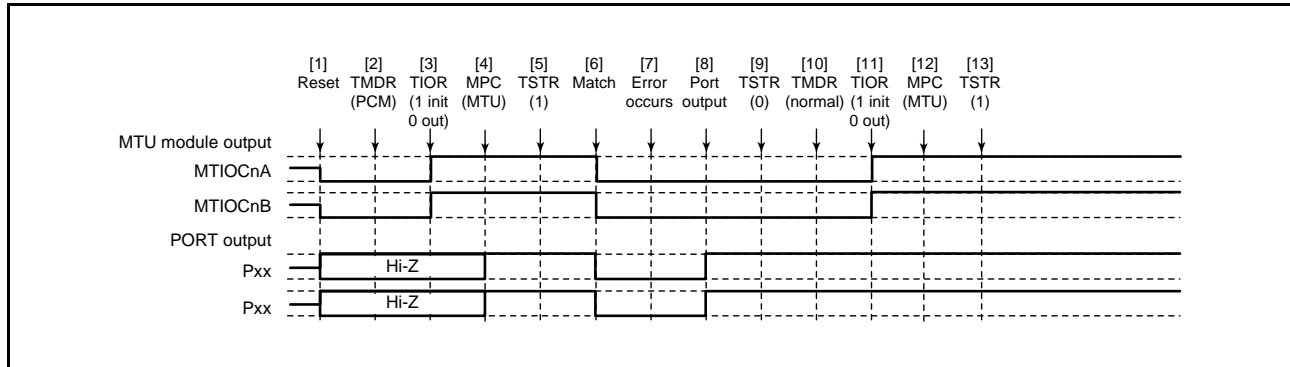
[11] Initialize the pins with TIOR.

[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 20.141 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.



**Figure 20.141 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 20.142 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

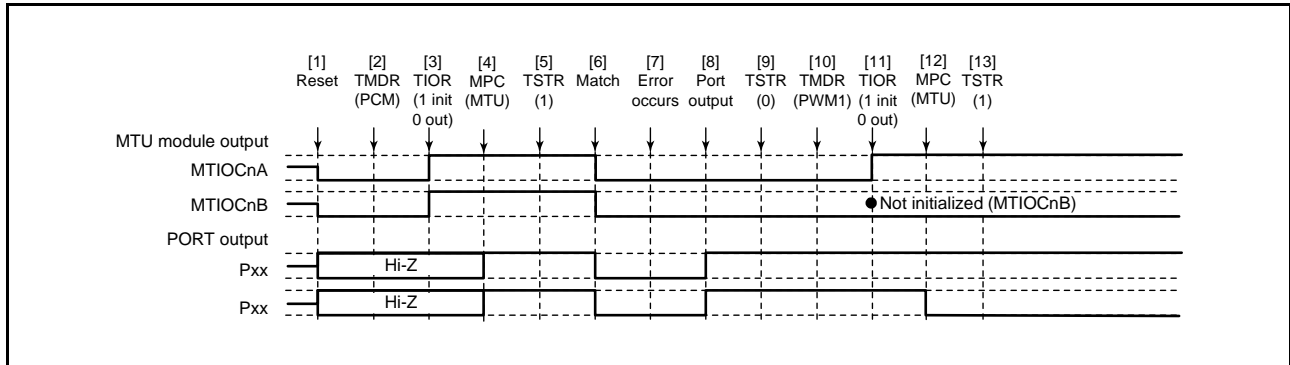


Figure 20.142 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

[1] to [9] are the same as in Figure 20.141.

[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 20.143 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

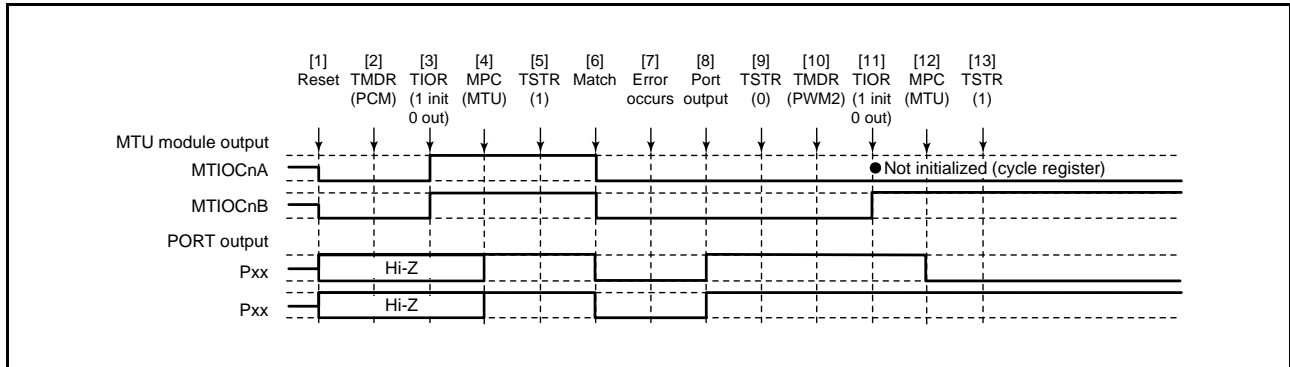


Figure 20.143 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

[1] to [9] are the same as in Figure 20.141.

[10] Set PWM mode 2.

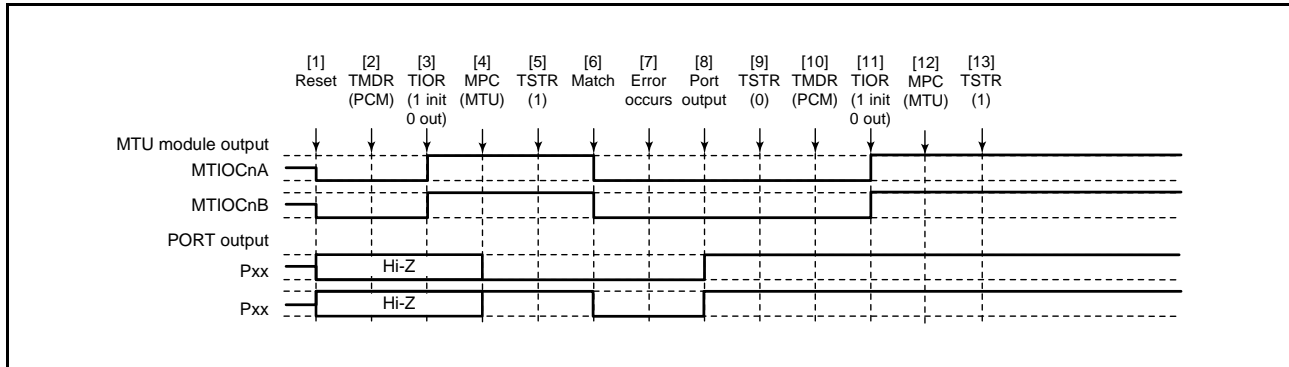
[11] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[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 20.144 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.



**Figure 20.144 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode**

[1] to [9] are the same as in Figure 20.141.

[10] This step is not necessary when restarting in phase counting mode.

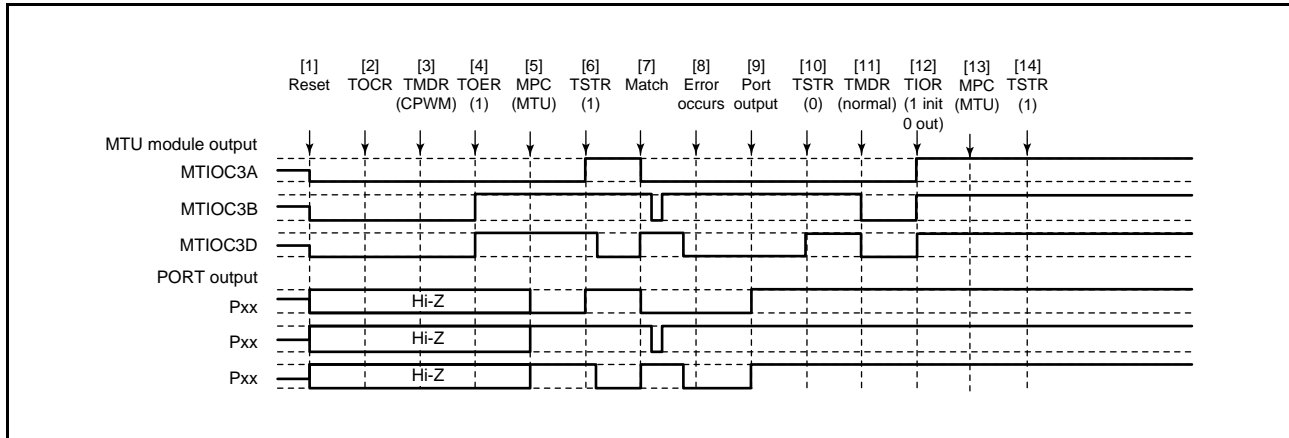
[11] Initialize the pins with TIOR.

[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 20.145 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

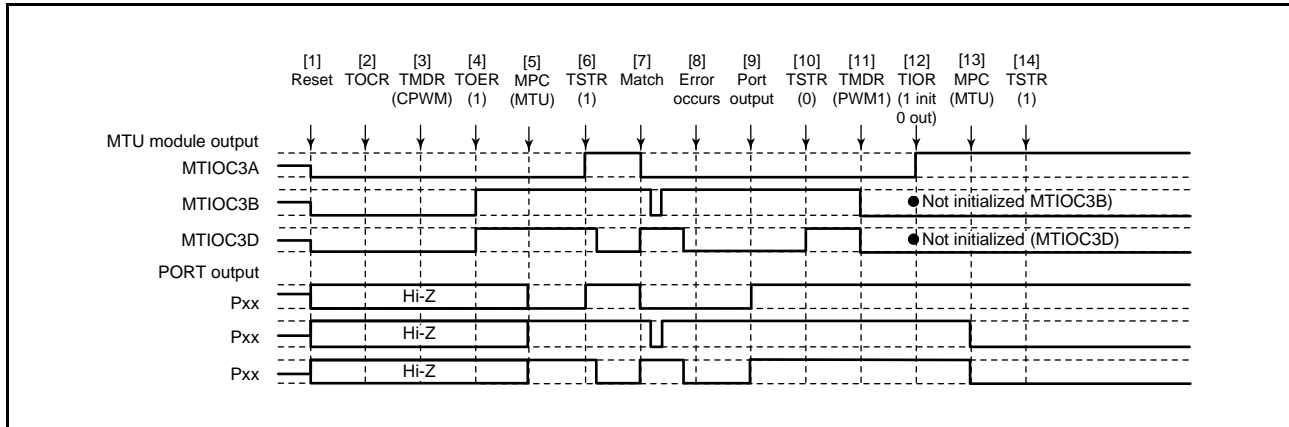


**Figure 20.145 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode**

- [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 20.146 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.



**Figure 20.146 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1**

[1] to [10] are the same as in Figure 20.145.

[11] Set PWM mode 1 (MTU output goes low).

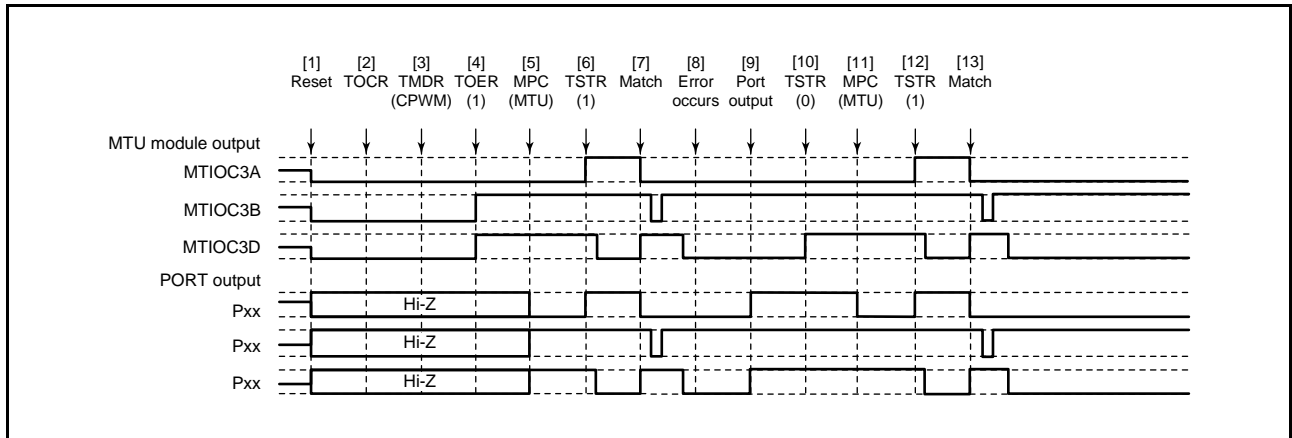
[12] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[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 20.147 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 20.147 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode**

[1] to [10] are the same as in Figure 20.145.

[11] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

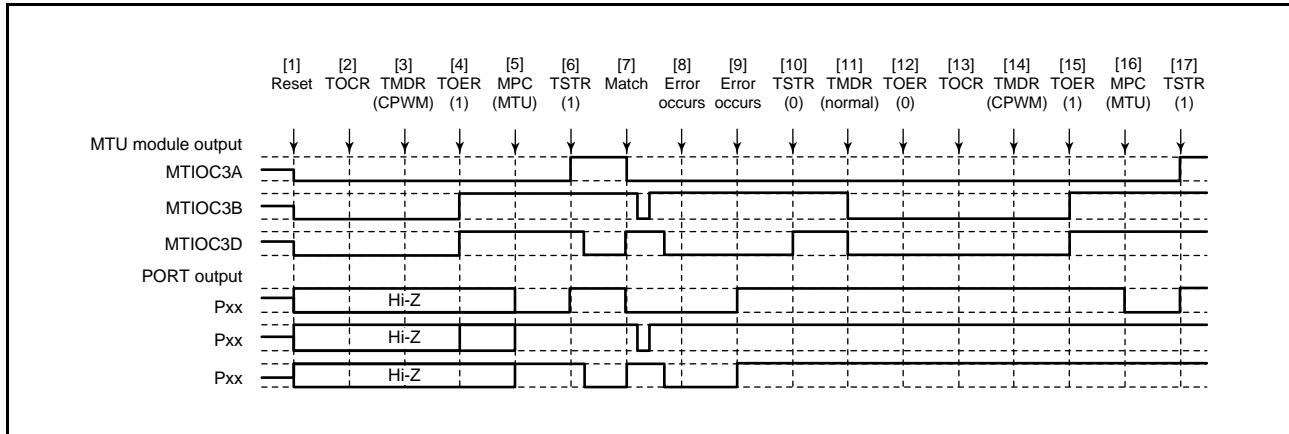
[12] Restart operation by setting TSTR.

[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 20.148 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 20.148 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode**

[1] to [10] are the same as in Figure 20.145.

[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.

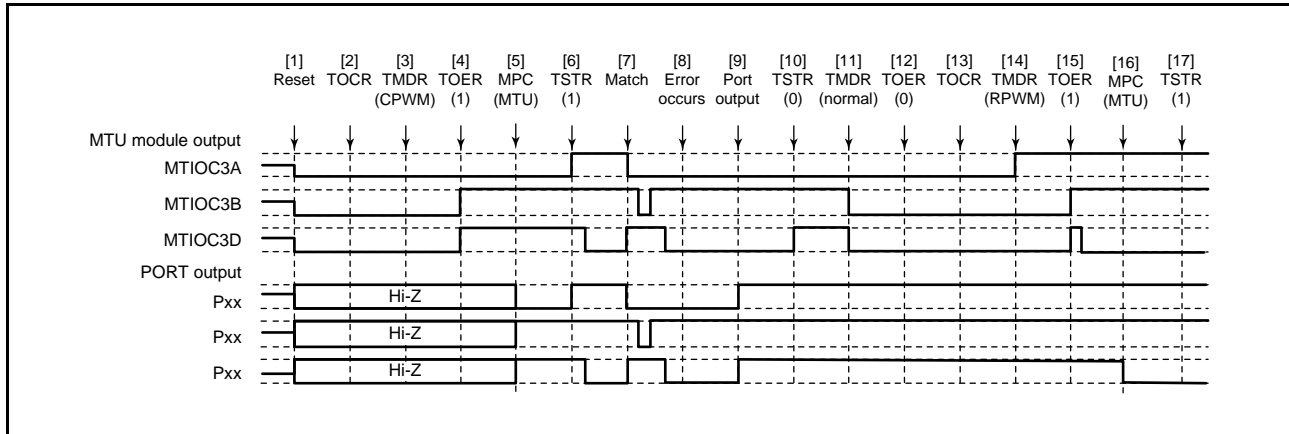
[15] Enable output in MTU3 and MTU4 with TOER.

[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 20.149 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 20.149 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode**

[1] to [10] are the same as in Figure 20.145.

[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.

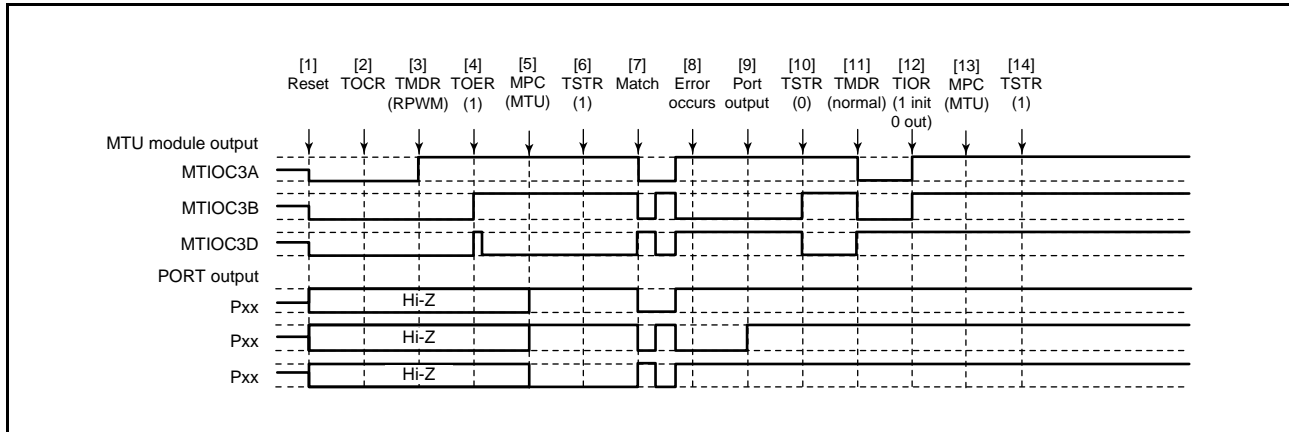
[15] Enable output in MTU3 and MTU4 with TOER.

[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 20.150 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

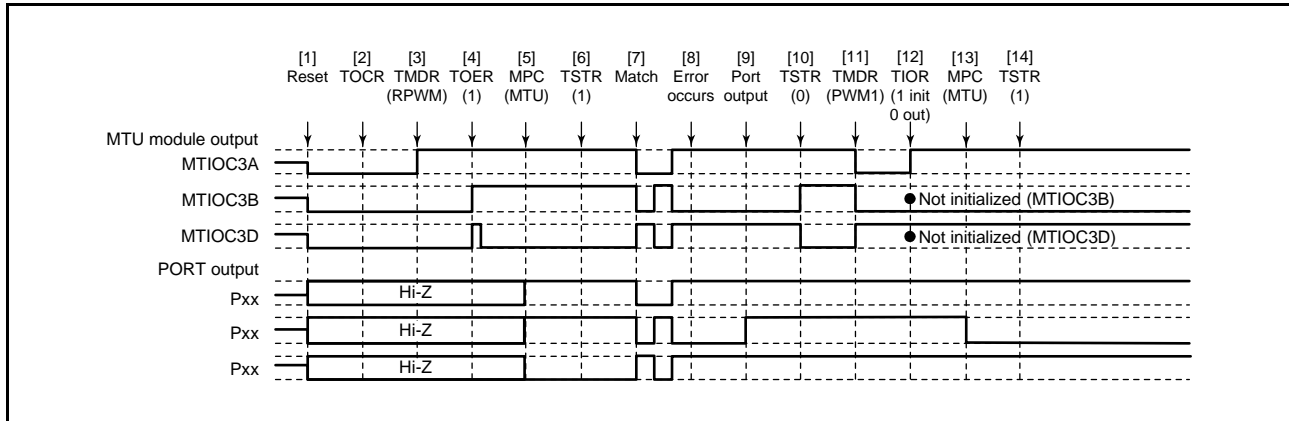


**Figure 20.150 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode**

- [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 20.151 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 20.151 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1**

[1] to [10] are the same as in Figure 20.150.

[11] Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).

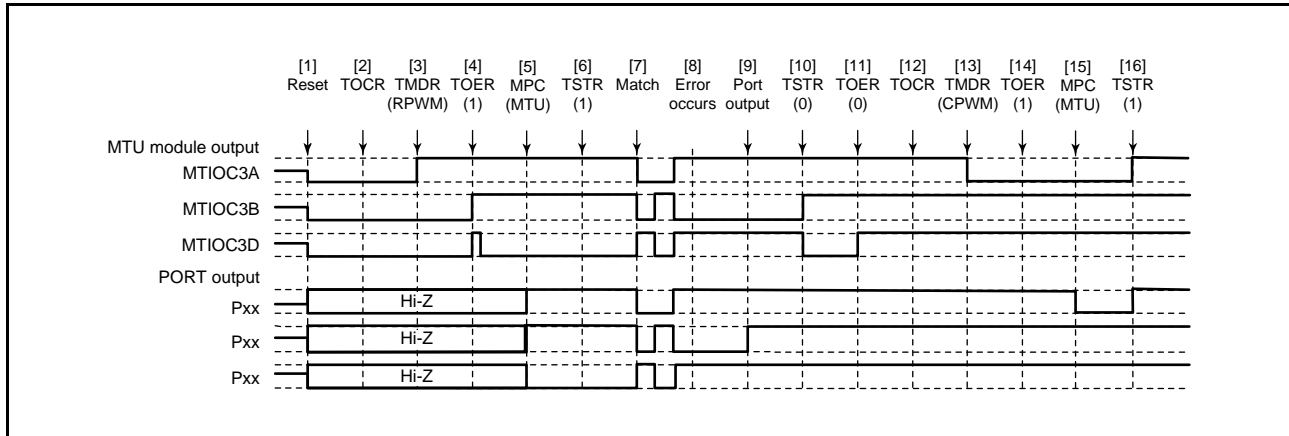
[12] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[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 20.152 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 20.152 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode**

[1] to [10] are the same as in Figure 20.150.

[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).

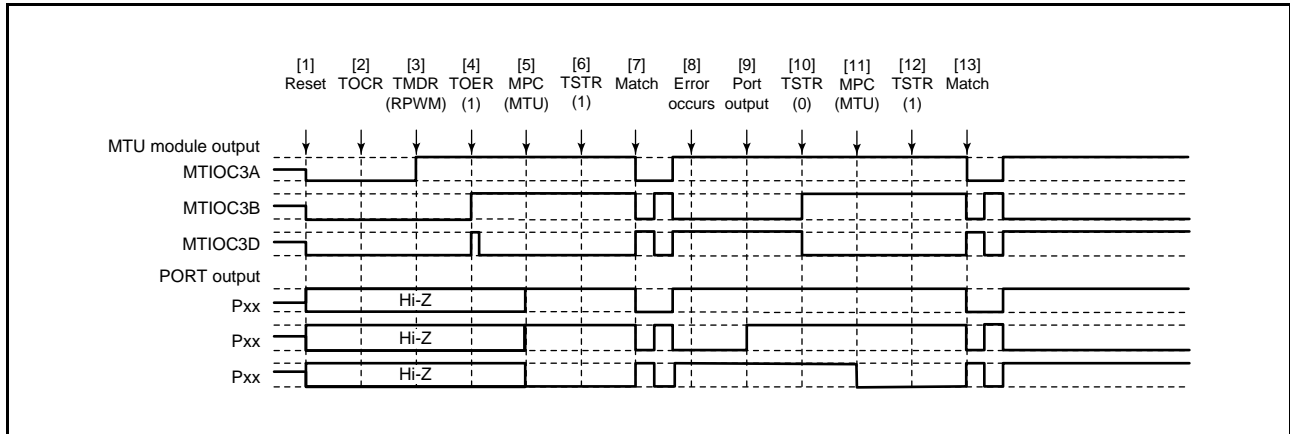
[14] Enable output in MTU3 and MTU4 with TOER.

[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 20.153 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 20.153 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode**

[1] to [10] are the same as in Figure 20.150.

[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.

## 20.8 Operations Linked by the ELC

### 20.8.1 Event Signal Output to the ELC

The MTU is capable of operation linked with another module set in advance when its interrupt request signal is used as an event signal by the event link controller (ELC).

The MTU outputs the event signal regardless of the setting of the corresponding interrupt request enable bit.

### 20.8.2 MTU Operations in Response to Receiving Event Signals from the ELC

The MTU can perform the following operations in response to the event set in advance in the ELSRn register of the event link controller (ELC).

#### (1) Count Start Operation

The MTU is selected the count start operation when using the ELOPA and ELOPB registers setting of the ELC. The ELOPA register functions to channels 1 to 3, and ELOPB register functions to channel 4. TMDR of the channel set by MTU should be set to the value after reset, 00h. When the specified event is generated by the ELSRn register, the TSTR.CSTn bit shown in Table 20.60 is set to 1, then the MTU counter is started.

However, when the specified event is generated while TSTR.CSTn bit is set to 1, the event is disabled. Table 20.60 lists the TSTR register bits used for each channel.

For details on the count start operation setting, refer to section 20.3.1, (1) Counter Operation.

**Table 20.60 Linkage Operating TSTR Register by the ELC**

Channel No.	TSTR register
MTU1	TSTR.CST1 bit
MTU2	TSTR.CST2 bit
MTU3	TSTR.CST3 bit
MTU4	TSTR.CST4 bit

#### (2) Input Capture Operation

The MTU is selected the input capture operation when using the ELOPA and ELOPB registers setting of the ELC. The ELOPA register handles channels 1 to 3, and ELOPB register handles channel 4. TMDR of the channel set by MTU should be set to the value after reset, 00h. When the specified event is generated by the ELSRn register, then the TCNT counter value capture to TGR register. When using the input capture operation, after setting the bit of MTU TIOR register to the input capture, TSTR.CSTn bit should be set to 1, and start the counter.

Then, the TIOcNA pin (input capture pin) input is disabled.

Table 20.61 lists the timer general register and timer I/O control register used in the input capture operation by the ELC. For details on the input capture setting, refer to section 20.3.1, (3) Input Capture Function.

**Table 20.61 Timer General Register and Timer I/O Control Register Used in the Input Capture Operation by the ELC**

Channel No.	Register Name	Bit Name of TIOR Register
MTU1	TGRA register	TIOR.IOA[3:0] bits
MTU2	TGRA register	TIOR.IOA[3:0] bits
MTU3	TGRA register	TIORH.IOA[3:0] bits
MTU4	TGRA register	TIORH.IOA[3:0] bits

### (3) Counter Restart Operation

The MTU is selected the count start operation when using the ELOPA and ELOPB registers setting of the ELC. The ELOPA register functions channels 1 to 3, and ELOPB register functions channel 4. TMDR of the channel set by MTU should be set to the value after reset, 00h. When the specified event is generated by the ELSRn register, then the TCNT (timer counter register) value is rewritten to initial value. When the CSTn bit in the TSTR register is 1, count operation can be continued. For details on the TSTR.CSTn bit, see Table 20.60.

## 20.8.3 Notes on MTU by Event Signal Reception from the ELC

The following describes usage notes when using MTU by the event link operation.

### (1) Count Start Operation

When the specified event is generated by the ELSRn register while write cycle is performed to the TSTR.CSTn bit, the write cycle is not performed to the TSTR.CSTn bit, and the setting to 1 takes precedence by generated event.

### (2) Count Restart Operation

When the specified event is generated by the ELSRn register while write cycle is performed to the TCNT counter, the write cycle is not performed to the TCNT counter, and count value initialization takes precedence by generated event.



## 21. 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.

In this section, “PCLK” is used to refer to PCLKB.

### 21.1 Overview

Table 21.1 lists the specifications of the POE, and Figure 21.1 shows a block diagram of the POE.

**Table 21.1 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 clock cycles 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 clock 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 can be 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 stop detection signal from the clock generation circuit, and a high-impedance request/interrupt request generating circuit as shown in Figure 21.1.

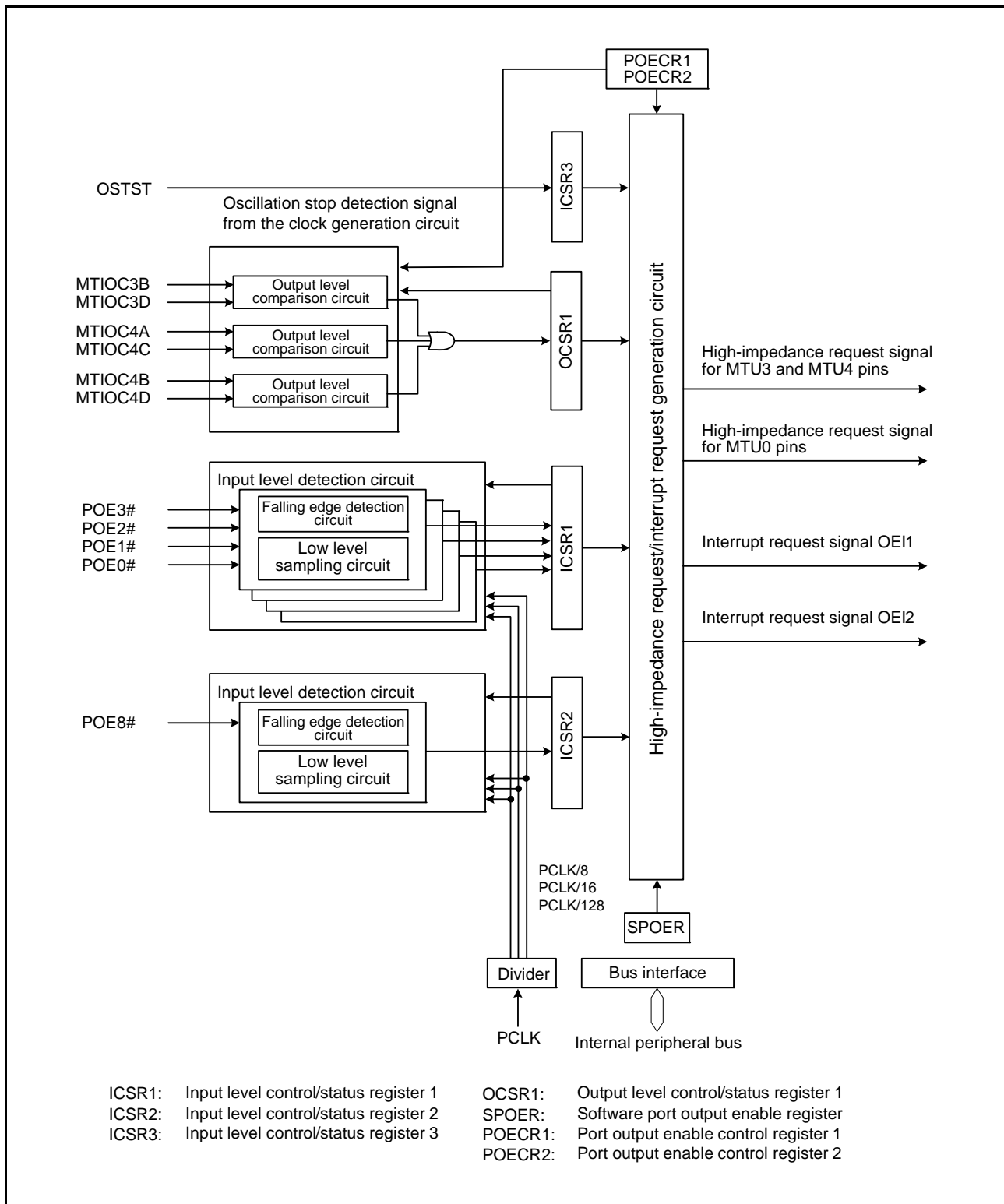


Figure 21.1 POE Block Diagram

Table 21.2 lists I/O pins to be used by the POE.

**Table 21.2 POE I/O Pins**

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 signal to place the MTU0 output 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 21.3 lists output-level comparisons with pin combinations.

**Table 21.3 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 PCLK clock cycles.
MTIOC4B and MTIOC4D	Output	(When the MTU.TOCR1.TOCS bit = 0: The active level is low level if the MTU.TOCR1.OLSP and OLSN bits are 0, and the active level is high level if the MTU.TOCR1.OLSP and OLSN bits are 1. When the MTU.TOCR1.TOCS bit = 1: The active level is low level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits are 0, and the active level is high level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits are 1.)

## 21.2 Register Descriptions

### 21.2.1 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 high-impedance request on the falling edge of the POE0# pin input. 0 1: Accepts a high-impedance request when the POE0# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when POE0# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a high-impedance 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 high-impedance request on the falling edge of the POE1# pin input. 0 1: Accepts a high-impedance request when the POE1# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when POE1# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a high-impedance 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 high-impedance request on the falling edge of the POE2# pin input. 0 1: Accepts a high-impedance request when the POE2# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when POE2# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a high-impedance 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 high-impedance request on the falling edge of the POE3# pin input. 0 1: Accepts a high-impedance request when the POE3# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when POE3# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a high-impedance 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: OE11 interrupt requests by the input level detection disabled 1: OE11 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, refer to section 21.3.5, 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.

[Setting condition]

- When the input set by POE0M[1:0] occurs at the POE0# pin

[Clearing condition]

- By writing 0 to POE0F after reading POE0F = 1

### POE1F Flag (POE1 Flag)

This flag indicates that a high-impedance request has been input to the POE1# pin.

[Setting condition]

- When the input set by POE1M[1:0] occurs at the POE1# pin

[Clearing condition]

- By writing 0 to POE1F after reading POE1F = 1

### POE2F Flag (POE2 Flag)

This flag indicates that a high-impedance request has been input to the POE2# pin.

[Setting condition]

- When the input set by POE2M[1:0] occurs at the POE2# pin

[Clearing condition]

- By writing 0 to POE2F after reading POE2F = 1

### POE3F Flag (POE3 Flag)

This flag indicates that a high-impedance request has been input to the POE3# pin.

[Setting condition]

- When the input set by POE3M[1:0] occurs at the POE3# pin

[Clearing condition]

- By writing 0 to POE3F after reading POE3F = 1

## 21.2.2 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 21.3 has simultaneously become an active level. If the POE2.PnCZEA (n = 1, 2, 3) bits 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.

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level

[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, refer to section 21.3.5, Release from the High-Impedance.

### 21.2.3 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 high-impedance request on the falling edge of the POE8# pin input 0 1: Accepts a high-impedance request when the POE8# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level. 1 0: Accepts a high-impedance request when the POE8# pin input has been sampled 16 times at PCLK/16 clock cycles and all are low level. 1 1: Accepts a high-impedance request when the POE8# pin input has been sampled 16 times at PCLK/128 clock cycles 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.

[Setting condition]

- When the input set by ICSR2.POE8M[1:0] bits occurs at the POE8# pin

[Clearing conditions]

- Writing 0 to POE8F after reading POE8F = 1  
When writing 0 to the flag while low-level sampling is selected for the ICSR2.POE8M[1:0] bits, the POE8# pin input must be at the high level.  
For details, refer to section 21.3.5, Release from the High-Impedance.

## 21.2.4 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 selects whether to place the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) in high-impedance.

[Setting condition]

- By writing 1 to CH34HIZ

[Clearing condition]

- By writing 0 to CH34HIZ after reading CH34HIZ = 1

### CH0HIZ Bit (MTU0 Output High-Impedance Enable)

This bit selects whether to place the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D) in high-impedance.

[Setting condition]

- By writing 1 to CH0HIZ

[Clearing condition]

- By writing 0 to CH0HIZ after reading CH0HIZ = 1



### 21.2.5 Port Output Enable Control Register 1 (POE2CR1)

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.

## 21.2.6 Port Output Enable Control Register 2 (POECR2)

Address(es): 0008 890Ch

b7	b6	b5	b4	b3	b2	b1	b0
—	P1CZEA	P2CZEA	P3CZEA	—	—	—	—
0	1	1	1	0	0	0	0

Value after reset:

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.

## 21.2.7 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	—	—	—	—	—	—	—	—	—
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
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 will not clear the flag while the oscillation-stopped detection signal is being asserted; in other words, it will not clear the flag before 10 PCLK clock cycles have elapsed after stopped oscillation was detected.

[Setting condition]

- Detection of the oscillation-stopped state

[Clearing condition]

- Writing 0 to the bit after having read its value as 1.

## 21.3 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 the ICSR2.POE8F flag is set to 1 with POECR1.PE0ZE and ICSR2.POE8E set to 1.
- SPOER setting  
When the SPOER.CH0HIZ bit is set to 1 with POECR1.PE0ZE set to 1.
- Detection of stopped oscillation  
When the 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 the ICSR2.POE8F flag is set to 1 with POECR1.PE1ZE and ICSR2.POE8E set to 1.
- SPOER setting  
When the SPOER.CH0HIZ bit is set to 1 with POECR1.PE1ZE set to 1.
- Detection of stopped oscillation  
When the 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 the ICSR2.POE8F flag is set to 1 with POECR1.PE2ZE and ICSR2.POE8E set to 1.
- SPOER setting  
When the SPOER.CH0HIZ bit is set to 1 with POECR1.PE2ZE set to 1.
- Detection of stopped oscillation  
When the 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 the ICSR2.POE8F flag is set to 1 with POECR1.PE3ZE and ICSR2.POE8E set to 1.
- SPOER setting  
When the SPOER.CH0HIZ bit is set to 1 with POECR1.PE3ZE set to 1.
- Detection of stopped oscillation  
When the 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 the ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POE2CR2.P1CZEA set to 1.
- MTIOC3B and MTIOC3D output level comparison  
When the OCSR1.OSF1 flag is set to 1 with POE2CR2.P1CZEA and OCSR1.OCE1 set to 1.
- SPOER setting  
When the SPOER.CH34HIZ bit is set to 1 with POE2CR2.P1CZEA set to 1.
- Detection of stopped oscillation  
When the ICSR3.OSTSTF flag is set to 1 with POE2CR2.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 the ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POE2CR2.P2CZEA set to 1.
- MTIOC4A and MTIOC4C output level comparison  
When the OCSR1.OSF1 flag is set to 1 with POE2CR2.P2CZEA and OCSR1.OCE1 set to 1.
- SPOER setting  
When the SPOER.CH34HIZ bit is set to 1 with POE2CR2.P2CZEA set to 1.
- Detection of stopped oscillation  
When the ICSR3.OSTSTF flag is set to 1 with POE2CR2.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 the ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POE2CR2.P3CZEA set to 1.
- MTIOC4B and MTIOC4D output level comparison  
When the OCSR1.OSF1 flag is set to 1 with POE2CR2.P3CZEA and OCSR1.OCE1 set to 1.
- SPOER setting  
When the SPOER.CH34HIZ bit is set to 1 with POE2CR2.P3CZEA set to 1.
- Detection of stopped oscillation  
When the ICSR3.OSTSTF flag is set to 1 with POE2CR2.P3CZEA and ICSR3.OSTSTE set to 1.

### 21.3.1 Input Level Detection Operation

If the input conditions set by the ICSR1 and ICSR2 registers 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 PCLK cycle, whether the falling edge will or will not be detected cannot be guaranteed. Figure 21.2 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.

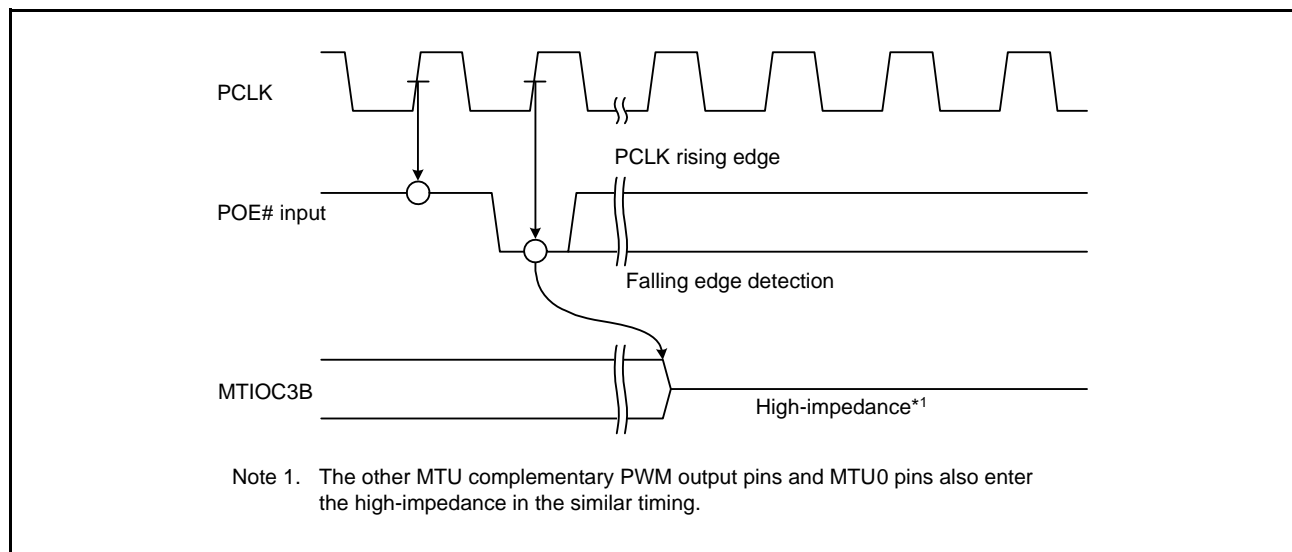


Figure 21.2 Falling Edge Detection

(2) Low-Level Detection

Figure 21.3 shows the low-level detection operation. When a low level is detected 16 times continuously with the sampling clock selected by the ICSR1 and ICSR2 registers, the detected level is recognized as low, and the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance. If even one high level is detected during this interval, the detected level is not recognized as low. 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.

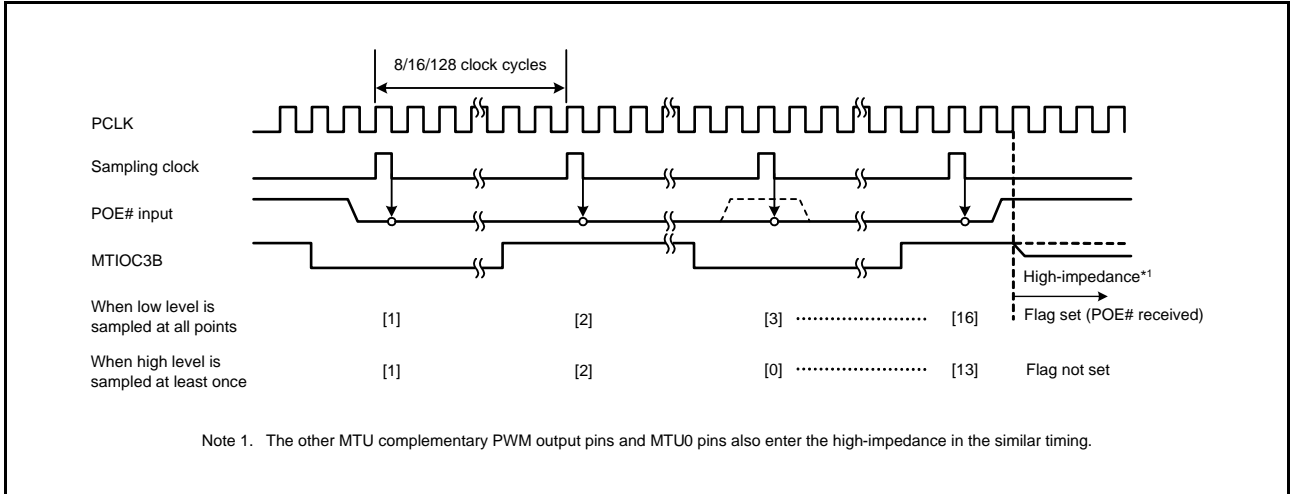


Figure 21.3 Low-Level Detection Operation

21.3.2 Output-Level Compare Operation

Figure 21.4 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.

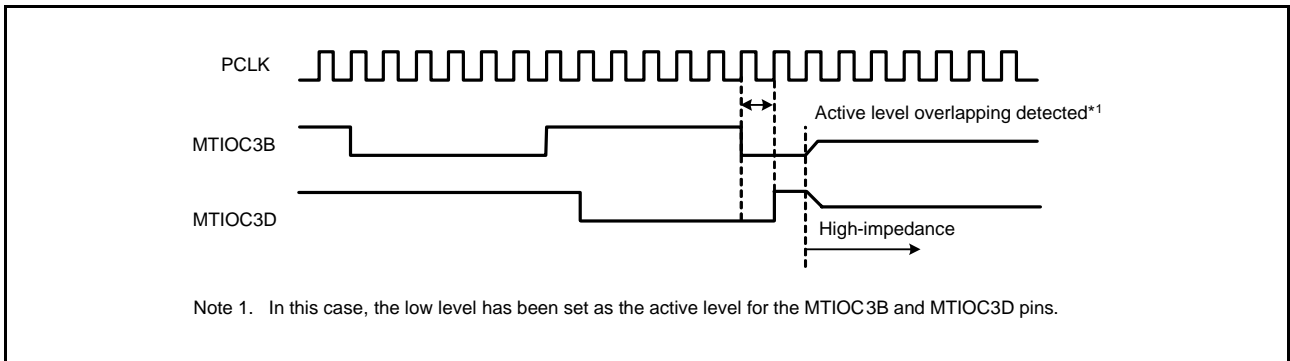


Figure 21.4 Output-Level Compare Operation

### 21.3.3 High-Impedance Control Using Registers

The high-impedance of the MTU complementary PWM output and MTU0 pins can be directly controlled by writing to the software port output enable register (SPOER).

Setting the SPOER.CH34HIZ bit to 1 places the MTU complementary PWM output pins (MTU3 and MTU4) specified by the POE2R register in the high-impedance.

Setting the SPOER.CH0HIZ bit to 1 places the MTU0 output pins specified by port output enable control register 1 (POE2R1) in the high-impedance.

### 21.3.4 High-Impedance Control on Detection of Stopped Oscillation

When the oscillation stop detection function in the clock generation circuit detects stopped oscillation while the ICSR3.OSTSTE bit is 1, the MTU complementary PWM output pins specified by the POE2R register and the MTU0 output pins specified by the POE2R1 register are placed in the high-impedance.

### 21.3.5 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 ICSR1.POE3F to POE0F flags and the ICSR2.POE8F flag. Note, however, that when low-level sampling is selected by the ICSR1.POE3M[1:0], POE2M[1:0], POE1M[1:0], and POE0M[1:0] bits, and the ICSR2.POE8M[1:0] bits, if a high level is being input to the corresponding pin from among POE0# to POE3# and POE8# but has not yet been detected, 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 OCSR1.OSF1 flag. 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 ICSR3.OSTSTF or ICSR3.OSTSTE bit 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.



## 21.4 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 21.4 lists 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 21.4 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.

## 21.5 Usage Notes

### 21.5.1 Transitions to Software Standby Mode

When the POE is used, do not make a transition to software standby mode. In this mode, the POE stops and thus the high-impedance of pins cannot be controlled.

### 21.5.2 When the POE Is Not Used

When the POE is not used, write 00h to port output enable control registers 1 and 2 (POECR1 and POECR2), respectively.

### 21.5.3 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 PMR and PmnPFS registers. When the pin has been specified as a general I/O pin, the POE does not control high-impedance outputs.

When using POE0#, POE1#, POE2#, POE3#, or POE8# for detecting the input level, set the pin function as a POE input pin by registers PMR and PmnPFS and clear flags ICSR1.POEkF and ICSR2.POE8F before starting the MTU (k = 0 to 3).

## 22. 8-Bit Timer (TMR)

This MCU 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 multi-function 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.

In this section, “PCLK” is used to refer to PCLKB.

### 22.1 Overview

Table 22.1 lists the specifications of the TMR.

Figure 22.1 shows a block diagram of the 8-bit timer module (unit 0), and Figure 22.2 shows that of the 8-bit timer module (unit 1).

**Table 22.1 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
Event link function (Output)	Compare match A, compare match B, and overflow (TMR0, TMR2)
Event link function (Input)	One of the following three operations proceeds in response to an event reception: <ol style="list-style-type: none"> <li>(1) Counting start operation (TMR0, TMR2)</li> <li>(2) Event counting operation (TMR0, TMR2)</li> <li>(3) Counting restart operation (TMR0, TMR2)</li> </ol>
DTC activation	DTC can be activated by compare match A interrupts or compare match B interrupts.
Low power consumption function	Each unit can be placed in a module stop state

Table 22.2 TMR Functions

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 TMCi0	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCi1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCi1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCi2	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCi3	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCi3
Counter clear		TMR0.TCORA TMR0.TCORB TMRi0	TMR1.TCORA TMR1.TCORB TMRi1	TMR0.TCORA + TMR1.TCORA  TMR0.TCORB + TMR1.TCORB  TMRi0	TMR2.TCORA TMR2.TCORB TMRi2	TMR3.TCORA TMR3.TCORB TMRi3	TMR2.TCORA + TMR3.TCORA  TMR2.TCORB + TMR3.TCORB  TMRi2
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	—
ELC output event	Compare match A	○	—	○	○	—	○
	Compare match B	○	—	○	○	—	○
	TCNT overflow	○	—	○	○	—	○
ELC input event	Counting start	○	—	—	○	—	—
	Event counting	○	—	—	○	—	—
	Counting restart	○	—	—	○	—	—
Module stop setting*1		MSTPCRA.MSTPA5 bit (unit 0), MSTPCRA.MSTPA4 bit (unit 1)					

○: Possible

—: Impossible

Note 1. For details, see section 11, Low Power Consumption.

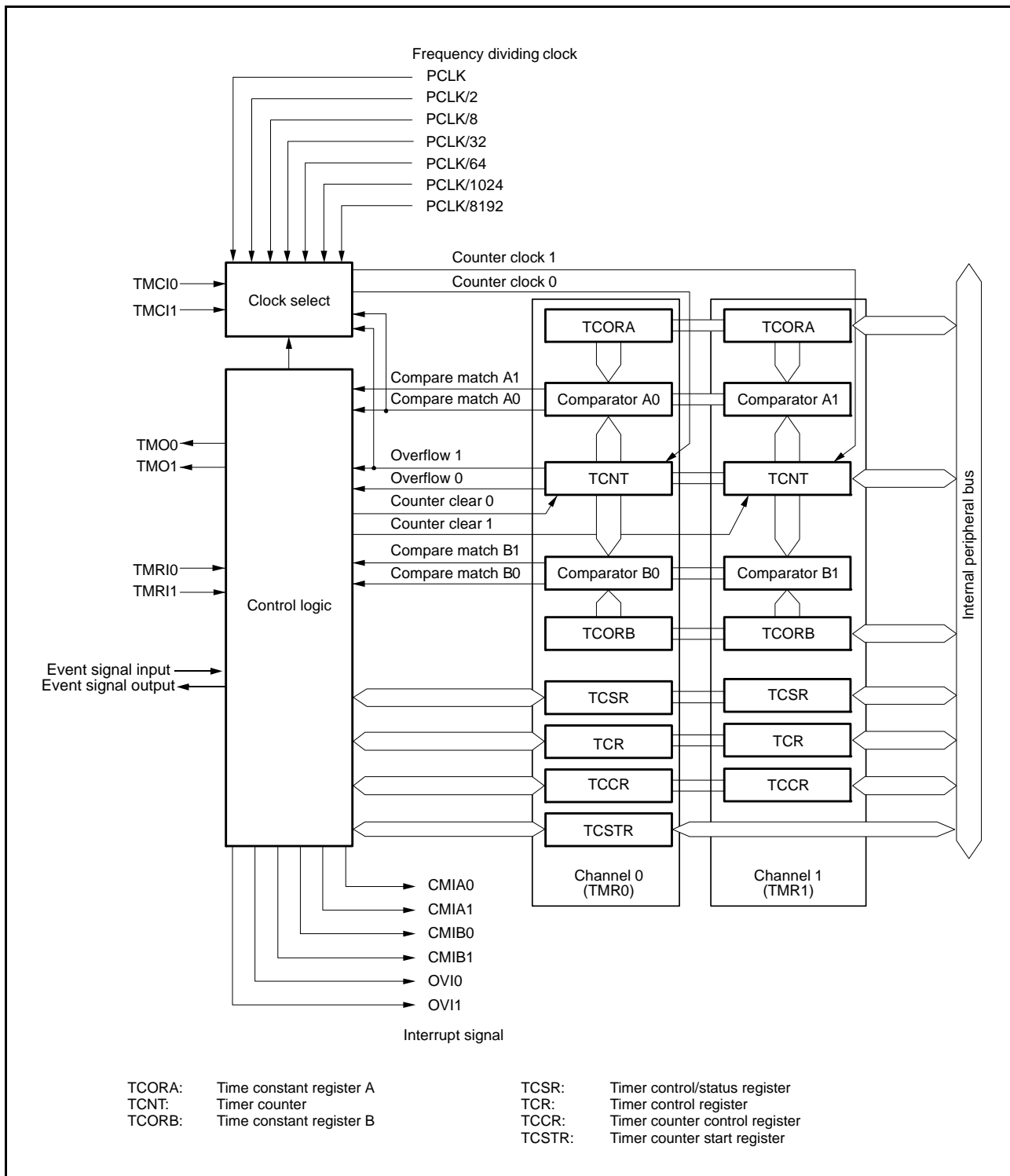


Figure 22.1 Block Diagram of TMR (Unit 0)

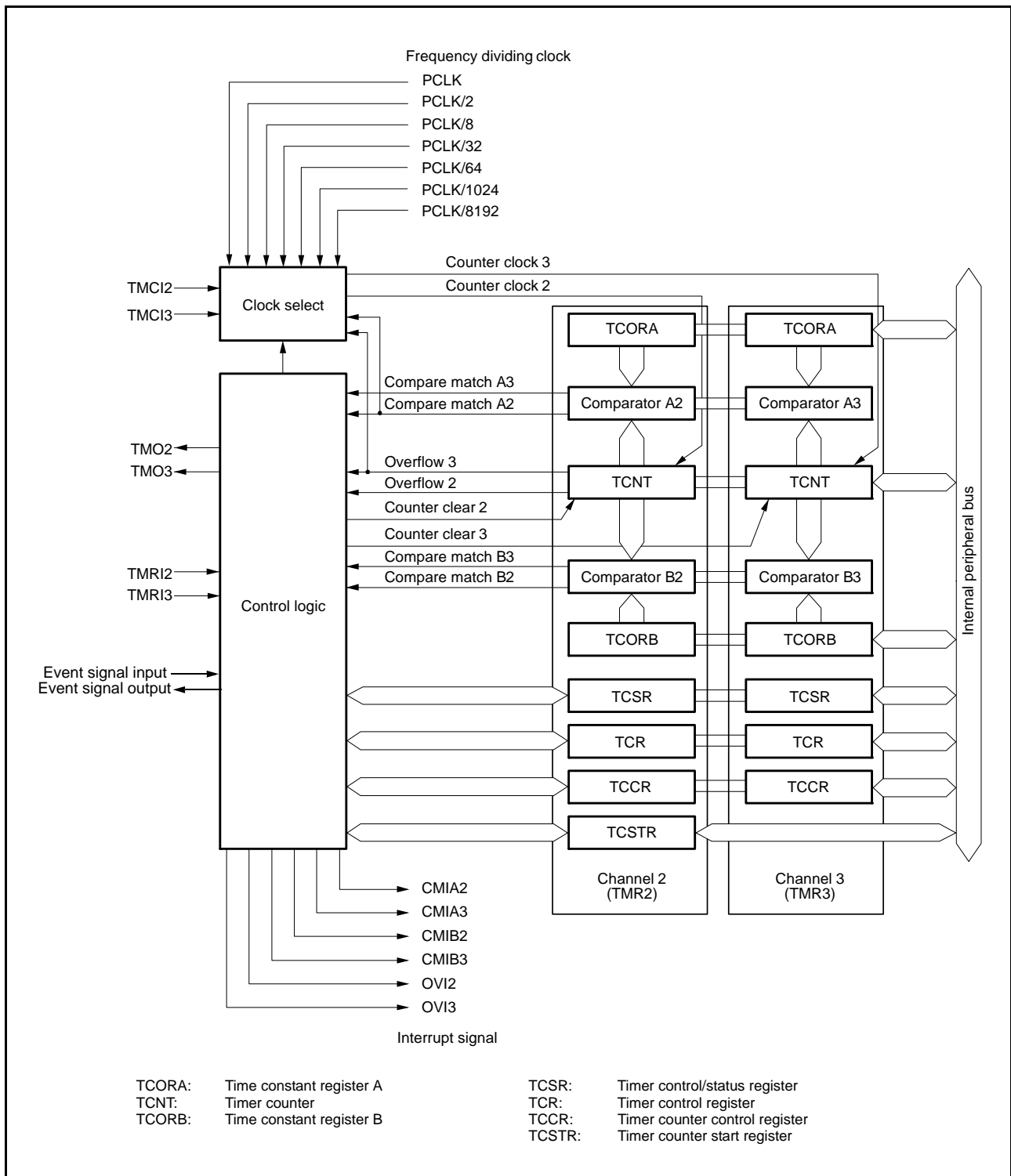


Figure 22.2 Block Diagram of TMR (Unit 1)

Table 22.3 lists the I/O pins of the TMR.

**Table 22.3 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

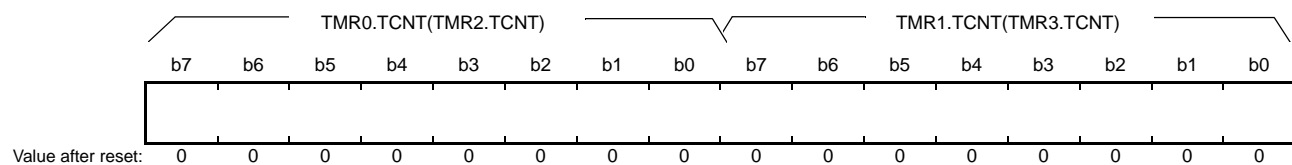
## 22.2 Register Descriptions

**Table 22.4 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

### 22.2.1 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.

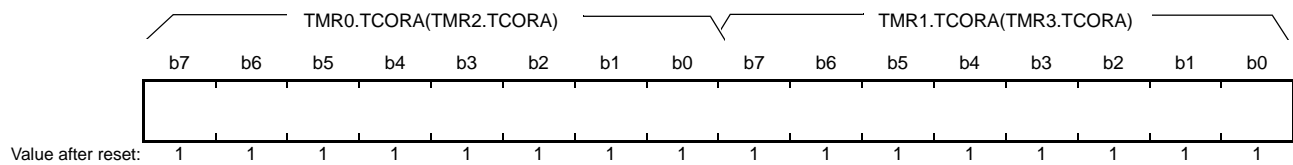
TCNT can be cleared by an external reset input signal, compare match A, or compare match B. Which compare match to be used for clearing is selected by the TCR.CCLR[1:0] bits.

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 14, Interrupt Controller (ICUb), and Table 22.6, TMR Interrupt Sources.

### 22.2.2 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.

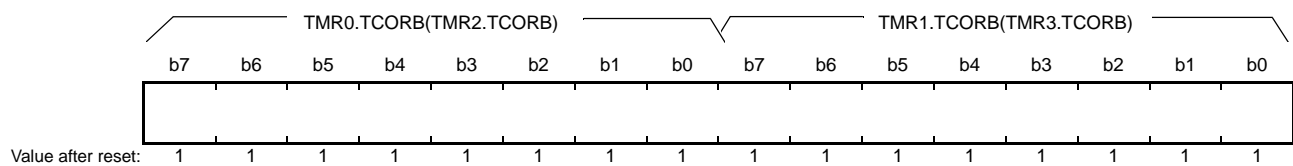
TMR0.TCORA and TMR1.TCORA (TMR2.TCORA and TMR3.TCORA) comprise a single 16-bit register so they can be accessed together by a word transfer instruction.

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 TMO pin can be freely controlled by this compare match A and the settings of the TCSR.OSA[1:0] bits.

### 22.2.3 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.

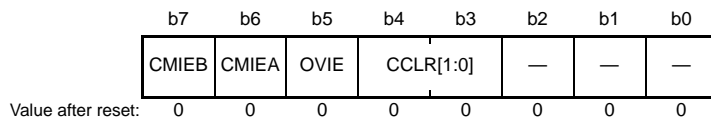
The value in TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare match B is generated, and a compare match B interrupt (low-level pulse) is output provided the interrupt request is enabled by the TCR.CMIEB bit.

However, comparison is not performed during writing to TCORBn. The timer output from the TMO pin can be freely controlled by this compare match B and the settings of the TCSRn.OSB[1:0] bits.



## 22.2.4 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* <sup>1</sup>	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 PORT<sub>n</sub>.PDR.B<sub>n</sub> bit for the corresponding pin to 0 and the PORT<sub>n</sub>.PMR.B<sub>n</sub> bit to 1. For details, see section 18, I/O Ports.

### CCLR[1:0] Bits (Counter Clear)

Select the condition by which TCNT is cleared.

### OVIE Bit (Timer Overflow Interrupt Enable)

Selects whether overflow interrupt requests (OVIn) issued by TCNT are enabled or disabled.

### CMIEA Bit (Compare Match Interrupt Enable A)

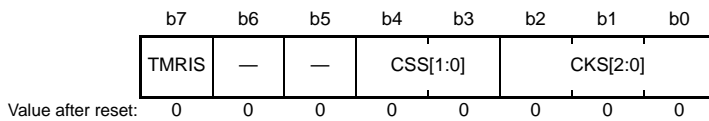
Selects whether compare match A interrupt requests (CMIA<sub>n</sub>) that are issued when the value of TCORA corresponds to that of TCNT are enabled or disabled.

### 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.

## 22.2.5 Timer Counter Control Register (TCCR)

Address(es): TMR0.TCCR 0008 820Ah, TMR1.TCCR 0008 820Bh, TMR2.TCCR 0008 821Ah, TMR3.TCCR 0008 821Bh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKS[2:0]	Clock Select*1	See Table 22.5.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 22.5.	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 18, 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 22.5.

### 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 22.5 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	0	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	0	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	0	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	0	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	0	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	0	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.		

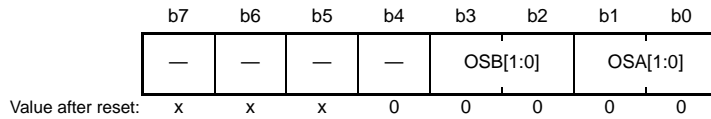
Note 1. To use an external reset, set the PORTn.PDR.Bn bit for the corresponding pin to 0 and the PORTn.PMR.Bn bit to 1. For details, see section 18, I/O Ports.

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.

## 22.2.6 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* <sup>1</sup>	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B* <sup>1</sup>	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When the OSA[1:0] and OSB[1:0] bits are all 0, the output enable signal corresponding to the TMO<sub>n</sub> pin is negated and a request for high-impedance output is issued to the I/O port. Timer output is low until the first compare match occurs after a reset when either of the OSA[1:0] or OSB[1:0] bits are 1.

### OSA[1:0] Bits (Output Select A)

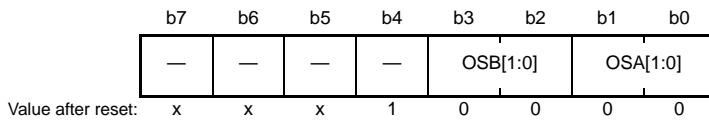
These bits select a method of TMO<sub>n</sub> pin output when compare match A of TCORA and TCNT occurs.

### OSB[1:0] Bits (Output Select B)

These bits select a method of TMO<sub>n</sub> 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*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (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. When the OSA[1:0] and OSB[1:0] bits are all 0, the output enable signal corresponding to the TMO<sub>n</sub> pin is negated and a request for high-impedance output is issued to the I/O port. Timer output is low until the first compare match occurs after a reset when either of the OSA[1:0] or OSB[1:0] bits are 1.

### OSA[1:0] Bits (Output Select A)

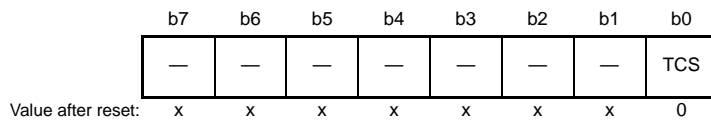
These bits select a method of TMO<sub>n</sub> pin output when compare match A of TCORA and TCNT occurs.

### OSB[1:0] Bits (Output Select B)

These bits select a method of TMO<sub>n</sub> pin output when compare match B of TCORB and TCNT occurs.

## 22.2.7 Time Count Start Register (TCSTR)

Address(es): TMR0.TCSTR 0008 820Ch, TMR2.TCSTR 0008 821Ch



Bit	Symbol	Bit Name	Description	R/W
b0	TCS	Timer Counter Status	0: Count stopped state in response to ELC. 1: Count start state in response to ELC.	R/W
b7 to b1	—	Reserved	These bits are read as an undefined value. The write value should be 0.	R/W

### TCS Bit (Timer Counter Status)

The TCS bit is used to check the state of the timer count in response to ELC.

When this bit is read as 1, it shows the timer start state in response to ELC. When this bit is read as 0, it shows the timer stopped state in response to ELC.

This bit is cleared by writing 0. Do not write 1 to this bit.

The TCS bit is valid only when the count start operation is selected by the ELOPD register of the event controller (ELC). For details, see section 22.7, Link Operation by ELC, or section 17, Event Link Controller (ELC).

## 22.3 Operation

### 22.3.1 Pulse Output

Figure 22.3 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.

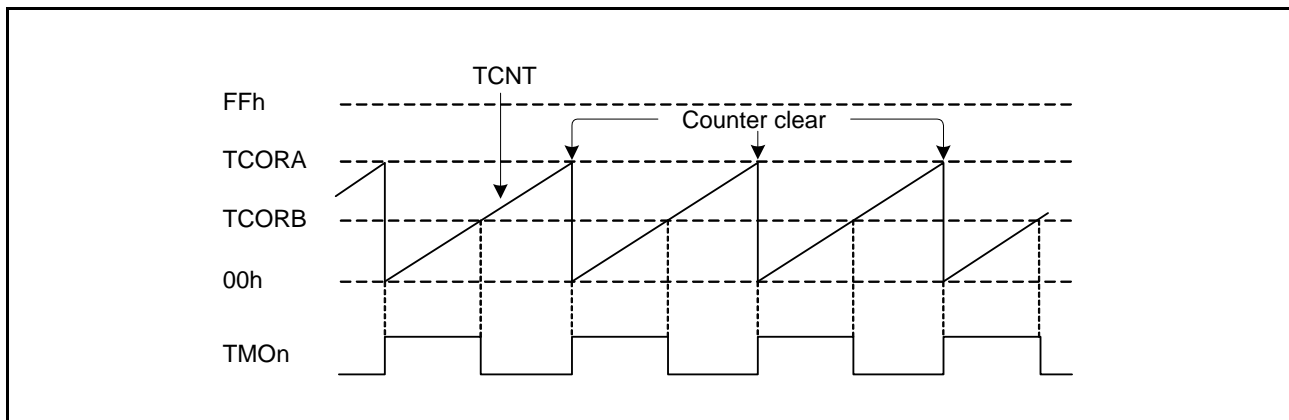


Figure 22.3 Example of Pulse Output (n = 0 to 3)

### 22.3.2 Reset Input

Figure 22.4 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.

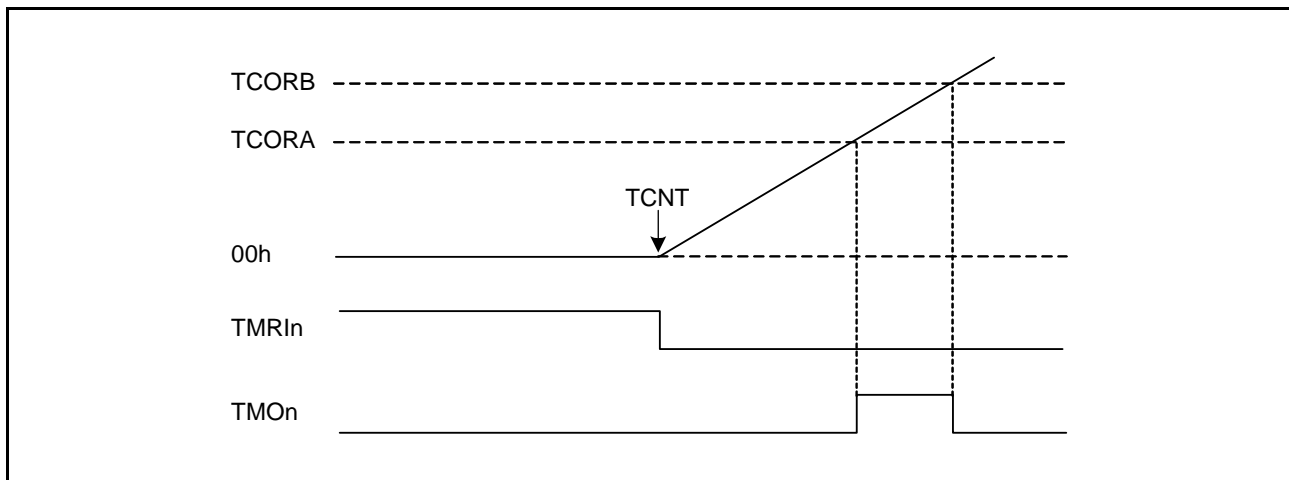


Figure 22.4 Example of Reset Input (n = 0 to 3)



## 22.4 Operation Timing

### 22.4.1 TCNT Count Timing

Figure 22.5 shows the count timing of TCNT for frequency dividing clock input. Figure 22.6 shows the count timing of TCNT for external clock input.

Note that the external clock pulse width must be at least 1.5 PCLK cycles for increment at a single edge, and at least 2.5 PCLK cycles for increment at both edges. The counter will not increment correctly if the pulse width is less than these values.

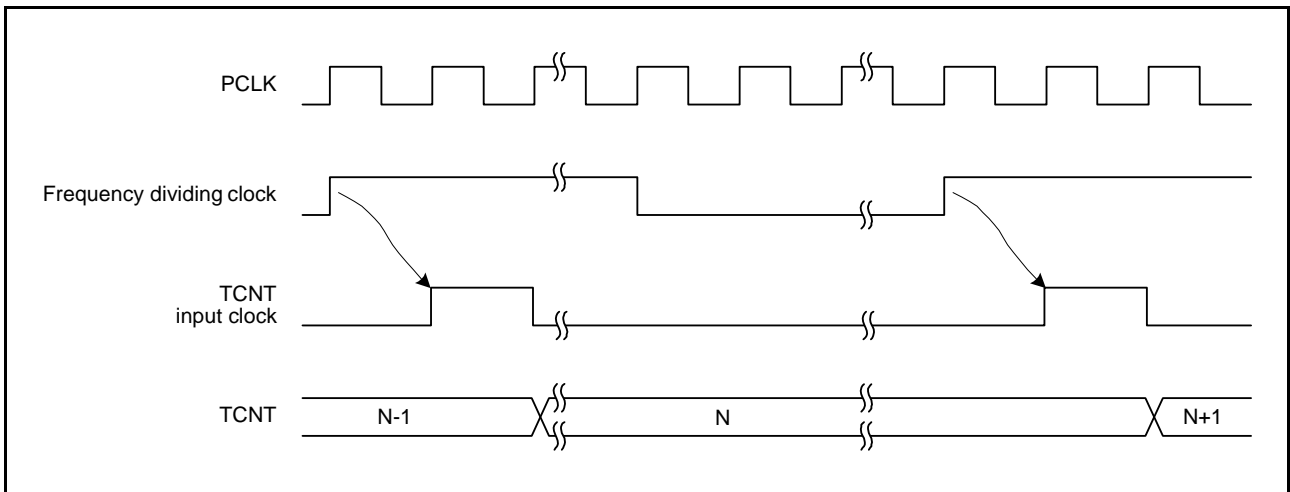


Figure 22.5 Count Timing for Frequency Dividing Clock Input

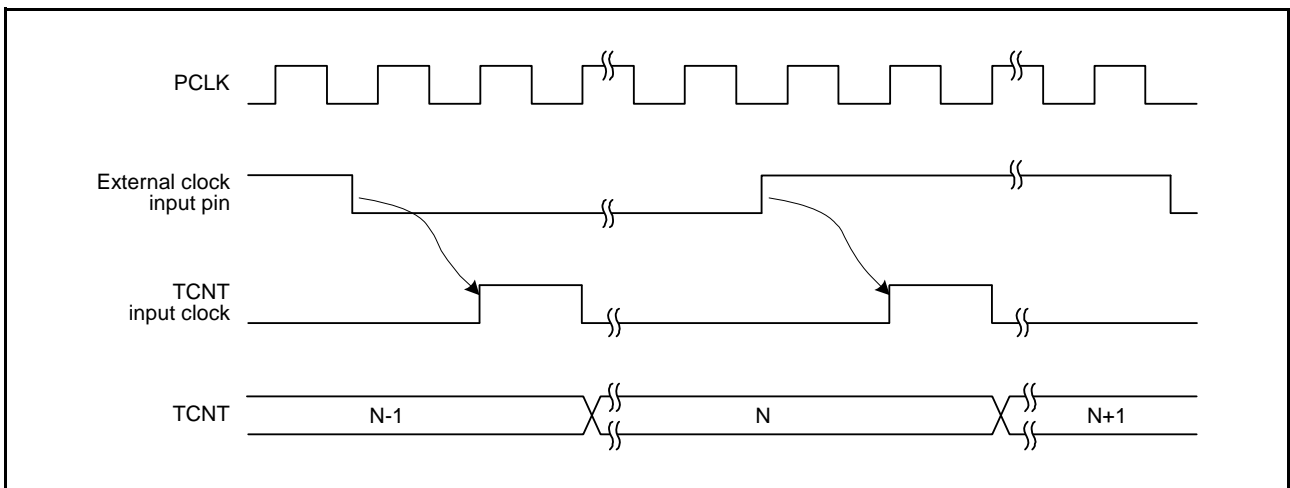


Figure 22.6 Count Timing for External Clock Input (at Both Edges)

### 22.4.2 Timing of Interrupt 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 22.7 shows the timing of output of the interrupt signal.

For the corresponding interrupt vector number, see section 14, Interrupt Controller (ICUb) and Table 22.6.

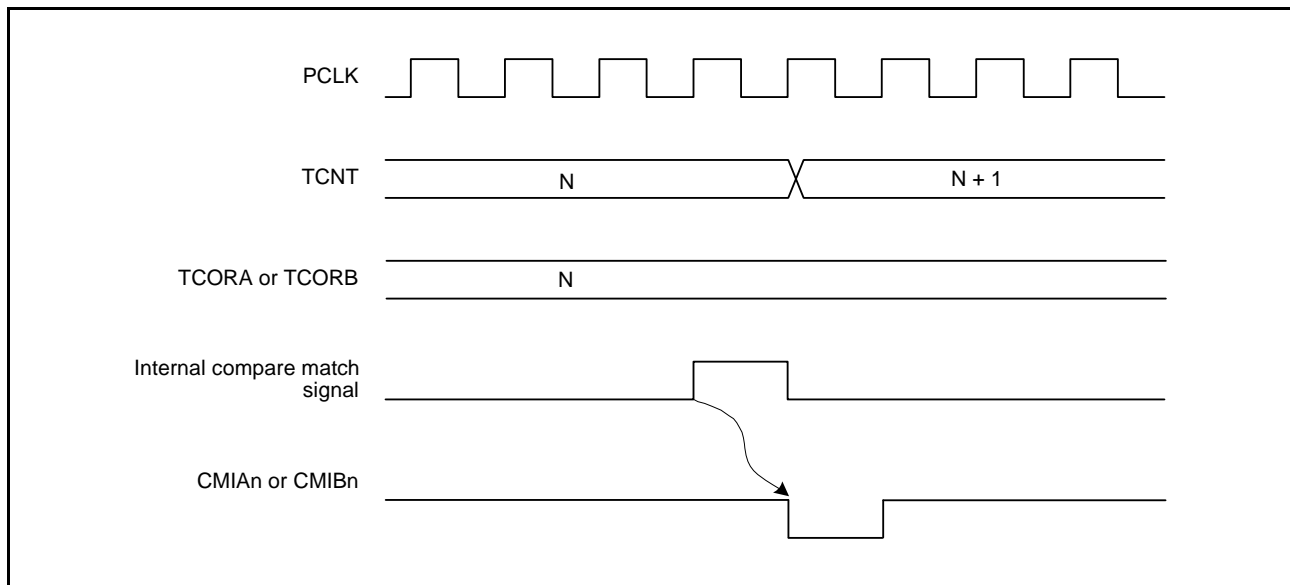


Figure 22.7 Timing of Interrupt Flag Setting to 1 at Compare Match ( $n = 0$  to 3)

### 22.4.3 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 22.8 shows the timing when the timer output is toggled by the compare match A signal.

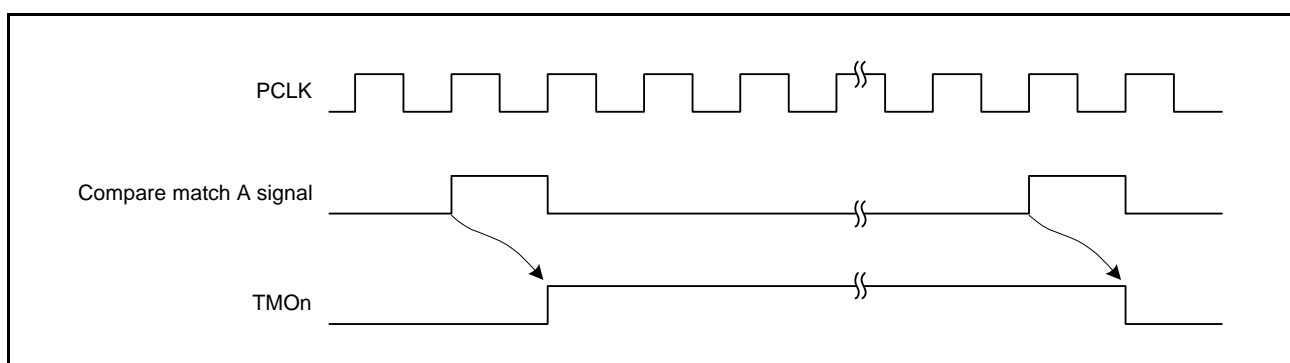


Figure 22.8 Timing of Timer Output at Compare Match A Signal ( $n = 0$  to 3)

### 22.4.4 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 22.9 shows the timing of this operation.

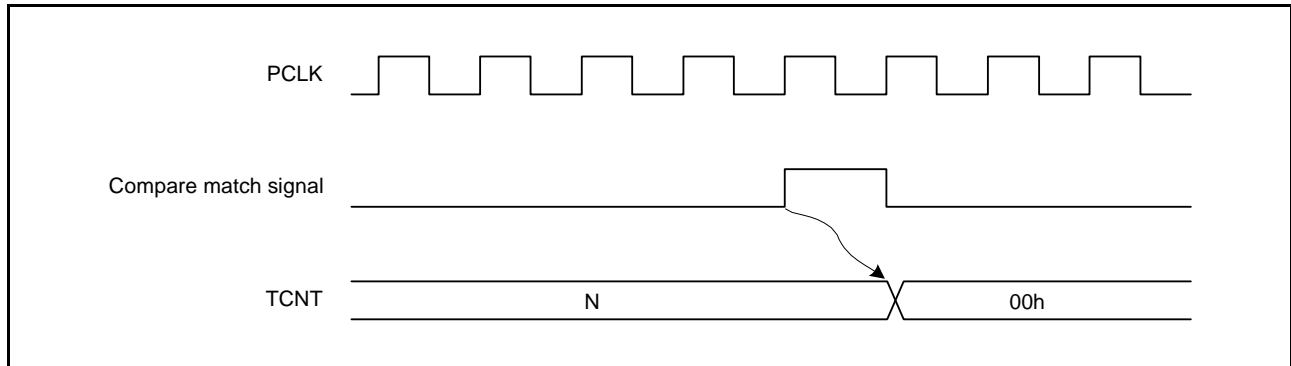


Figure 22.9 Timing of Counter Clear by Compare Match

### 22.4.5 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 2 PCLK cycles are required from an external reset input to clearing of TCNT. Figure 22.10 and Figure 22.11 show the timing of this operation.

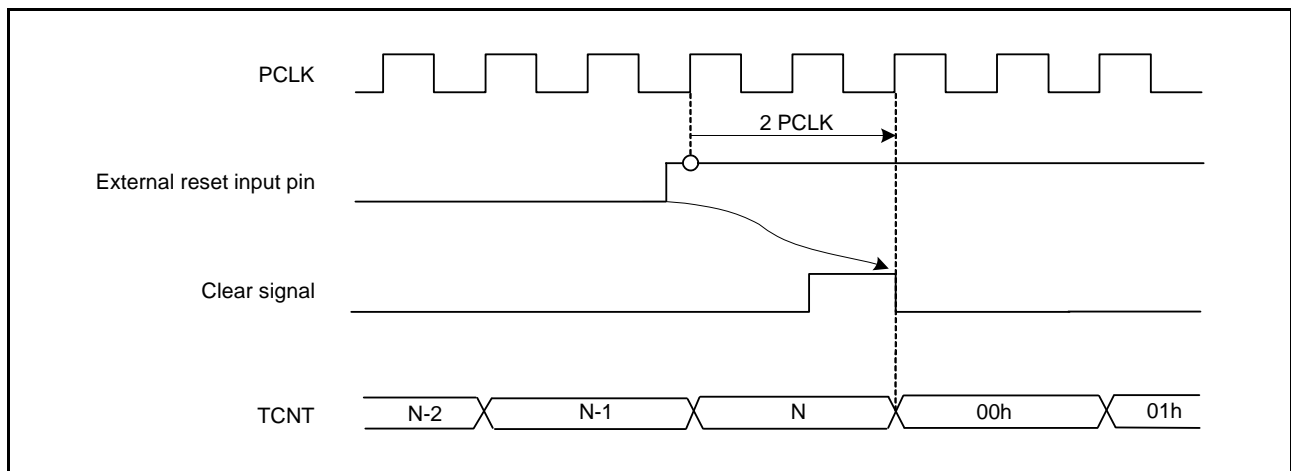


Figure 22.10 Timing of Clearance by External Reset (Rising Edge)

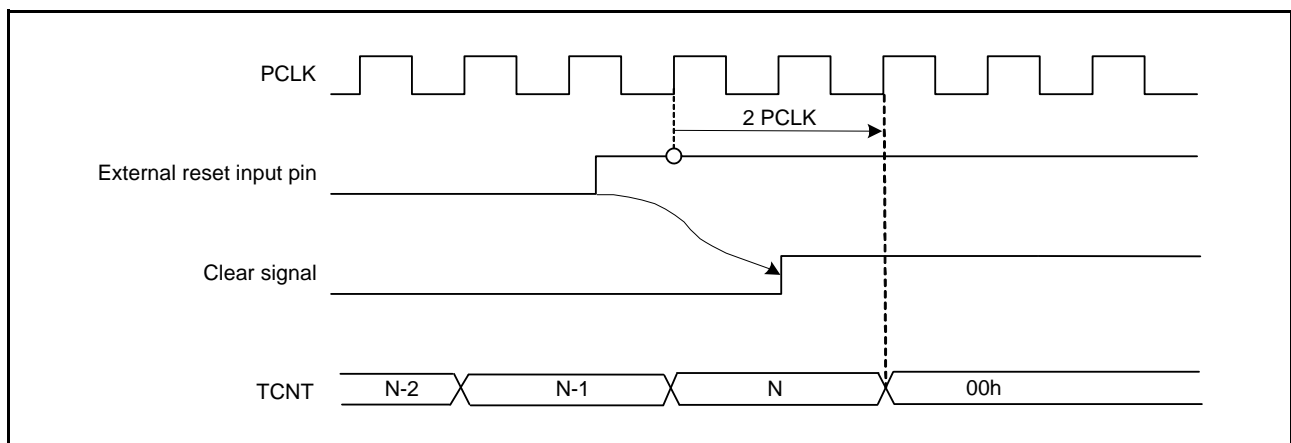


Figure 22.11 Timing of Clearance by External Reset (High Level)

### 22.4.6 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 22.12 shows the timing of output of the interrupt signal.

For the corresponding interrupt vector number, see section 14, Interrupt Controller (ICUb) and Table 22.6.

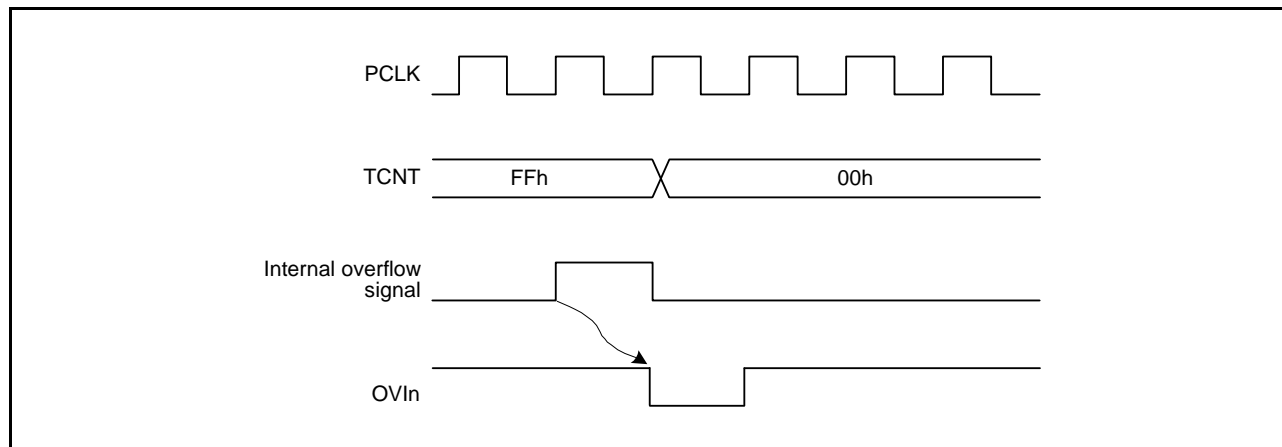


Figure 22.12 Timing of Overflow Interrupt Flag Setting to 1 (n = 0 to 3)

## 22.5 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.

### 22.5.1 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 TMRI0 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.

### 22.5.2 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.

## 22.6 Interrupt Sources

### 22.6.1 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 22.6.

It is also possible to activate the DTC by means of CMIA<sub>n</sub> and CMIB<sub>n</sub> interrupts.

**Table 22.6 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	

## 22.7 Link Operation by ELC

### 22.7.1 Event Signal Output to ELC

The TMR uses the event link controller (ELC) to perform link operation to the previously specified module using the interrupt request signal as the event signal. The TMR outputs compare match A, compare match B, and overflow signals as event signals. Channels that can be used in this way are TMR0 and TMR2.

The event signal can be output regardless of the setting of the corresponding interrupt request enable bits (TMR0.TCR.OVIE or TMR2.TCR.OVIE, TMR0.TCR.CMIEA or TMR2.TCR.CMIEA, and TMR0.TCR.CMIEB or TMR2.TCR.CMIEB). For details, see section 17, Event Link Controller (ELC).

The event output function can be used for the cascaded operation.

### 22.7.2 TMR Operation when Receiving an Event Signal from ELC

The TMR can perform either of the following operations upon the event previously specified by the ELSRn register of the ELC. However, the ELC does not support the cascaded operation.

#### (1) Count Start

When the TMR count start operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the TCSTR.TCS bit is set to 1, starting the TMR count operation. After the TMR count start operation is selected by the ELOPD register of the ELC, use the TCCR.CKS[2:0] and CSS[1:0] bits to select the count source.

If the specified event occurs while the TCS bit is 1, the event is ignored.

Write 0 to the TCSTR.TCS bit to stop counting.

When the count start event is input in the count stopped state, the TMR starts counting again according to the CKS[2:0] and CSS[1:0] bits.

The TCS bit is valid only when the ELOPD.TMR0MD and ELOPD.TMR2MD bits of the ELC select the count start operation.

#### (2) Event Count

When the TMR event count operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the events are counted as the count source regardless of the TCCR.CKS[2:0] and CSS[1:0] bit settings. Reading the counter value returns the number of events that have been actually input.

#### (3) Count Restart

When the TMR count restart operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the TCNT counter value is modified to the initial value. If the CKS[2:0] and CSS[1:0] bit settings are not disabling the clock input, the count operation is continued.

### 22.7.3 Notes on Operating TMR According to an Event Signal from ELC

The following describes the notes on operating the TMR using the event link feature.

#### (1) Count Start

When the event specified by *ELSRn* occurs during the write cycle to the *TCSTR.TCS* bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

#### (2) Event Count

When the event specified by *ELSRn* occurs during the write cycle to the *TCNT*, the cycle is not completed; event count operation according to the event occurrence takes priority.

#### (3) Count Restart

When the event specified by *ELSRn* occurs during the write cycle to the *TCNT*, the cycle is not completed; count value initialization according to the event occurrence takes priority.



## 22.8 Usage Notes

### 22.8.1 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.

### 22.8.2 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).

$$f = \text{PCLK} / (N + 1)$$

### 22.8.3 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 22.13.

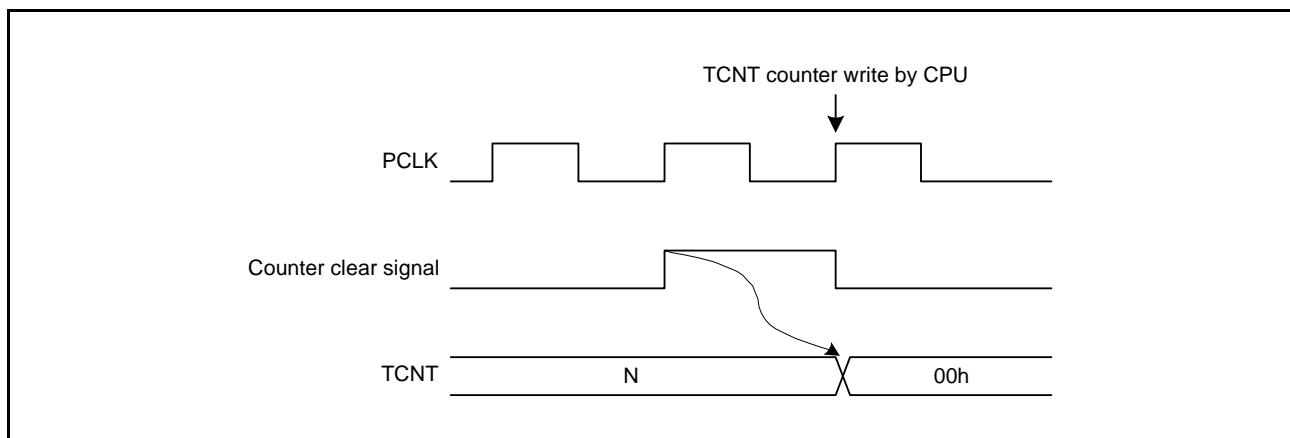


Figure 22.13 Conflict between TCNT Write and Counter Clear

### 22.8.4 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 22.14.

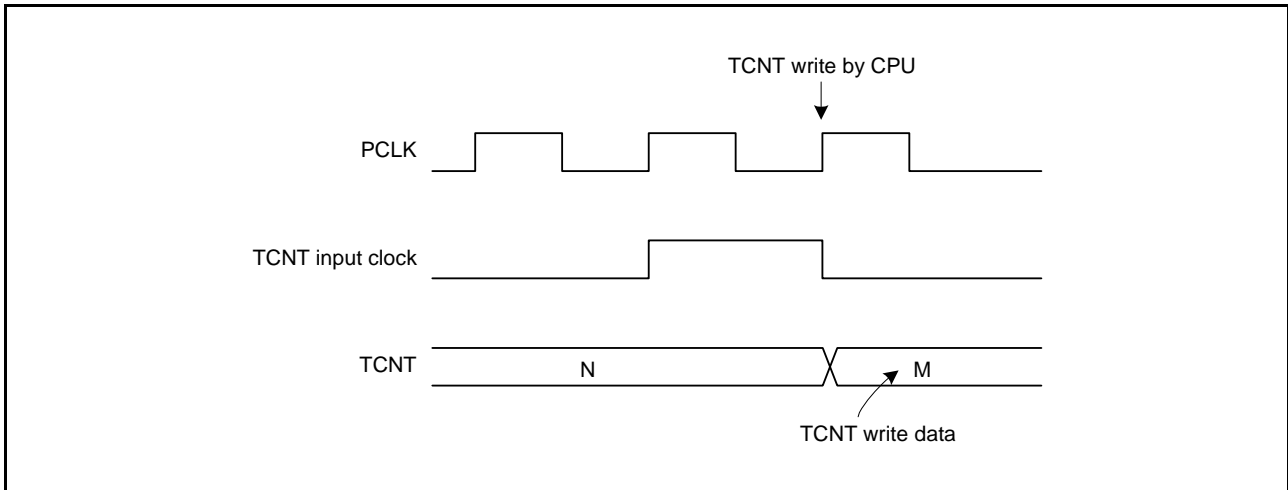


Figure 22.14 Conflict between TCNT Write and Increment

### 22.8.5 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 as shown in Figure 22.15, the write takes priority and the compare match signal does not reach High level.

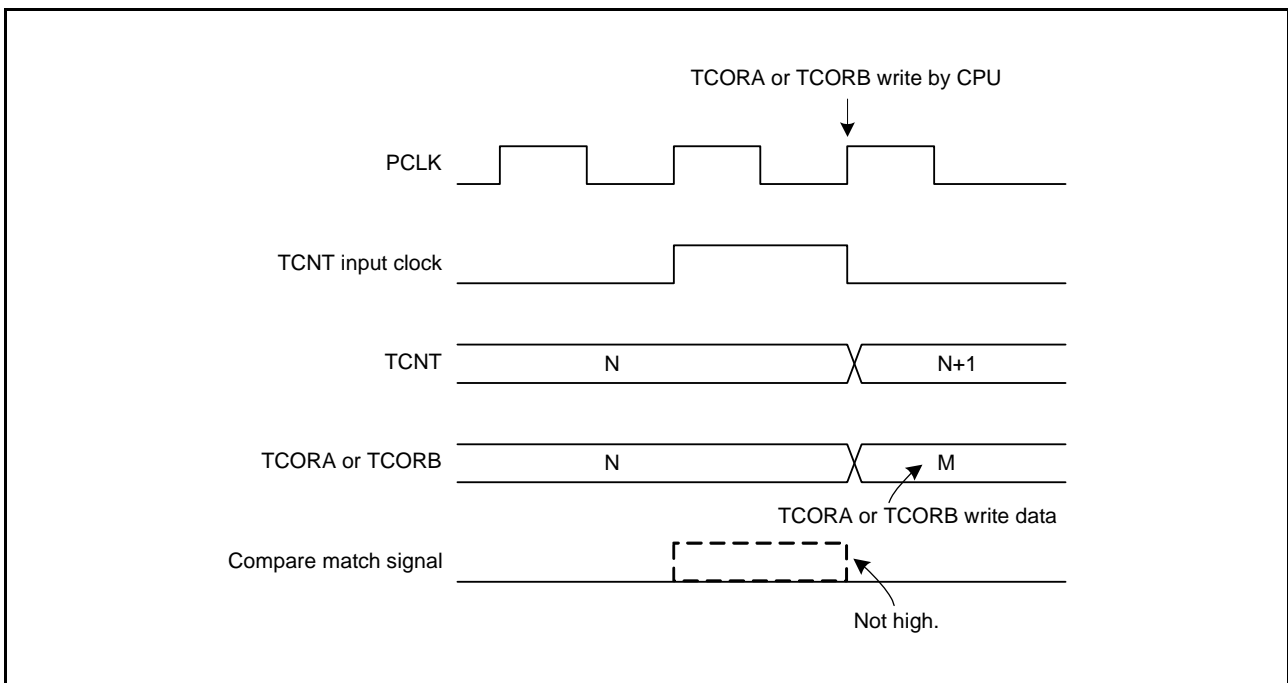


Figure 22.15 Conflict between TCORA or TCORB Write and Compare Match

### 22.8.6 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 22.7.

**Table 22.7 Timer Output Priorities**

Output Setting	Priority
Toggle output	High
High output	↑
Low output	
No change	Low

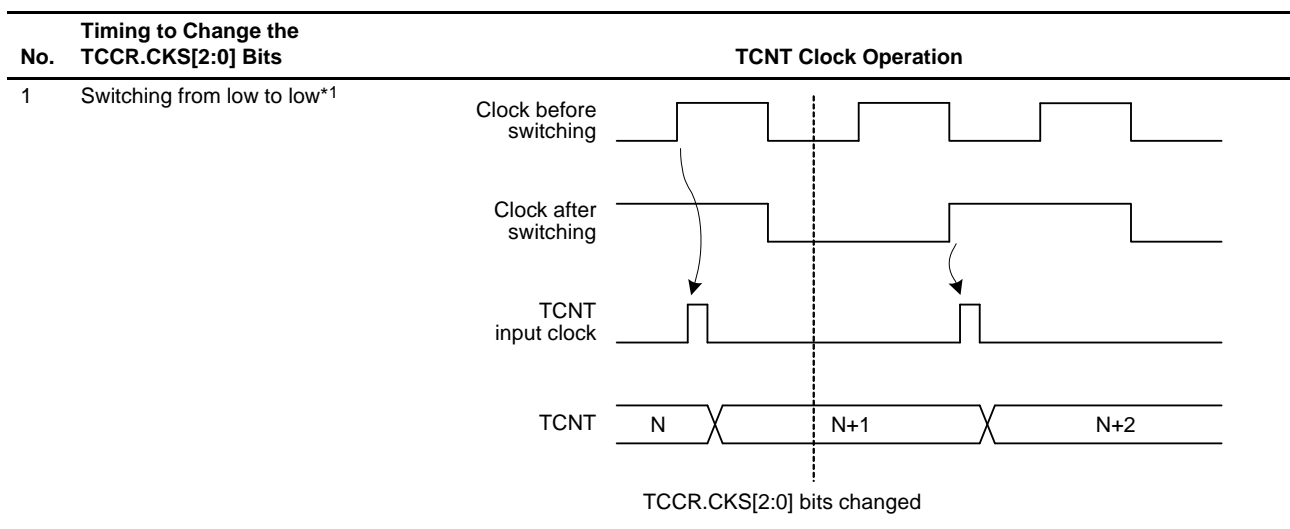
### 22.8.7 Switching of Frequency Dividing Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the frequency dividing clock is switched. Table 22.8 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.

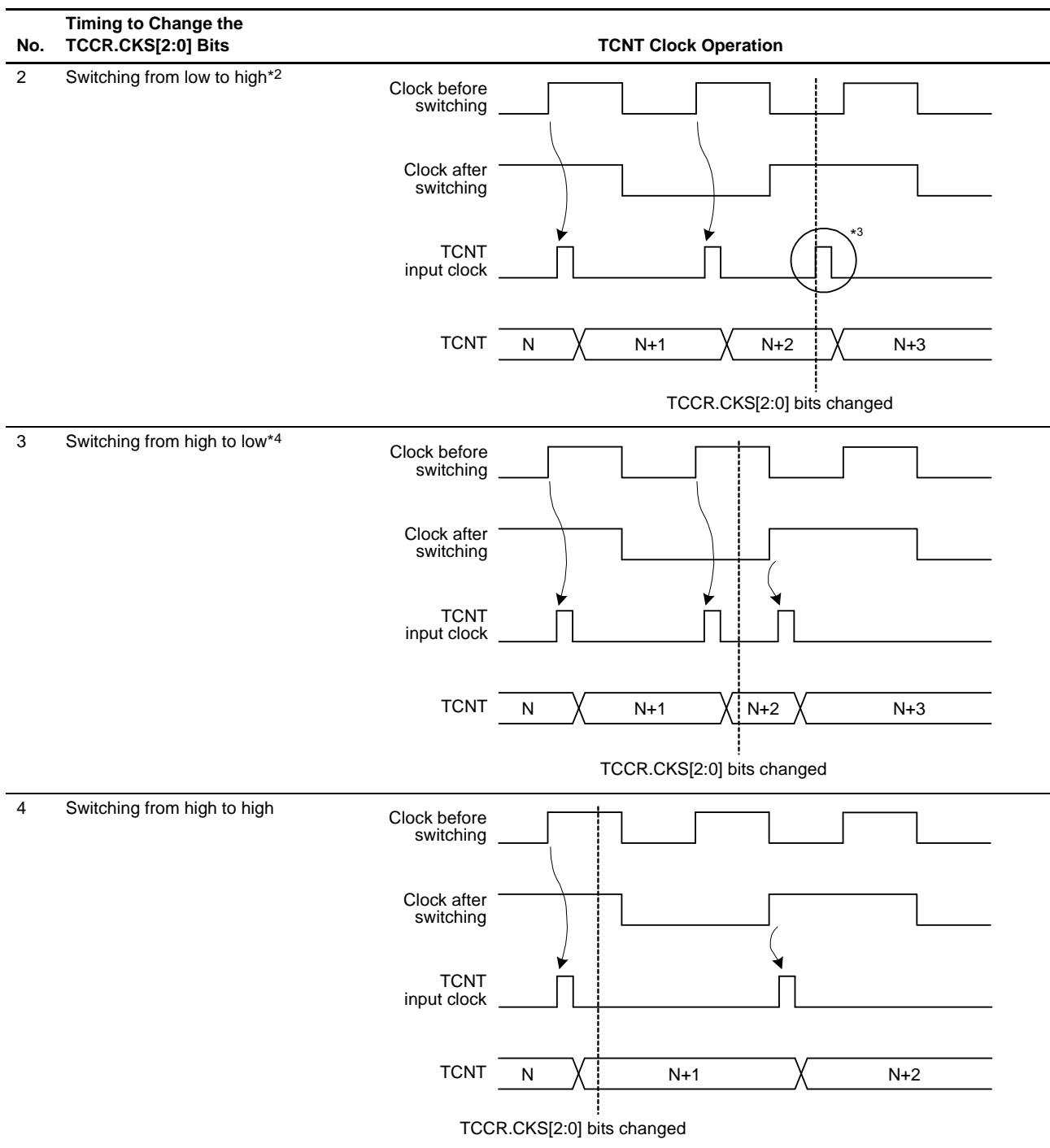
When TCNT clock is generated from an frequency dividing clock, the rising edge of the frequency dividing clock pulse are always monitored. If the signal levels of the clocks before and after switching change from low to high as shown in No. 2 in Table 22.8, the change is considered as an edge. Therefore, a TCNT clock pulse is generated and TCNT is incremented.

The erroneous increment of TCNT can also happen when switching between internal and frequency dividing clocks.

**Table 22.8 Switching of Frequency Dividing Clocks and TCNT Operation (1/2)**



**Table 22.8 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.

### 22.8.8 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.

### 22.8.9 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 22.16 shows operation timing when the compare match interrupt signal is continuously output.

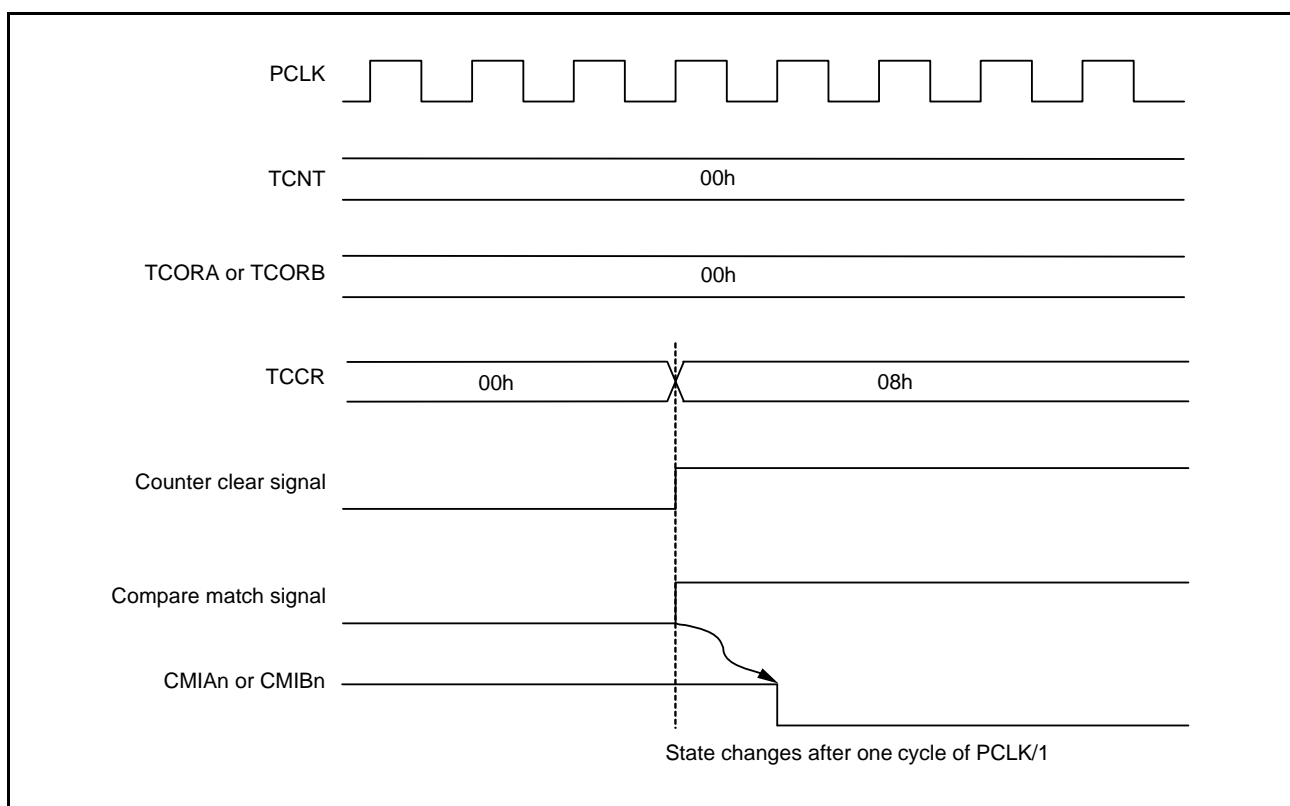


Figure 22.16 Continuous Output of Compare Match Interrupt Signal (n = 0 to 3)

## 23. Compare Match Timer (CMT)

This MCU 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.

In this section, “PCLK” is used to refer to PCLKB.

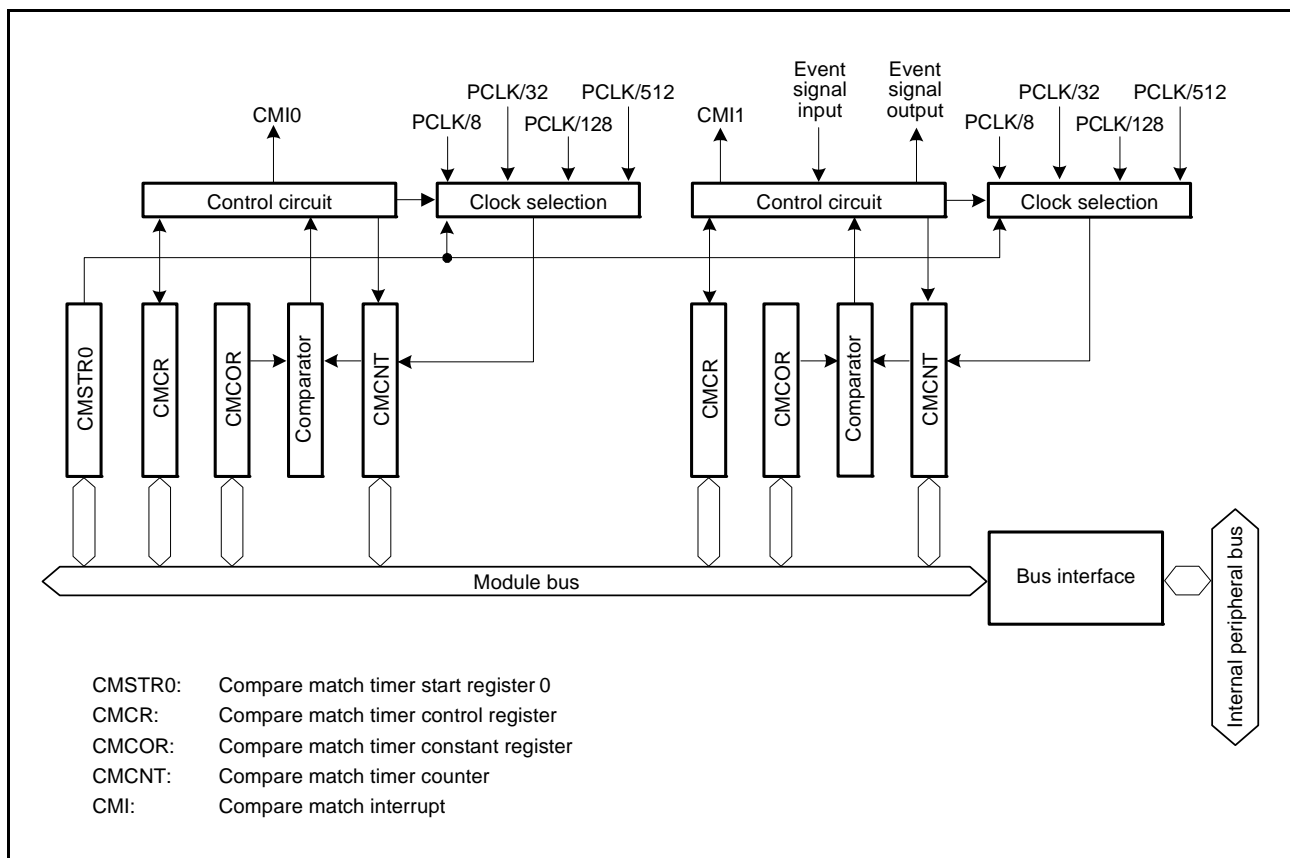
### 23.1 Overview

Table 23.1 lists the specifications for the CMT.

Figure 23.1 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. Compare match timer start register 0 (CMSTR0) and compare match interrupts (CMI0 and CMI1) of unit 0 correspond to compare match timer start register 1 (CMSTR1) and compare match interrupts (CMI2 and CMI3) of unit 1.

**Table 23.1 CMT Specifications**

Item	Description
Count clocks	<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 for each channel.</li> </ul>
Interrupt	A compare match interrupt can be requested for each channel.
Event link function (output)	An event signal is output upon a CMT1 compare match.
Event link function (input)	Linking to the specified module is possible. Count start, event count, or count restart is possible upon the specified event.
Low power consumption function	Each unit can be placed in a module stop state.

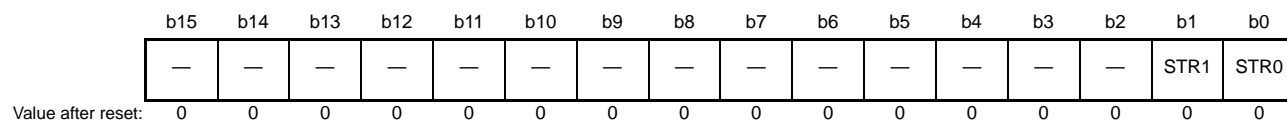


**Figure 23.1 CMT (Unit 0) Block Diagram**

## 23.2 Register Descriptions

### 23.2.1 Compare Match Timer Start Register 0 (CMSTR0)

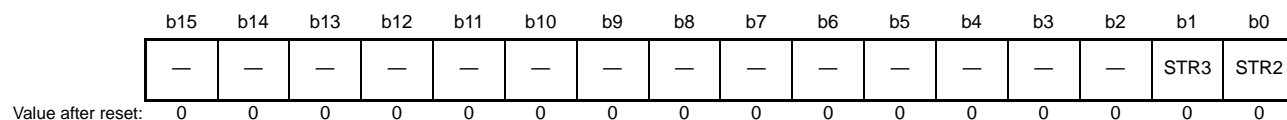
Address(es): 0008 8000h



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

### 23.2.2 Compare Match Timer Start Register 1 (CMSTR1)

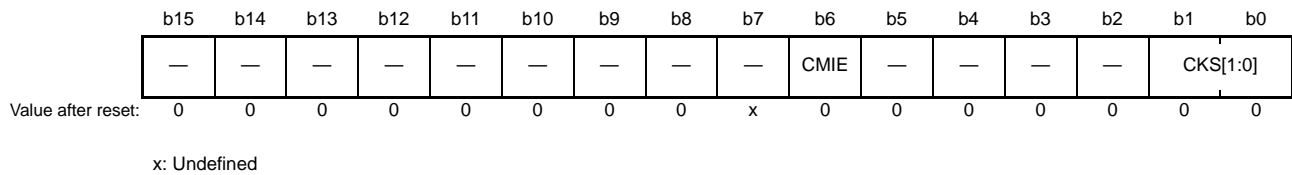
Address(es): 0008 8010h



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

### 23.2.3 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 source from four frequency dividing clocks obtained by dividing the peripheral module clock (PCLK).

When the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up on the clock selected with the CKS[1:0] bits.

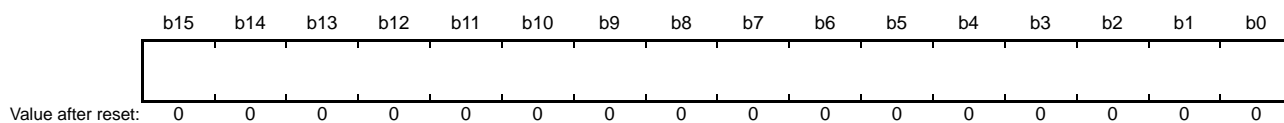
#### CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0 to 3) generation when the CMCNT counter and the CMCOR register values match.



### 23.2.4 Compare Match Counter (CMCNT)

Address(es): CMT0.CMCNT 0008 8004h, CMT1.CMCNT 0008 800Ah, CMT2.CMCNT 0008 8014h, CMT3.CMCNT 0008 801Ah



The CMCNT counter is a readable/writable up-counter.

When an frequency dividing clock is selected by the CMCOR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.

When the value in the CMCNT counter and the value in the CMCOR register match, the CMCNT counter is set to 0000h. At the same time, a compare match interrupt (CMI<sub>n</sub>) (n = 0 to 3) is generated.

### 23.2.5 Compare Match Constant Register (CMCOR)

Address(es): CMT0.CMCOR 0008 8006h, CMT1.CMCOR 0008 800Ch, CMT2.CMCOR 0008 8016h, CMT3.CMCOR 0008 801Ch



The CMCOR register is a readable/writable register to set a value for compare match with the CMCNT counter.

### 23.3 Operation

#### 23.3.1 Periodic Count Operation

When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.

When the value in the counter and the value in the register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. The CMCNT counter then starts counting up again from 0000h. Figure 23.2 shows the operation of the CMCNT counter.

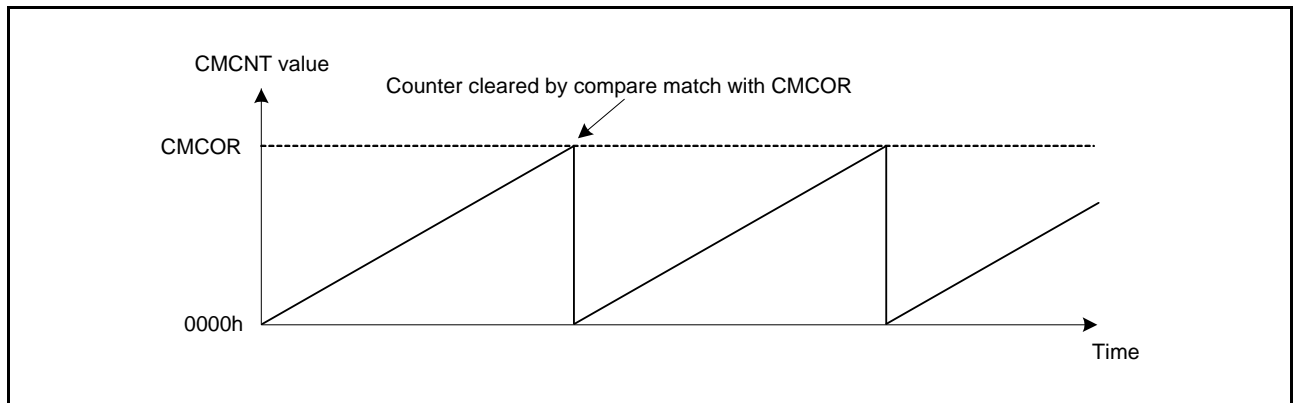


Figure 23.2 CMCNT Counter Operation

#### 23.3.2 CMCNT Count Timing

As the count clock to be input to the CMCNT counter, one of four frequency dividing clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral clock (PCLK) can be selected with the CMCR.CKS[1:0] bits.

Figure 23.3 shows the timing of the CMCNT counter.

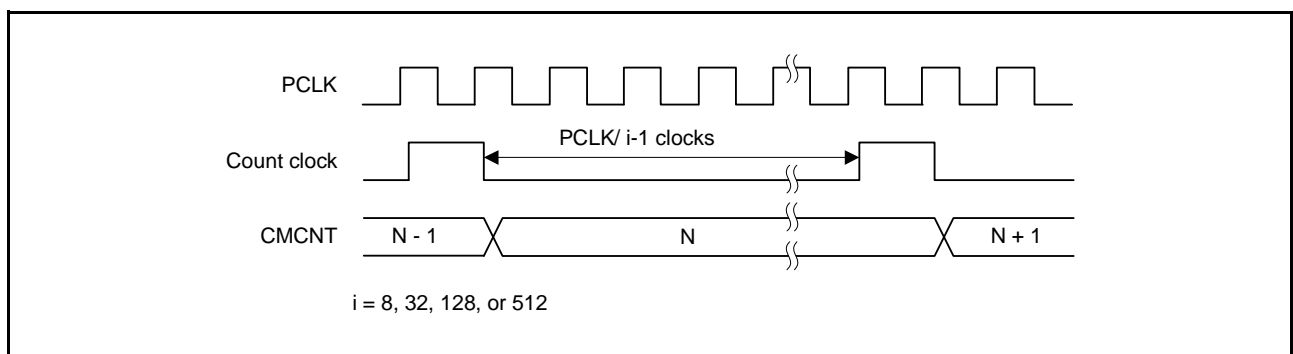


Figure 23.3 CMCNT Count Timing

## 23.4 Interrupts

### 23.4.1 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 request is used to generate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 14, Interrupt Controller (ICUb).

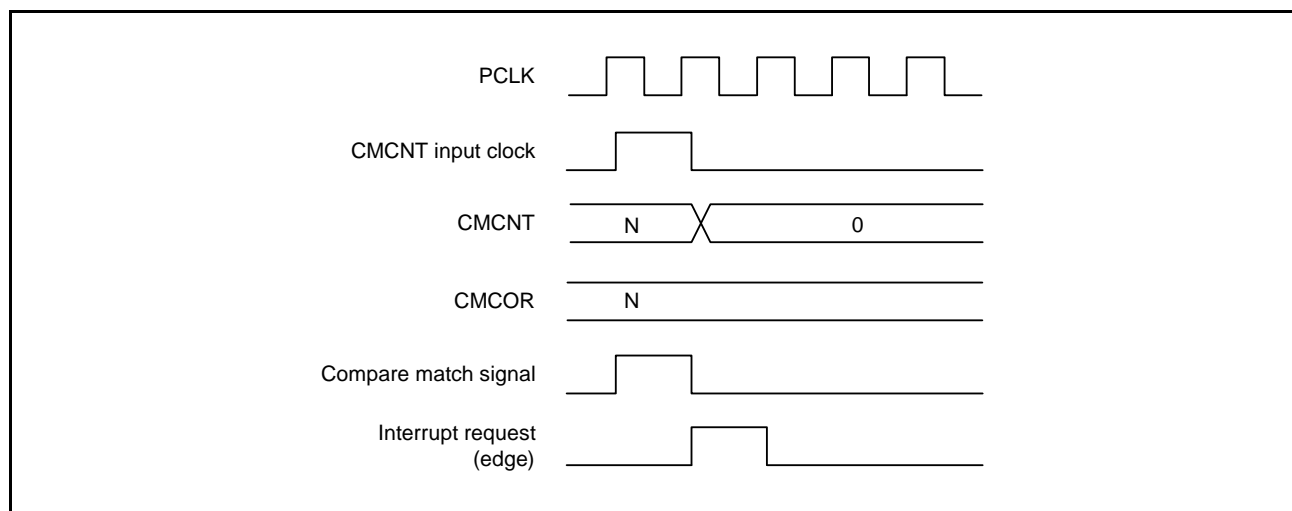
**Table 23.2 CMT Interrupt Sources**

Name	Interrupt Sources	DTC Activation
CMI0	Compare match in CMT0	Possible
CMI1	Compare match in CMT1	Possible

### 23.4.2 Timing of Compare Match Interrupt Generation

When the CMCNT counter and the CMCOR register 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 the CMCOR register and the CMCNT counter, the compare match signal is not generated until the next the CMCNT counter input clock.

Figure 23.4 shows the timing of a compare match interrupt.



**Figure 23.4 Timing of a Compare Match Interrupt**

## 23.5 Link Operations by ELC

### 23.5.1 Event Signal Output to ELC

The CMT uses the event link controller (ELC) to perform link operation to a preset module using the interrupt request signal as the event signal. The CMT outputs the event signal upon a CMT1 compare match.

The event signal can be output regardless of the setting of the corresponding interrupt request enable bit (CMTn.CMCR.CMIE).

### 23.5.2 CMT Operation When Receiving an Event Signal from ELC

The CMT can perform either of the following operations upon the event preset by the ELSR7 register of the ELC.

#### (1) Count Start

When the CMT count start operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs, the CMSTR0.STR1 bit is set to 1, starting the CMT count operation.

However, if the specified event occurs while the CMSTR0.STR1 bit is 1, the event is ignored.

#### (2) Event Count

When the CMT event count operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs with the CMSTR0.STR1 bit being 1, the events are counted as the count source regardless of the CMT1.CMCR.CKS[1:0] bit setting. Reading the counter value returns the number of events that have been actually input.

#### (3) Count Restart

When the CMT count restart operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs, the CMT1.CMCNT counter value is modified to the initial value. If the CMSTR0.STR1 bit is 1 here, the count operation can be continued.

### 23.5.3 Notes on Operating CMT According to an Event Signal from ELC

The following describes the notes on operating the CMT using the event link feature.

#### (1) Count Start

When the event specified by ELSR7 occurs during the write cycle to the CMSTR0.STR1 bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

#### (2) Event Count

When the event specified by ELSR7 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; event count operation according to the event occurrence takes priority.

#### (3) Count Restart

When the event specified by ELSR7 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; count value initialization according to the event occurrence takes priority.

## 23.6 Usage Notes

### 23.6.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the module stop control register. After a reset, the CMT is in the module stop state. The registers can be accessed by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

### 23.6.2 Conflict between CMCNT Counter Writing and Compare Match

When the compare match signal is generated while writing to the CMCNT counter, clearing the CMCNT counter has priority over writing to it. In this case, the CMCNT counter is not written to. Figure 23.5 shows the timing to clear the CMCNT counter.

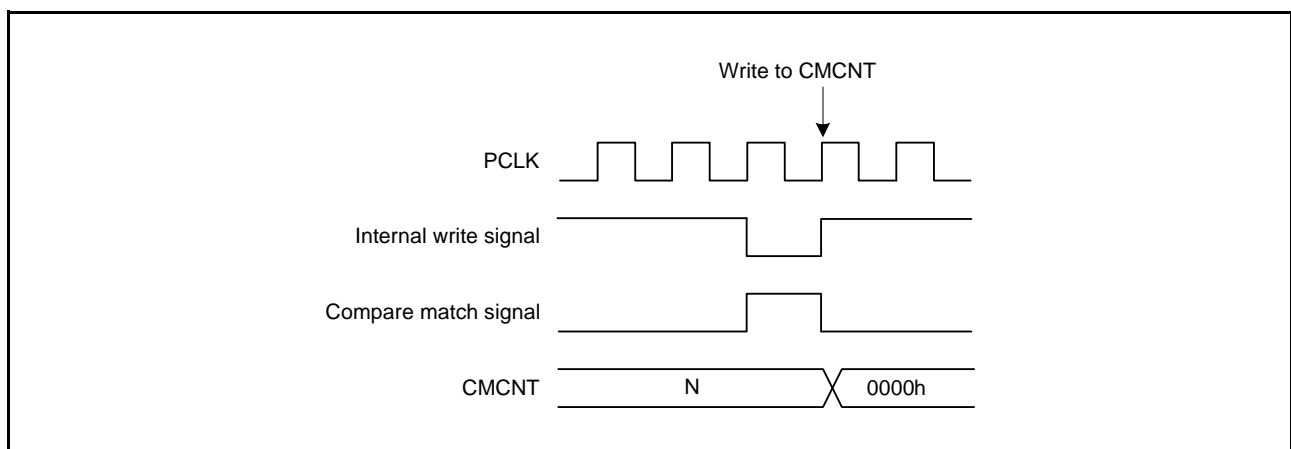


Figure 23.5 Conflict between CMCNT Counter Writing and Compare Match

### 23.6.3 Conflict between CMCNT Counter Writing and Incrementing

If writing to the counter and the incrementing conflict, the writing has priority over the incrementing. Figure 23.6 shows the timing to write the CMCNT counter.

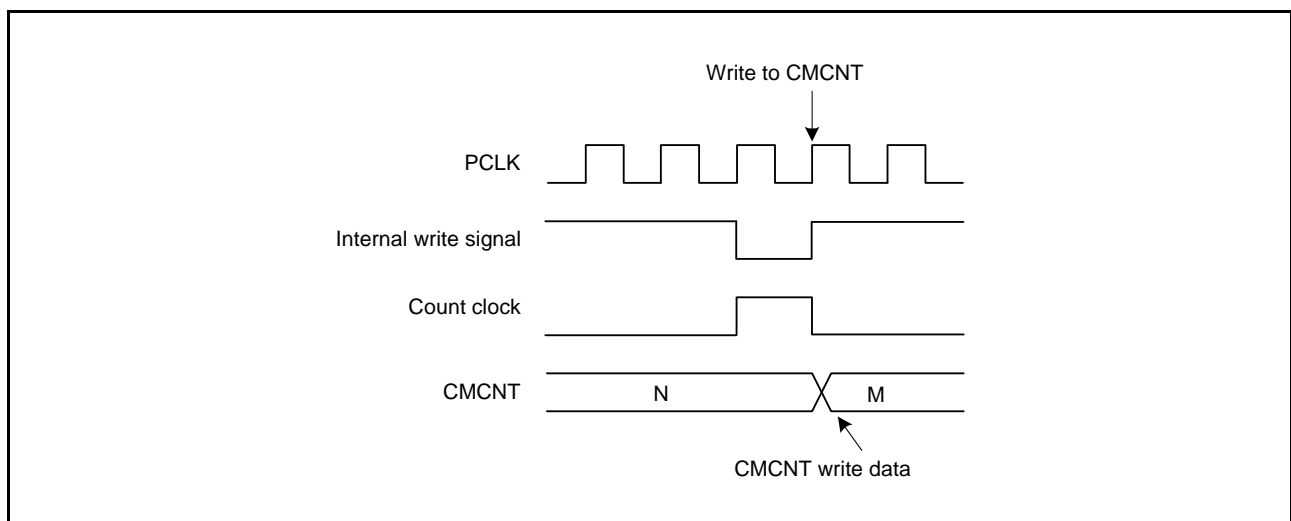


Figure 23.6 Conflict between CMCNT Counter Writing and Incrementing

## 24. Realtime Clock (RTCA)

In this section, “PCLK” is used to refer to PCLKB.

### 24.1 Overview

The RTC has two types of counting modes: calendar count mode and binary count mode. They are used by switching the register settings.

For calendar count mode, the RTC has a 100 year calendar from 2000 to 2099 and automatically adjusts dates for leap years.

For binary count mode, the RTC does not count in terms of years, months, dates, day-of-week, hours, or minutes; it counts seconds, and retains the information as a serial value. This mode can be used for calendars other than the Gregorian calendar.

The RTC uses the 128-Hz clock which is acquired by the count source divided by the prescaler as the basic clock. Year, month, date, day-of-week, a.m./p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted in 1/128 second units.

Table 24.1 lists the specifications of the RTC, Figure 24.1 shows a block diagram of the RTC, and Table 24.2 shows the pin configuration of the RTC.

**Table 24.1 RTC Specifications**

Item	Description
Count mode	Calendar count mode/binary count mode
Count source*1	Sub-clock (XCIN)
Clock and calendar functions	<ul style="list-style-type: none"> <li>Calendar count mode Year, month, date, day-of-week, hour, minute, second are counted, BCD display 12 hours/24 hours mode switching function 30 seconds adjustment function (a number less than 30 is rounded down to 00 seconds, and 30 seconds or more are rounded up to one minute) Automatic adjustment function for leap years</li> <li>Binary count mode Count seconds in 32 bits, binary display</li> <li>Common to both modes Start/stop function The sub-second digit is displayed in binary units (1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, or 64 Hz). Clock error correction function Clock (1-Hz/64-Hz) output</li> </ul>
Interrupts	<ul style="list-style-type: none"> <li>Alarm interrupt (ALM) As an alarm interrupt condition, selectable which of the below is compared with: Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected Binary count mode: Each bit of the 32-bit binary counter</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) An interrupt is generated at either of the following timings: - When a carry from the 64-Hz counter to the second counter is generated. - When the 64-Hz counter is changed and the R64CNT register is read at the same time.</li> <li>Recovery from software standby mode can be performed by an alarm interrupt or periodic interrupt</li> </ul>

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB)  $\geq$  the frequency of the count source clock.

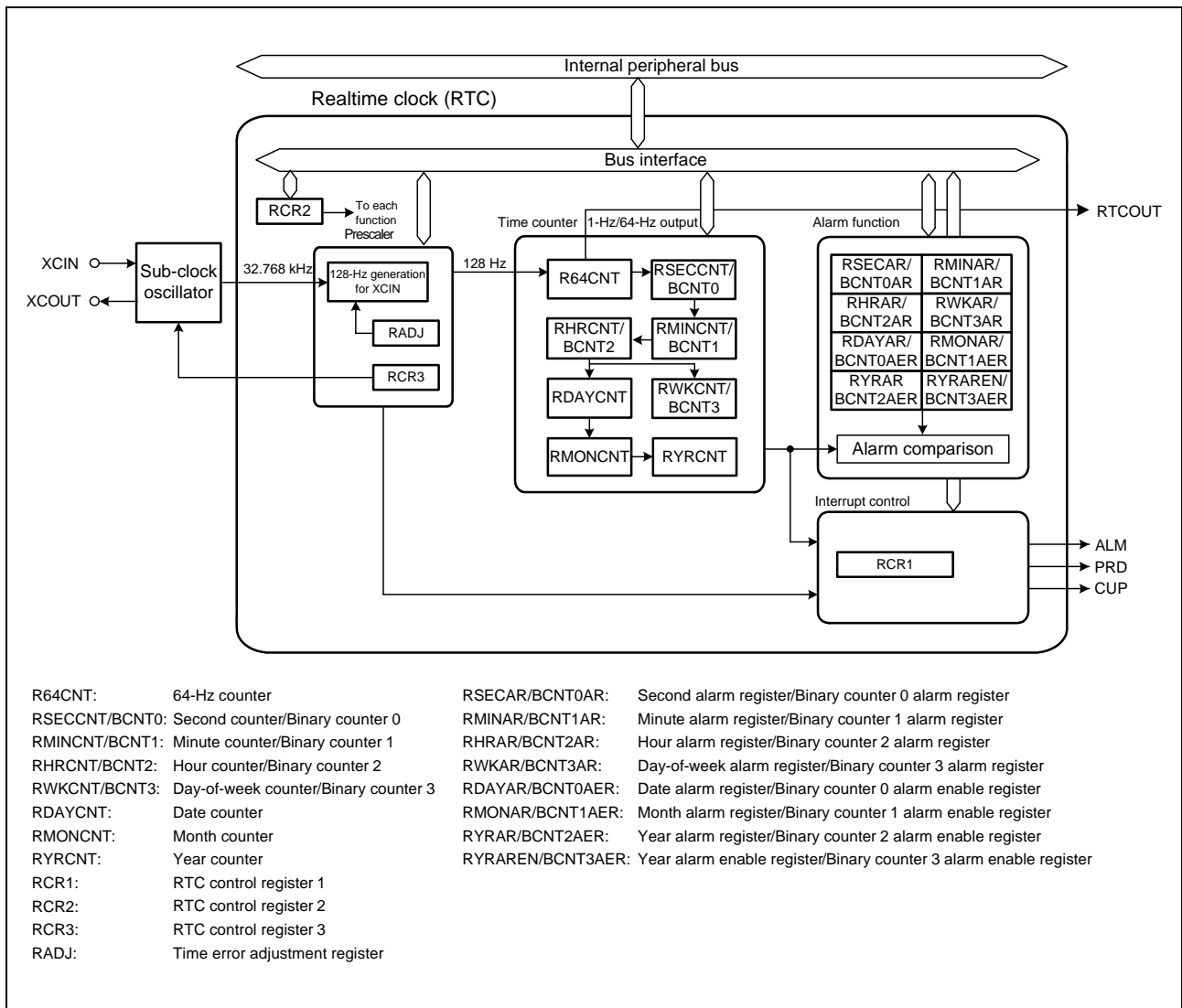


Figure 24.1 Block Diagram of RTC

Table 24.2 Pin Configuration of RTC

Pin Name	I/O	Function
XCIN	Input	Connect a 32.768-kHz crystal to these pins.
XCOUT	Output	
RTCOUT	Output	This pin is used to output a 1-Hz/64-Hz waveform.

## 24.2 Register Descriptions

When writing to or reading from RTC registers, do so in accordance with section 24.5.5, Notes 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. 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, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate. 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 immediately after setting any of these registers. For details, refer to section 24.5.4, Transitions to Low Power Consumption Modes after Setting Registers.

### 24.2.1 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	64 Hz	Indicate the state between 1 Hz and 64 Hz of the sub-second digit.	R
b1	F32HZ	32 Hz		R
b2	F16HZ	16 Hz		R
b3	F8HZ	8 Hz		R
b4	F4HZ	4 Hz		R
b5	F2HZ	2 Hz		R
b6	F1HZ	1 Hz		R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

The R64CNT counter is used in both calendar count mode and in binary count mode.

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-second range can be confirmed by reading this counter.

This counter is set to 00h by an RTC software reset or executing 30-second adjustment.

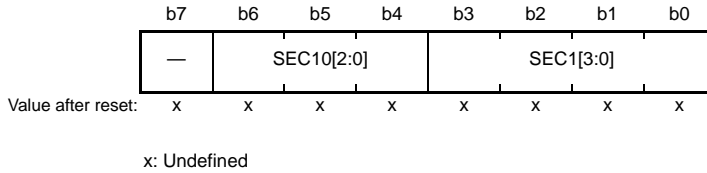
To read this counter, follow the procedure in section 24.3.5, Reading 64-Hz Counter and Time.



### 24.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0)

(1) In calendar count mode:

Address(es): 0008 C402h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Count	Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	SEC10[2:0]	10-Second Count	Counts from 0 to 5 for 60-second counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

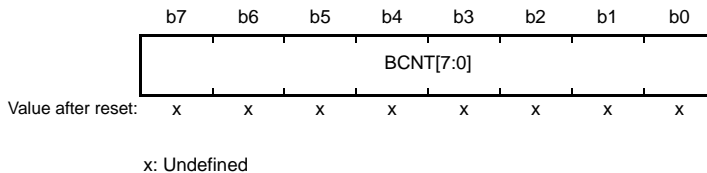
The RSECCNT counter is used for setting and counting the BCD-coded second value. It counts carries generated once per second in the 64-Hz counter.

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 24.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): 0008 C402h



The BCNT0 counter is a readable/writable 32-bit binary counter b7 to b0.

The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.

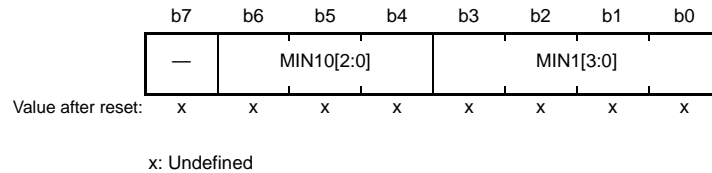
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 24.3.5, Reading 64-Hz Counter and Time.

### 24.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

(1) In calendar count mode:

Address(es): 0008 C404h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Count	Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	MIN10[2:0]	10-Minute Count	Counts from 0 to 5 for 60-minute counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

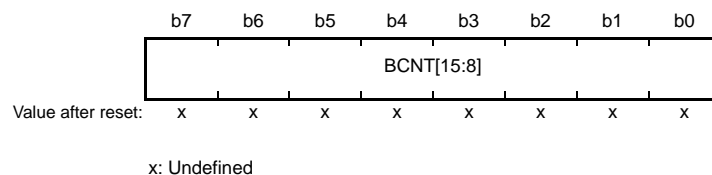
The RMINCNT counter is used for setting and counting the BCD-coded minute value. It counts carries generated once per minute in the second counter.

A value from 00 through 59 (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 24.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): 0008 C404h



The BCNT1 counter is a readable/writable 32-bit binary counter b15 to b8.

The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.

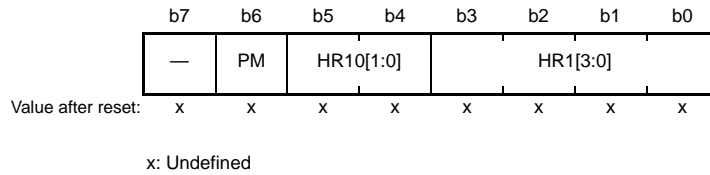
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 24.3.5, Reading 64-Hz Counter and Time.

## 24.2.4 Hour Counter (RHCNT)/Binary Counter 2 (BCNT2)

### (1) In calendar count mode:

Address(es): 0008 C406h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Count	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]	10-Hour Count	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 the set value.	R/W

The RHCNT counter 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 hours mode bit (RCR2.HR24).

When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)

When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)

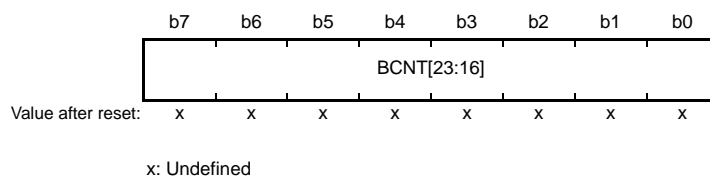
If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

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 24.3.5, Reading 64-Hz Counter and Time.

### (2) In binary count mode:

Address(es): 0008 C406h



The BCNT2 counter is a readable/writable 32-bit binary counter b23 to b16.

The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.

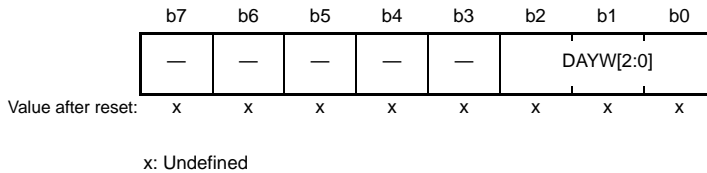
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 24.3.5, Reading 64-Hz Counter and Time.

## 24.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3)

(1) In calendar count mode:

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 the set value.	R/W

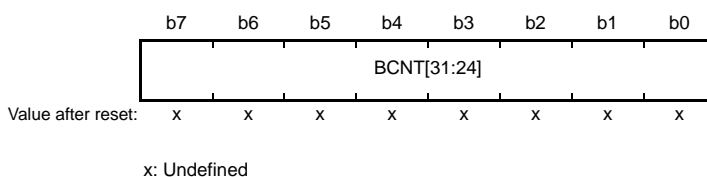
The RWKCNT counter is used for setting and counting in the coded day-of-week value. It counts carries generated once per day in the hour counter.

A value from 0 through 6 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 24.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): 0008 C408h



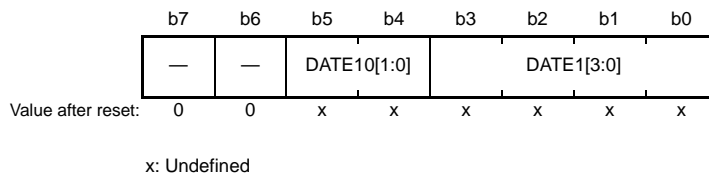
The BCNT3 counter is a readable/writable 32-bit binary counter b31 to b24.

The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter. 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 24.3.5, Reading 64-Hz Counter and Time.

### 24.2.6 Date Counter (RDAYCNT)

Address(es): 0008 C40Ah



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1-Day Count	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]	10-Day Count	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

The RDAYCNT counter is used in calendar count mode.

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.

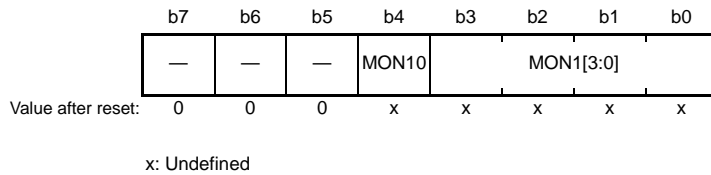
Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.

A value from 01 through 31 (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 24.3.5, Reading 64-Hz Counter and Time.

### 24.2.7 Month Counter (RMONCNT)

Address(es): 0008 C40Ch



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1-Month Count	Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
b4	MON10	10-Month Count	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

The RMONCNT counter is used in calendar count mode.

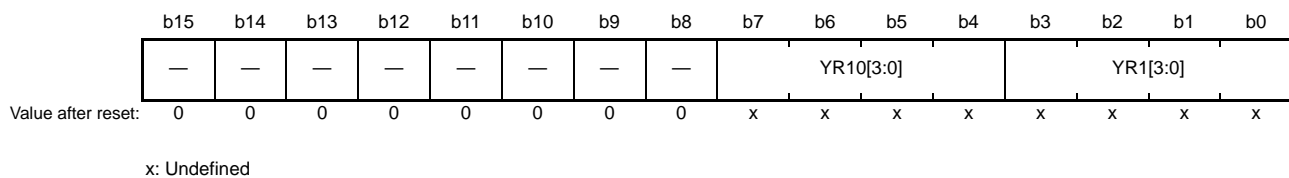
RMONCNT is used for setting and counting the BCD-coded month value. It counts carries generated once per month in the date counter.

A value from 01 through 12 (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 24.3.5, Reading 64-Hz Counter and Time.

### 24.2.8 Year Counter (RYRCNT)

Address(es): 0008 C40Eh



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	1-Year Count	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]	10-Year Count	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

The RYRCNT counter is used in calendar count mode.

RYRCNT is used for setting and counting the BCD-coded year value. It counts carries generated once per year in the month counter.

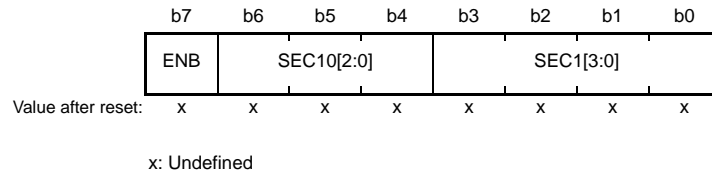
A value from 00 through 99 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 24.3.5, Reading 64-Hz Counter and Time.

## 24.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR)

### (1) In calendar count mode:

Address(es): 0008 C410h



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	0: The register value is not compared with the RSECCNT counter value. 1: The register value is compared with the RSECCNT counter value.	R/W

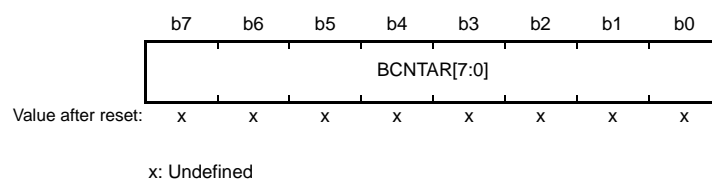
RSECAR is an alarm register corresponding to the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.

RSECAR values from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

This register is set to 00h by an RTC software reset.

### (2) In binary count mode:

Address(es): 0008 C410h

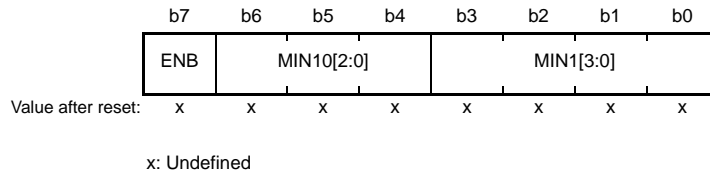


The BCNT0AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b7 to b0. This register is set to 00h by an RTC software reset.

### 24.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR)

(1) In calendar count mode:

Address(es): 0008 C412h



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	0: The register value is not compared with the RMINCNT counter value. 1: The register value is compared with the RMINCNT counter value.	R/W

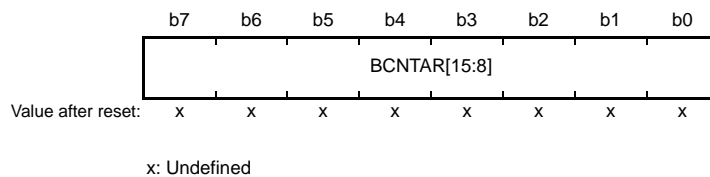
RMINAR is an alarm register corresponding to the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.

RMINAR values from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C412h



The BCNT1AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b15 to b8.

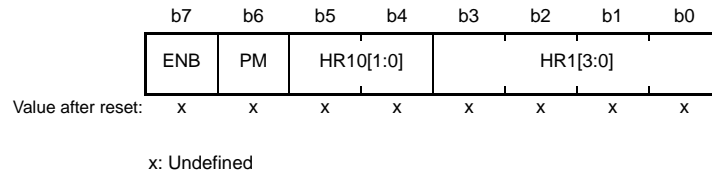
This register is set to 00h by an RTC software reset.



### 24.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR)

(1) In calendar count mode:

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	0: The register value is not compared with the RHCNT counter value. 1: The register value is compared with the RHCNT counter 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 IR flag corresponding to the ALM interrupt 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 (in BCD)

When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)

If a value outside of this range is specified, the RTC does not operate correctly.

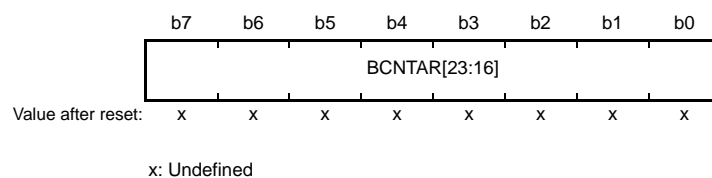
When the RCR2.HR24 bit is 0, be sure to set the PM bit.

When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect.

This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C414h



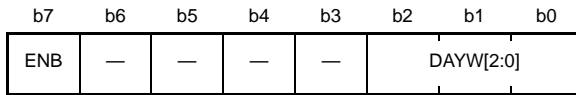
The BCNT2AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b23 to b16.

This register is set to 00h by an RTC software reset.

### 24.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR)

(1) In calendar count mode:

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 the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RWKCNT counter value. 1: The register value is compared with the RWKCNT counter value.	R/W

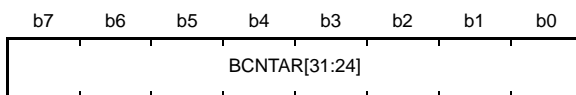
RWKAR is an alarm register corresponding to the coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.

RWKAR values from 0 through 6 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C416h



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

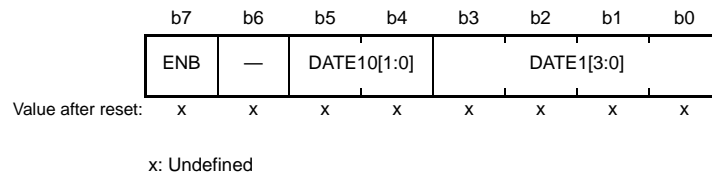
x: Undefined

The BCNT3AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b31 to b24. This register is set to 00h by an RTC software reset.

### 24.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER)

(1) In calendar count mode:

Address(es): 0008 C418h



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 the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RDAYCNT counter value. 1: The register value is compared with the RDAYCNT counter value.	R/W

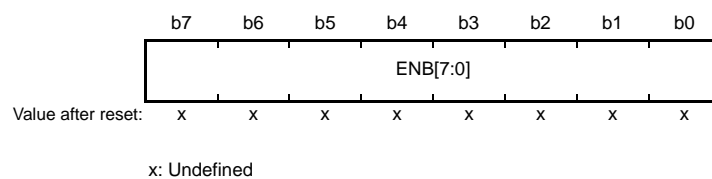
RDAYAR is an alarm register corresponding to the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.

RDAYAR values from 01 through 31 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C418h



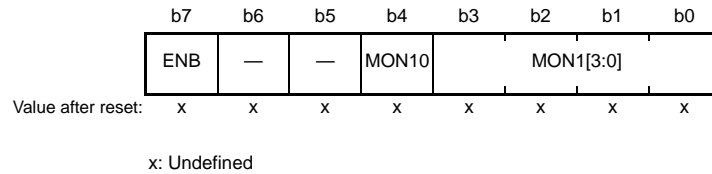
The BCNT0AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b7 to b0. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.

This register is set to 00h by an RTC software reset.

## 24.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER)

### (1) In calendar count mode:

Address(es): 0008 C41Ah



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 the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RMONCNT counter value. 1: The register value is compared with the RMONCNT counter value.	R/W

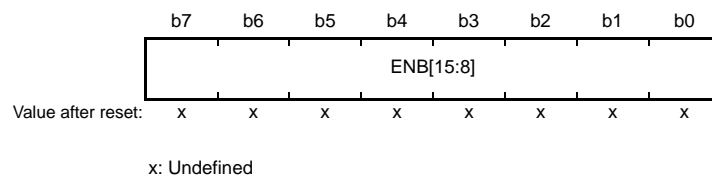
RMONAR is an alarm register corresponding to the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.

RMONAR values from 01 through 12 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

This register is set to 00h by an RTC software reset.

### (2) In binary count mode:

Address(es): 0008 C41Ah



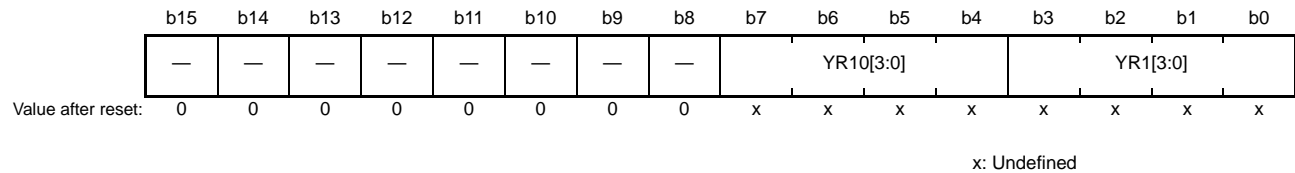
The BCNT1AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b15 to b8. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.

This register is set to 00h by an RTC software reset.

## 24.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER)

### (1) In calendar count mode:

Address(es): 0008 C41Ch



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

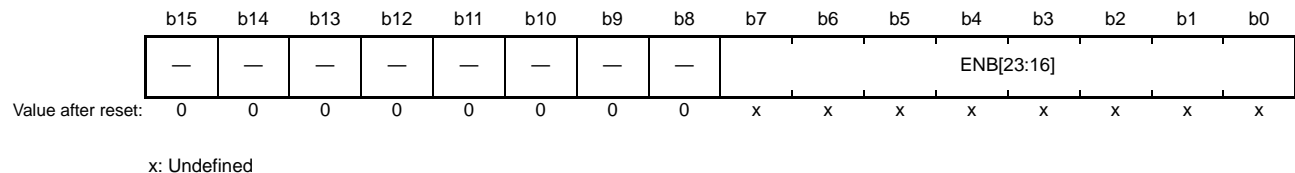
RYRAR is an alarm register corresponding to the BCD-coded year counter RYRCNT.

RYRAR values from 00 through 99 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

This register is set to 0000h by an RTC software reset.

### (2) In binary count mode:

Address(es): 0008 C41Ch



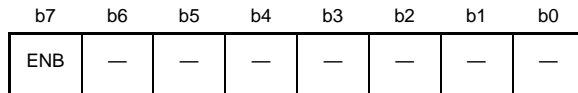
The BCNT2AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b23 to b16. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.

This register is set to 0000h by an RTC software reset.

## 24.2.16 Year Alarm Enable Register (RYRAREN)/Binary Counter 3 Alarm Enable Register (BCNT3AER)

(1) In calendar count mode:

Address(es): 0008 C41Eh



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

x: Undefined

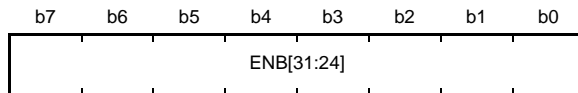
Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RYRCNT counter value. 1: The register value is compared with the RYRCNT counter value.	R/W

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 IR flag corresponding to the ALM interrupt is set to 1.

This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C41Eh



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

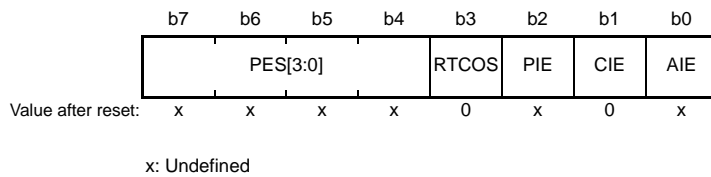
x: Undefined

The BCNT3AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b31 to b24. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.

This register is set to 00h by an RTC software reset.

## 24.2.17 RTC Control Register 1 (RCR1)

Address(es): 0008 C422h



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	RTCOS	RTCOUT Output Select	0: RTCOUT outputs 1 Hz. 1: RTCOUT outputs 64 Hz.	R/W																																				
b7 to b4	PES[3:0]	Periodic Interrupt Select	<table style="font-size: small; border: none;"> <tr> <td style="text-align: right;">b7</td> <td style="text-align: left;">b4</td> <td></td> </tr> <tr> <td>0 1 1 0</td> <td></td> <td>: A periodic interrupt is generated every 1/256 second.</td> </tr> <tr> <td>0 1 1 1</td> <td></td> <td>: A periodic interrupt is generated every 1/128 second.</td> </tr> <tr> <td>1 0 0 0</td> <td></td> <td>: A periodic interrupt is generated every 1/64 second.</td> </tr> <tr> <td>1 0 0 1</td> <td></td> <td>: A periodic interrupt is generated every 1/32 second.</td> </tr> <tr> <td>1 0 1 0</td> <td></td> <td>: A periodic interrupt is generated every 1/16 second.</td> </tr> <tr> <td>1 0 1 1</td> <td></td> <td>: A periodic interrupt is generated every 1/8 second.</td> </tr> <tr> <td>1 1 0 0</td> <td></td> <td>: A periodic interrupt is generated every 1/4 second.</td> </tr> <tr> <td>1 1 0 1</td> <td></td> <td>: A periodic interrupt is generated every 1/2 second.</td> </tr> <tr> <td>1 1 1 0</td> <td></td> <td>: A periodic interrupt is generated every 1 second.</td> </tr> <tr> <td>1 1 1 1</td> <td></td> <td>: A periodic interrupt is generated every 2 seconds.</td> </tr> <tr> <td colspan="3">Other than above: No periodic interrupts are generated.</td> </tr> </table>	b7	b4		0 1 1 0		: A periodic interrupt is generated every 1/256 second.	0 1 1 1		: A periodic interrupt is generated every 1/128 second.	1 0 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.			R/W
b7	b4																																							
0 1 1 0		: A periodic interrupt is generated every 1/256 second.																																						
0 1 1 1		: A periodic interrupt is generated every 1/128 second.																																						
1 0 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.																																								

The RCR1 register is used in both calendar count mode and in binary count mode.

Bits AIE, PIE, and PES[3:0] are updated synchronously with the count source. When the RCR1 register is modified, check that all the bits have been updated before proceeding to the next processing.

### AIE Bit (Alarm Interrupt Enable)

This bit enables or disables alarm interrupt requests.

### CIE Bit (Carry Interrupt Enable)

This bit enables and disables interrupt requests when a carry to the RSECCNT/BCNT0 register occurs, or when a carry to the 64-Hz counter (R64CNT) occurs while reading the 64-Hz counter.

### PIE Bit (Periodic Interrupt Enable)

This bit enables or disabled a periodic interrupt.

### RTCOS Bit (RTCOUT Output Select)

This bit selects the RTCOUT output period. The RTCOS bit must be rewritten while count operation is stopped (the RCR2.START bit is 0) and RTCOUT output is disabled (the RCR2.RTCOE bit is 0). When the RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled. For details on controlling I/O ports, refer to section 19.3.1, Procedure for Specifying I/O Pin Functions.

### PES[3:0] Bits (Periodic Interrupt Select)

These bits specify the period for the periodic interrupt. A periodic interrupt is generated with the period specified by these bits.

## 24.2.18 RTC Control Register 2 (RCR2)

(1) In calendar count mode:

Address(es): 0008 C424h

	b7	b6	b5	b4	b3	b2	b1	b0
	CNTM D	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: Prescaler and time counter are stopped. 1: Prescaler and time counter 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 the target registers for RTC software reset *1 are initialized</li> </ul> </li> <li>In reading               <ul style="list-style-type: none"> <li>0: In normal time operation, or an RTC software reset has completed.</li> <li>1: During an 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 time 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	0: Automatic adjustment is disabled. 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select	0: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every minute. 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 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	CNTMD	Count Mode Select	0: The calendar count mode. 1: The binary count mode.	R/W

Note 1. R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP

The RCR2 register is related to hours mode, automatic adjustment function, enabling the RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

### START Bit (Start)

This bit stops or restarts the prescaler or time counter operation.

The START bit is updated in synchronization with the next count source. When the START bit is modified, check that the bit has been updated before proceeding to the next processing.

### RESET Bit (RTC Software Reset)

This bit initializes the prescaler and registers to be reset by RTC software.

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 set to 0.

When 1 is written to the RESET bit, check that the bit is set 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 1 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 set to 0 after the 30-second adjustment is completed. In case when 1 is written to the ADJ30 bit, check that the bit is set 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 set to 0 by an RTC software reset.

#### **RTCOE Bit (RTCOU Output Enable)**

This bit enables output of a 1-Hz/64-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 (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.

The AADJE bit is set to 0 by an RTC software reset.

#### **AADJP Bit (Automatic Adjustment Period Select)**

This bit selects the automatic-adjustment period.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.

The AADJP bit is set to 0 by an 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.

#### **CNTMD Bit (Count Mode Select)**

This bit specifies whether the RTC count mode is operated in calendar count mode or in binary count mode.

When setting the count mode, execute an RTC software reset and start again from the initial settings.

This bit is updated synchronously with the count source, and its value is fixed before the RTC software reset is completed.

For details on initial settings, refer to section 24.3.1, Outline of Initial Settings of Registers after Power On.

## (2) In binary count mode:

Address(es): 0008 C424h

b7	b6	b5	b4	b3	b2	b1	b0
CNTMD	—	AADJP	AADJE	RTCOE	—	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: The 32-bit binary counter, 64-Hz counter, and prescaler are stopped. 1: The 32-bit binary counter, 64-Hz counter, and prescaler are in normal operation.	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 the target registers for RTC software reset*1 are initialized</li> </ul> </li> <li>In reading               <ul style="list-style-type: none"> <li>0: In normal time operation, or an RTC software reset has completed.</li> <li>1: During an RTC software reset</li> </ul> </li> </ul>	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	RTCOE	RTCOOUT Output Enable	0: RTCOUT output disabled. 1: RTCOUT output enabled.	R/W
b4	AADJE	Automatic Adjustment Enable	0: Automatic adjustment is disabled. 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select	0: Adds or subtracts the RADJ.ADJ[5:0] bits from the prescaler count value every 32 seconds 1: Adds or subtracts the RADJ.ADJ[5:0] bits from the prescaler count value every 8 seconds	R/W
b6	—	Reserved	This bit is undefined. The write value should be 0.	R/W
b7	CNTMD	Count Mode Select	0: The calendar count mode. 1: The binary count mode.	R/W

Note 1. R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP

**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, and then make next settings.

**RESET Bit (RTC Software Reset)**

This bit initializes the prescaler and registers to be reset by RTC software.

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 set to 0.

When 1 is written to the RESET bit, check that the bit is set to 0, and then make next settings.

**RTCOE Bit (RTCOUT Output Enable)**

This bit enables output of a 1-Hz/64-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 an RTCOUT signal is to be output from an external pin, enable the port control as well as setting this bit.

**AADJE Bit (Automatic Adjustment Enable)**

This bit controls (enables or disables) automatic adjustment.

Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.

The AADJE bit is set to 0 by an RTC software reset.

**AADJP Bit (Automatic Adjustment Period Select)**

This bit selects the automatic-adjustment period.

Correction period can be selected from 32 second units or 8 second units in binary count mode.

Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.

The AADJP bit is set to 0 by an RTC software reset.

**CNTMD Bit (Count Mode Select)**

This bit specifies whether the RTC count mode is operated in calendar count mode or in binary count mode.

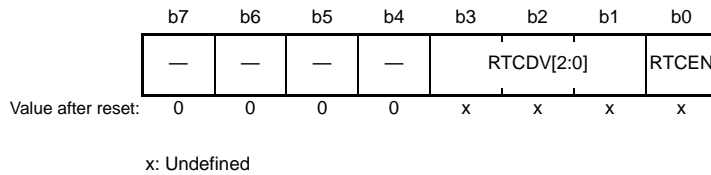
When setting the count mode, execute an RTC software reset and start again from the initial settings.

This bit is updated synchronously with the count source, and its value is fixed before the RTC software reset is completed.

For details on initial settings, refer to section 24.3.1, Outline of Initial Settings of Registers after Power On.

## 24.2.19 RTC Control Register 3 (RCR3)

Address(es): 0008 C426h



Bit	Symbol	Bit Name	Description	R/W
b0	RTCEN	Sub-Clock Oscillator Control	0: Sub-clock oscillator is stopped. 1: Sub-clock oscillator is operating.	R/W
b3 to b1	RTCDV[2:0]	Sub-Clock Oscillator Drive Capacity Control	b3 b1 0 0 0: Medium drive capacity (4.4 pF type) 0 0 1: High drive capacity (6.0 pF type) 0 1 0: Low drive capacity (3.7 pF type) 1 0 0: Standard drive capacity (12.5 pF type) Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RCR3 register is used for controlling the sub-clock oscillator in the clock generation circuit. For details on controlling the sub-clock oscillator, refer to section 9, Clock Generation Circuit.

This register is a function common to calendar count mode and binary count mode.

When this register is modified, check that all the bits have been updated before proceeding to the next processing.

### RTCEN Bit (Sub-Clock Oscillator Control)

The RTCEN bit and a clock generation circuit register control whether to operate or stop the sub-clock oscillator. If one of the bits is set so as to enable the operation, the sub-clock oscillator runs.

When using the sub-clock as the count source to the RTC, set the sub-clock oscillator using the RTCEN bit.

### RTCDV[2:0] Bits (Sub-Clock Oscillator Drive Capacity Control)

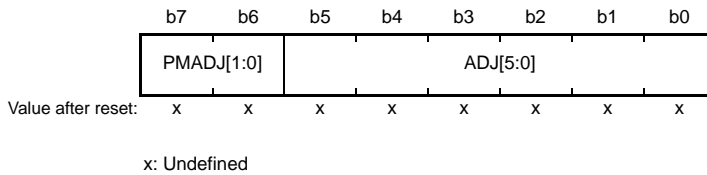
These bits control the drive capacity of the sub-clock oscillator. Set the RTCDV[2:0] bits when the SOSCCR.SOSTP bit is 1 and the RCR3.RTCEN bit is 0.

#### (1) Notes on using a low CL crystal unit

When the signal level of any pin near the XCIN or XCOOUT pin is changed, the oscillation accuracy of the sub-clock oscillator may be affected. The accuracy is affected differently depending on the board traces and how the signal level of any pin near the XCIN or XCOOUT pin is changed. When designing a board using a low CL crystal unit, refer to the application note “Design Guide for Low CL Sub-clock Circuits” (R01AN1012EJ0100) to reduce the influence from noise.

## 24.2.20 Time Error Adjustment Register (RADJ)

Address(es): 0008 C42Eh



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

Adjustment is performed by the addition to or subtraction from the prescaler.

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 (disabling automatic adjustment) 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 before continuing with further processing.

This register is set to 00h by an RTC software reset.

### ADJ[5:0] Bits (Adjustment Value)

These bits specify the adjustment value (the number of sub-clock cycles) 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.

### 24.3 Operation

#### 24.3.1 Outline of Initial Settings of Registers after Power On

After the power is turned on, the initial settings for the clock setting, count mode setting, time error adjustment, time setting, alarm, and interrupt should be performed.

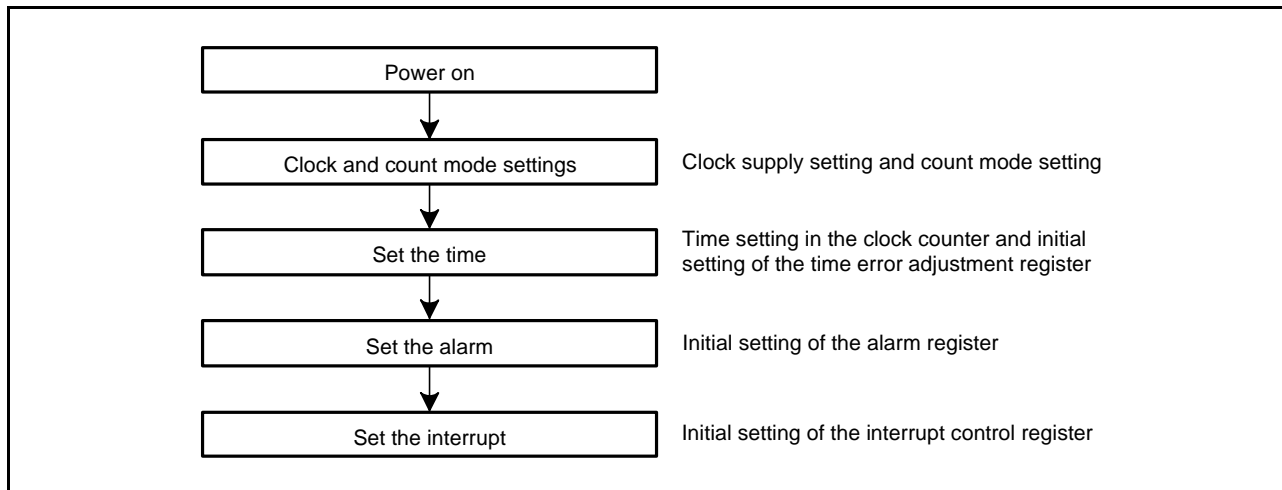
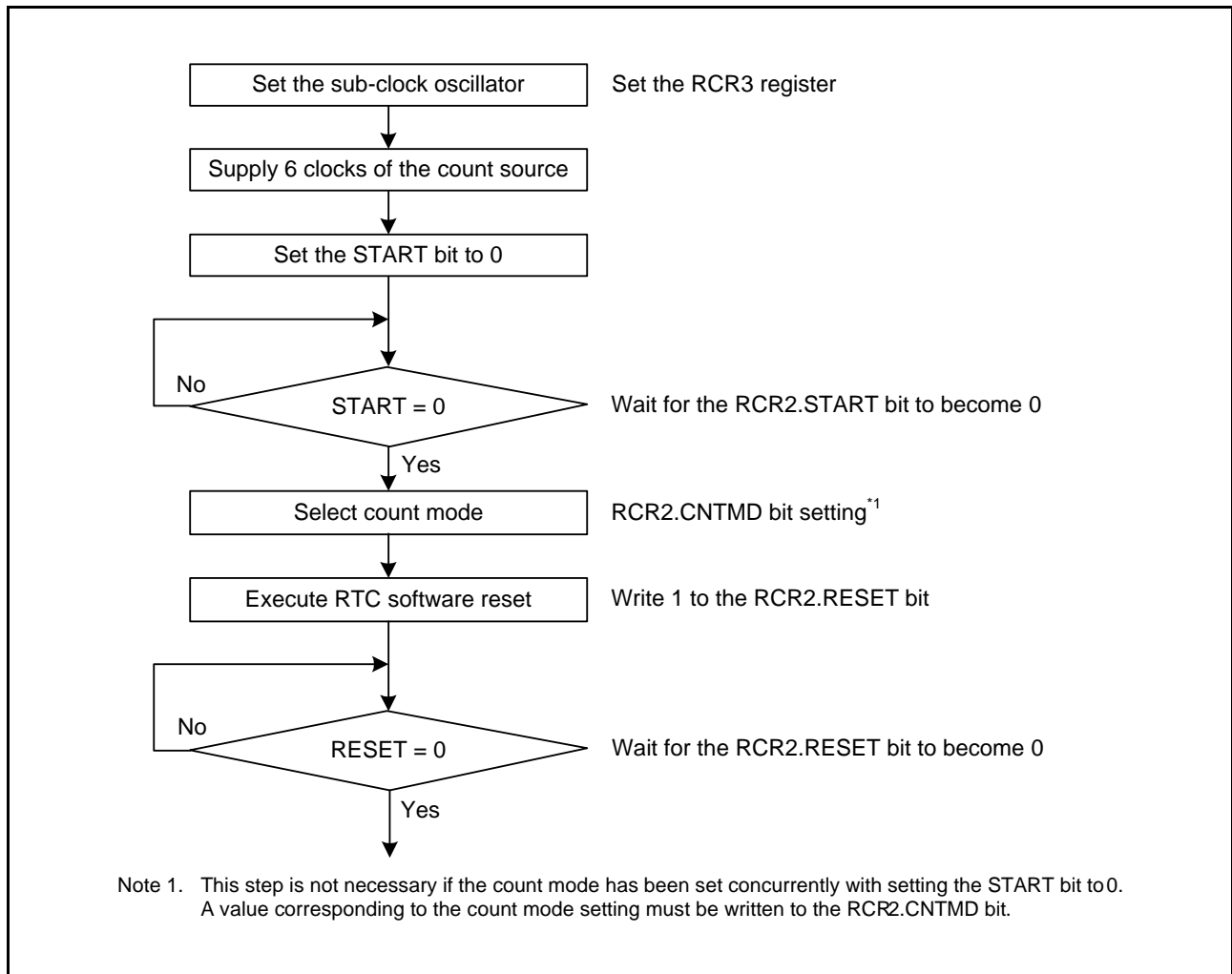


Figure 24.2 Outline of Initial Settings after Power On

### 24.3.2 Clock and Count Mode Setting Procedure

Figure 24.3 shows how to set the clock and the count mode.



**Figure 24.3 Clock and Count Mode Setting Procedure**

### 24.3.3 Setting the Time

Figure 24.4 shows how to set the time.

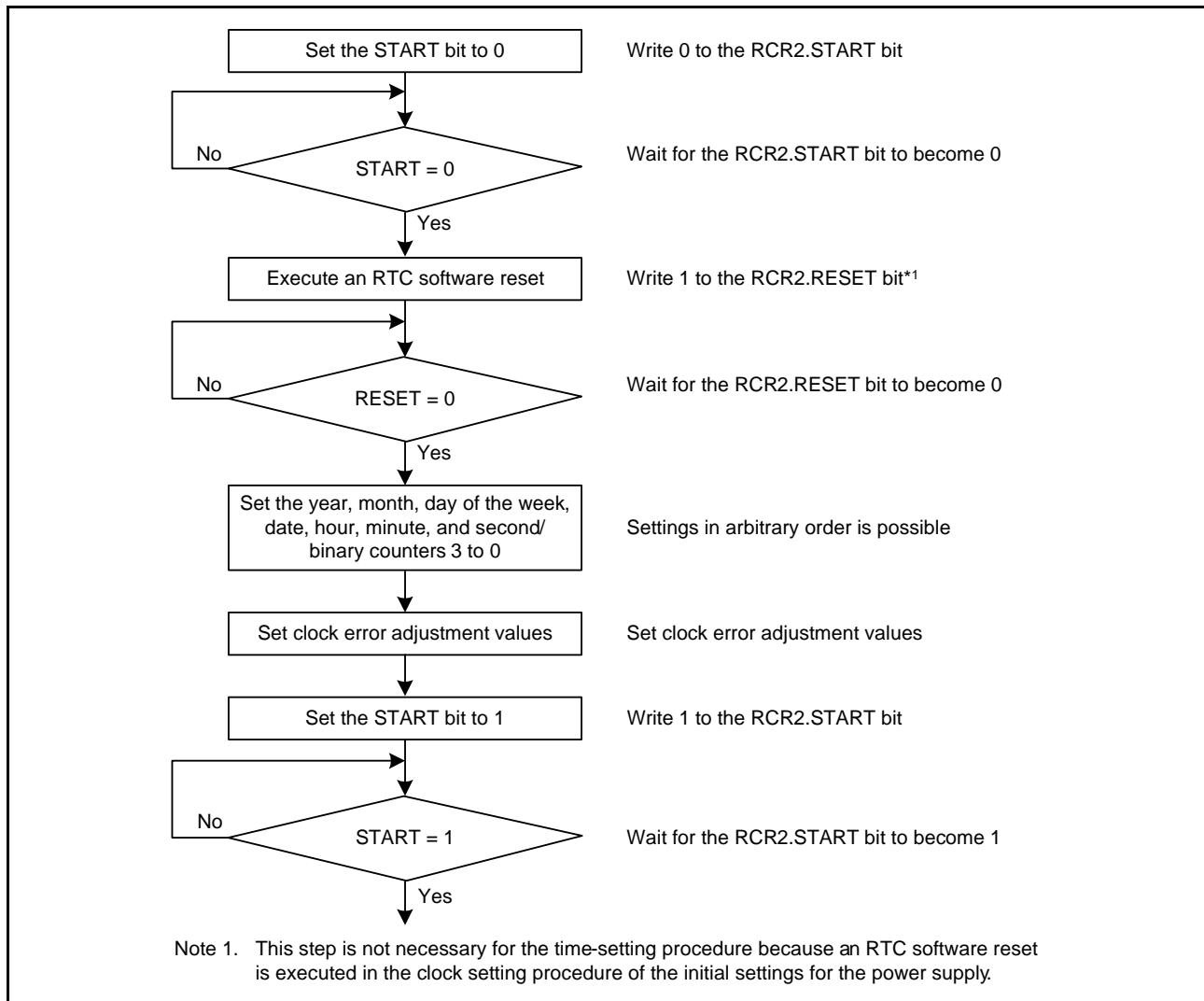


Figure 24.4 Setting the Time

### 24.3.4 30-Second Adjustment

Figure 24.5 shows how to execute 30-second adjustment.

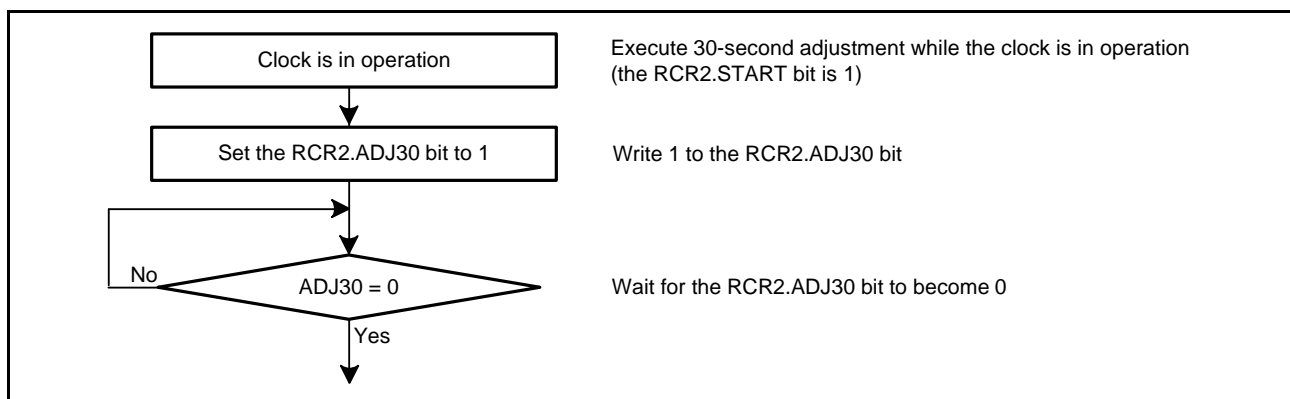


Figure 24.5 30-Second Adjustment



### 24.3.5 Reading 64-Hz Counter and Time

Figure 24.6 shows how to read the 64-Hz counter and time.

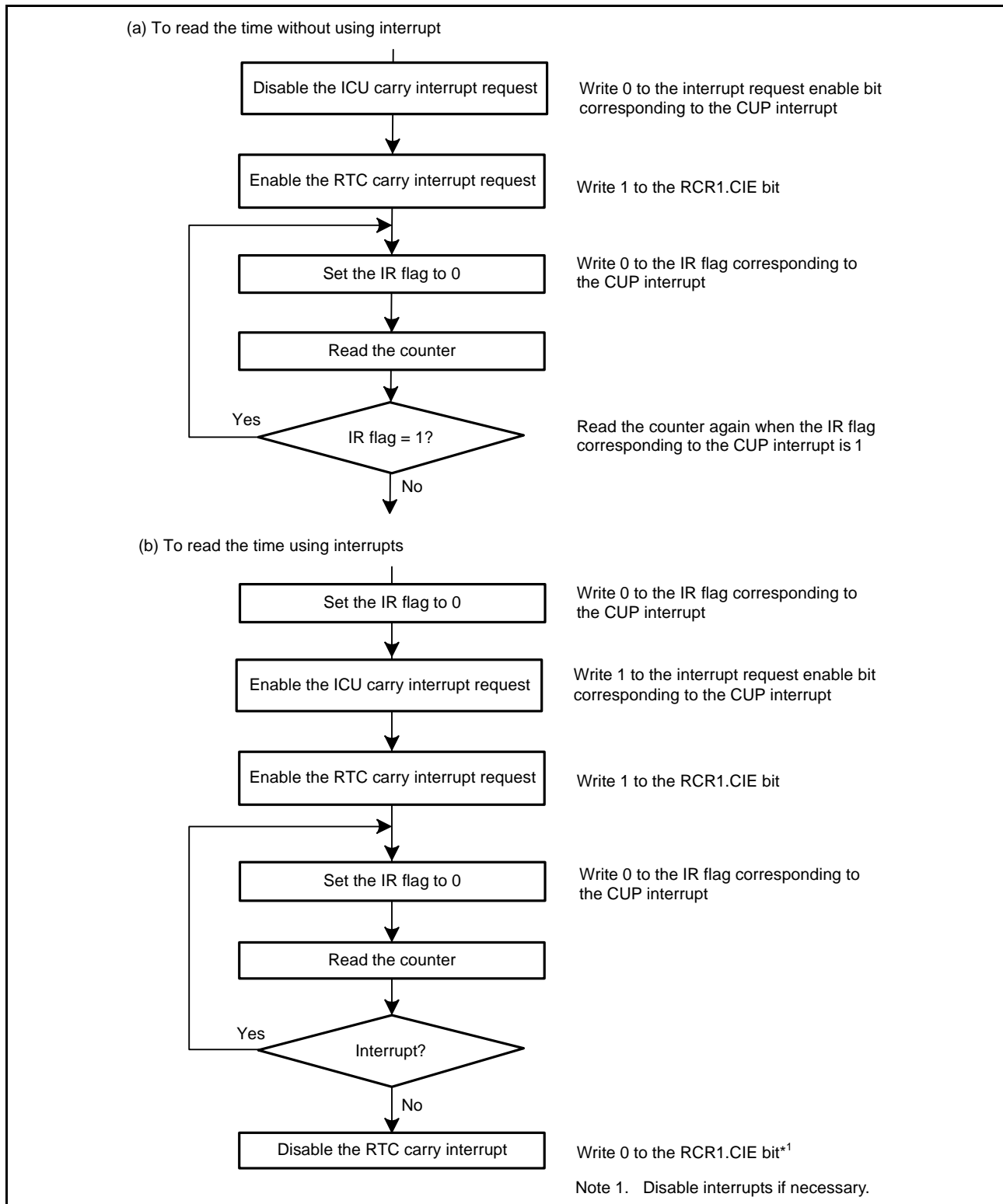
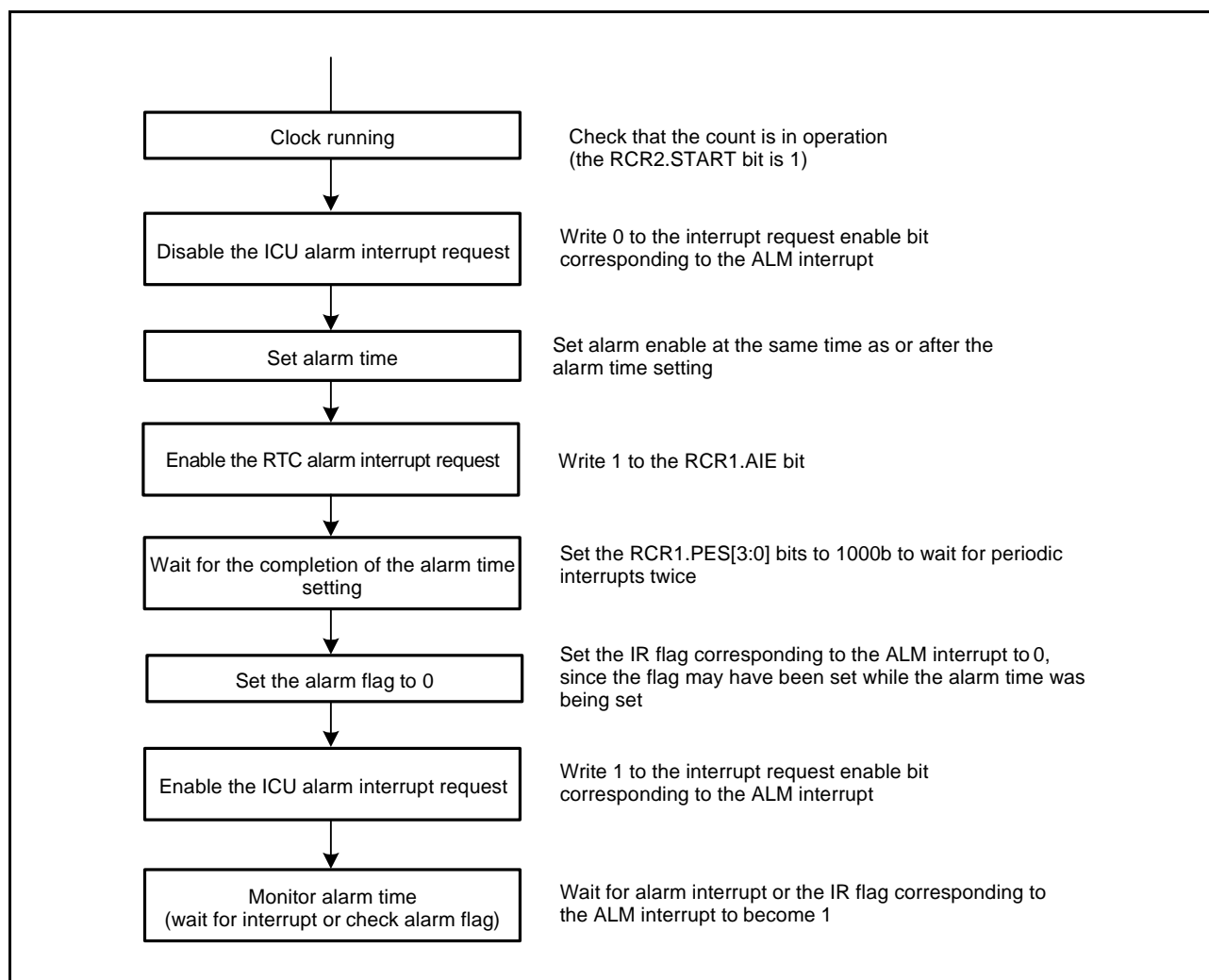


Figure 24.6 Reading Time

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) in Figure 24.6, and the procedure using carry interrupts in (b). To keep the program simple, method (a) should be used in most cases.

### 24.3.6 Alarm Function

Figure 24.7 shows how to use the alarm function.



**Figure 24.7 Using Alarm Function**

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. 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.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the alarm enable register corresponding to the target bit of the alarm, and set the alarm time to the alarm register. For bits that are not target of the alarm, write 0 to the ENB bit of the alarm enable register.

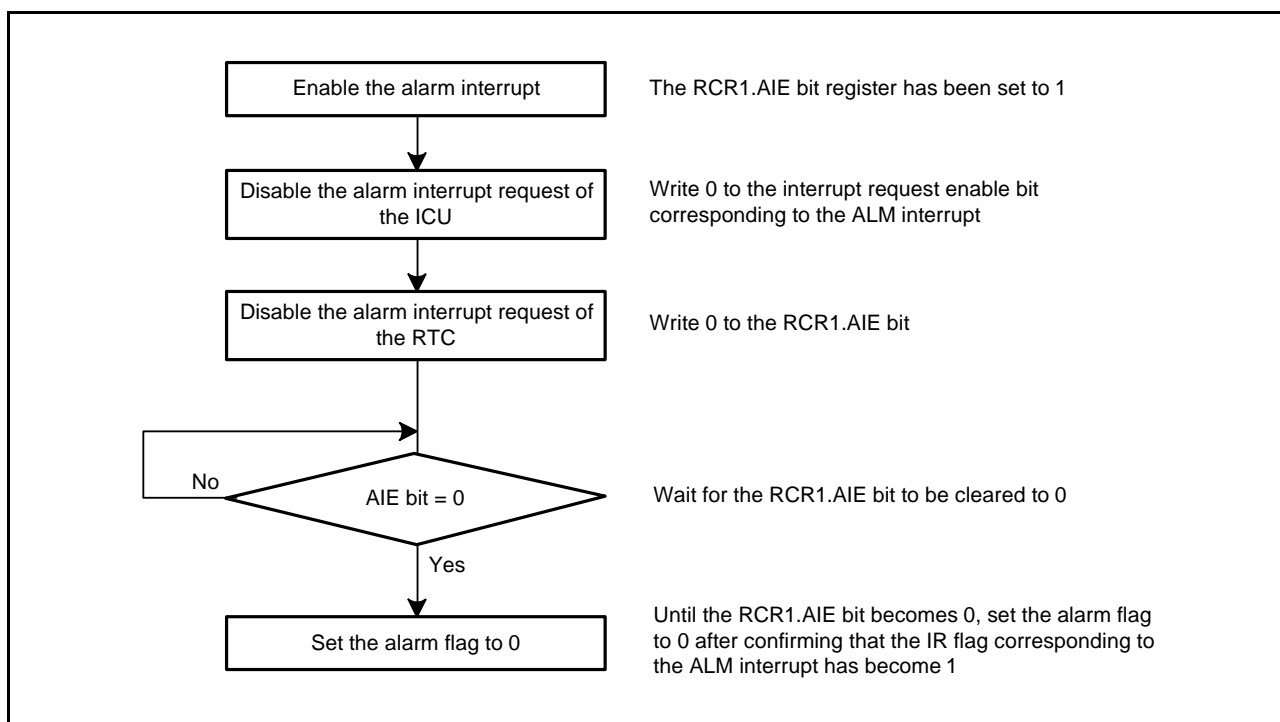
When the counter and the alarm time match, the IR flag corresponding to the ALM interrupt is set to 1. Alarm detection can be confirmed by reading this bit, but an interrupt should be used in most cases. If 1 has been set in the interrupt request enable bit corresponding to the ALM interrupt, an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

Writing 0 sets the IR flag corresponding to the ALM interrupt to 0.

When the counter and the alarm time match in a low power consumption state, the MCU returns from the low power consumption state.

### 24.3.7 Procedure for Disabling Alarm Interrupt

Figure 24.8 shows the procedure for disabling the enabled alarm interrupt request.



**Figure 24.8 Procedure for Disabling Alarm Interrupt Request**

### 24.3.8 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 1 second of operation when the sub-clock is selected, 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.

#### 24.3.8.1 Automatic Adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit 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 RCR2.AADJE bit elapses.

Examples are shown below.

[Example 1] Sub-clock running at 32.769 kHz

Adjustment procedure:

When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 60 (3Ch)

[Example 2] Sub-clock running at 32.766 kHz

Adjustment procedure:

When the sub-clock is running at 32.766 kHz, 1 second elapses every 32,766 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs slow by two clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 20 (14h)

[Example 3] Sub-clock running at 32.764 kHz

Adjustment procedure:

At 32.764 kHz, 1 second elapses on 32,764 clock cycles. Since the RTC operates for 32,768 clock cycles as 1 second, the clock is delayed for four clock cycles per second. In 8 seconds, the delay is 32 clock cycles, so correction can be made by proceeding the clock for 32 clock cycles every 8 seconds.

Register settings when the RCR2.CNTMD bit is 1

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 32 (20h)

### 24.3.8.2 Adjustment by Software

Enable adjustment by software by setting the RCR2.AADJE bit 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:

When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by one clock cycle per second, so adjustment can take the form of setting the clock back by one cycle every second.

Register settings:

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 1 (01h)

This is written to the RADJ register once per 1-second interrupt.

### 24.3.8.3 Procedure for Changing the Mode of Adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

Changing from adjustment by software to automatic adjustment:

- (1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
- (2) Set the RCR2.AADJE bit to 1 (automatic adjustment is enabled).
- (3) Use the RCR2.AADJP bit 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 software to automatic adjustment:

- (1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
- (2) Set the RCR2.AADJE bit to 0 (adjustment by software is enabled).
- (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 the RADJ register.

### 24.3.8.4 Procedure for Stopping Adjustment

Stop adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

## 24.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 24.3 lists interrupt sources for the RTC.

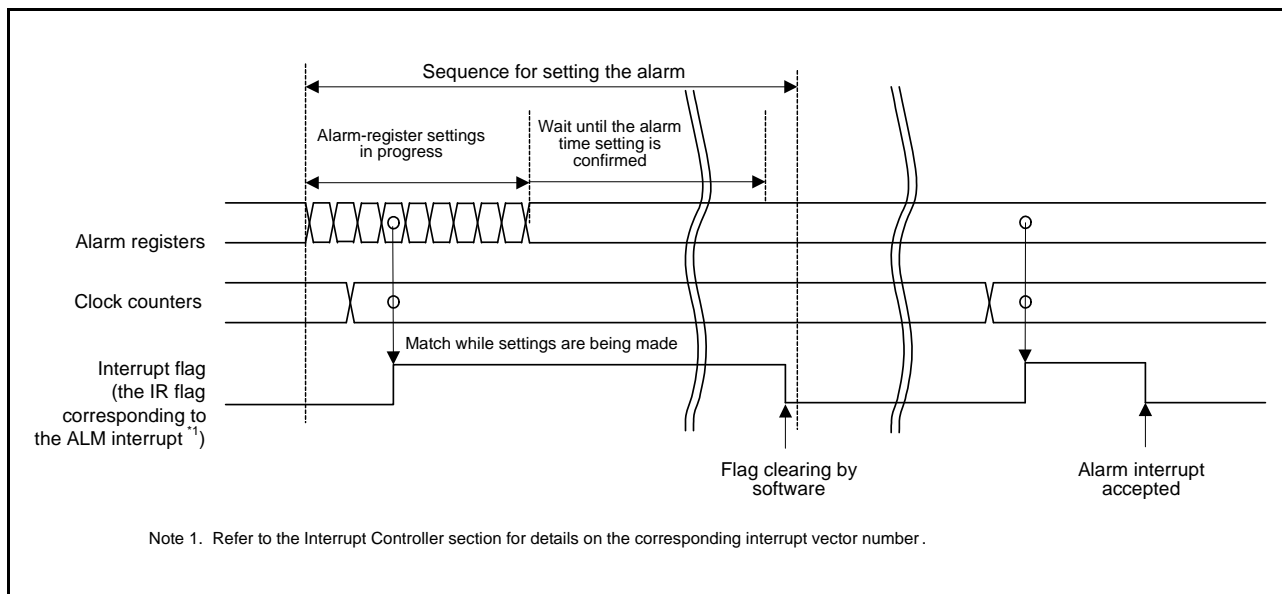
**Table 24.3 RTC Interrupt Sources**

Name	Interrupt Sources
ALM	Alarm interrupt
PRD	Periodic interrupt
CUP	Carry interrupt

### (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 section 24.3.6, Alarm Function).

Since there is a possibility that the interrupt flag may be set to 1 when the settings of the alarm registers match the clock counters, wait for the alarm time settings to be confirmed and set the IR flag corresponding to the ALM interrupt to 0 again after modifying values of the alarm registers. Once the interrupt flag for the alarm interrupt has been set to 1 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 24.9 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 the RCR1.PES[3:0] bits.

(3) Carry interrupt (CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

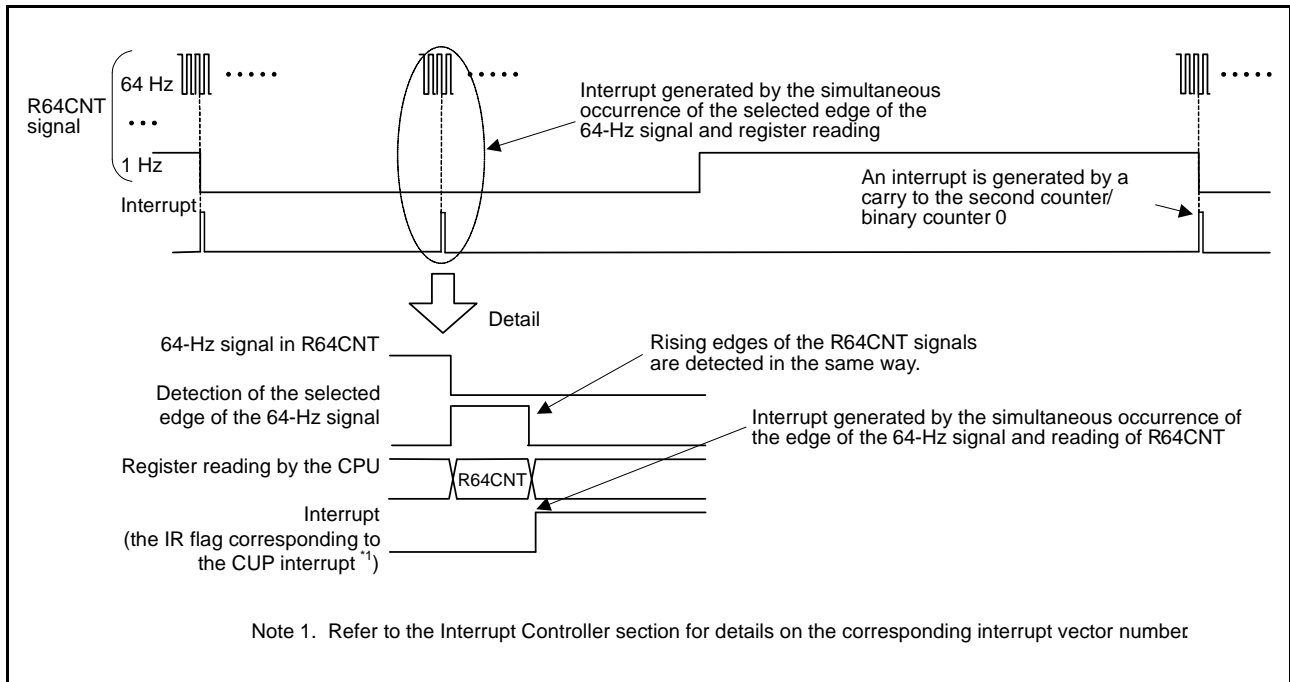


Figure 24.10 Carry Interrupt (CUP) Timing Chart

## 24.5 Usage Notes

### 24.5.1 Register Writing during Counting

The following registers should not be written to during counting (while the RCR2.START bit = 1).

RSECCNT/BCNT0, RMINCNT/BCNT1, RHRCNT/BCNT2, RDAYCNT, RWKCNT/BCNT3, RMONCNT, RYRCNT, RCR1.RTCOS, RCR2.RTCOE, RCR2.HR24

The counter must be stopped before writing to any of the above registers.

### 24.5.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 24.11.

The generation and period of the periodic interrupt can be changed by the setting of the RCR1.PES[3:0] bits. However, since the prescaler, R64CNT, and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting of the RCR1.PES[3:0] bits.

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.

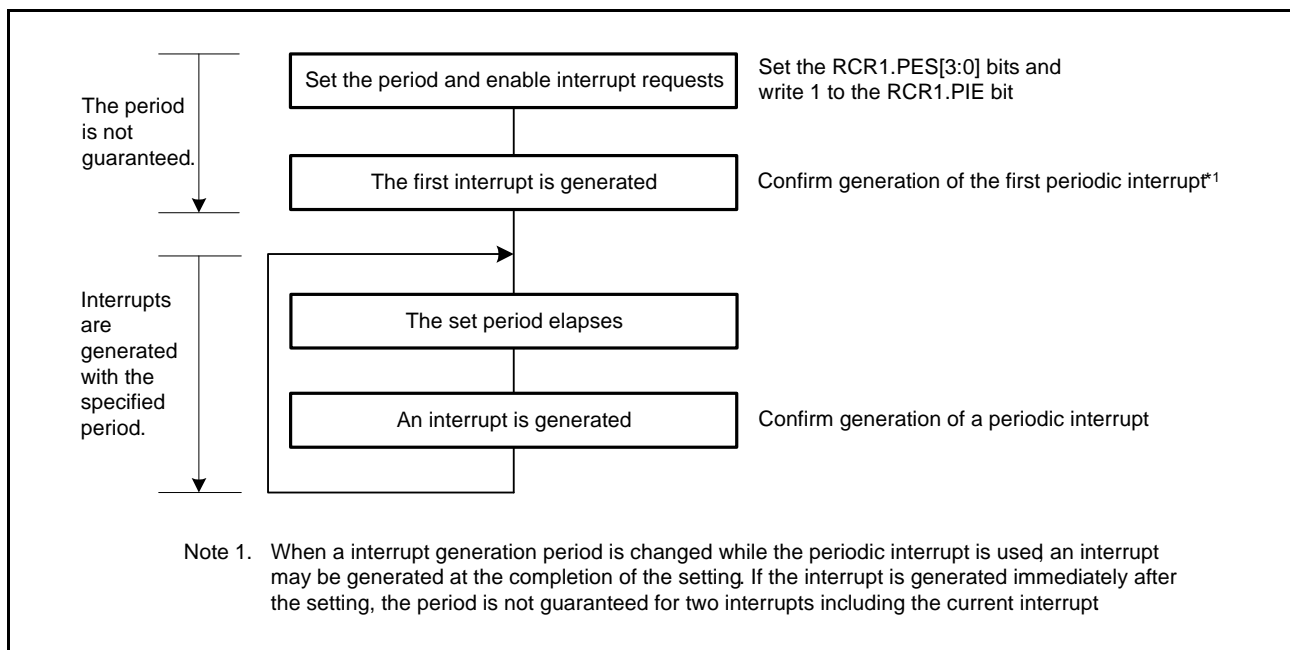


Figure 24.11 Using Periodic Interrupt Function

### 24.5.3 RTCOUT (1-Hz/64-Hz) Clock 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 RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted according to the adjustment value.



#### 24.5.4 Transitions to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (software standby mode) 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.

#### 24.5.5 Notes When Writing to and Reading from Registers

- When reading a counter register such as the second counter/binary counter after having written to the counter register, follow the procedure in section 24.3.5, Reading 64-Hz Counter and Time.
- The value written to the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, or RCR3 register is reflected when four read operations are performed after writing.
- The values written to the RCR1.CIE, RTCOS, and RCR2.RTCOE bits can be read immediately after writing.
- To read the value from the timer counter after return from a reset, or period in software standby mode, wait for 1/128 second while the clock is operating (RCR2.START bit = 1).
- After a reset is generated, write to the RTC register when six cycles of the count source clock have elapsed.

#### 24.5.6 Changing the Count Mode

When changing the count mode (calendar/binary), set the RCR2.START bit to 0, stop counting operation, then start again from the initial setting. For details on initial setting, refer to section 24.3.1, Outline of Initial Settings of Registers after Power On.

### 24.5.7 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 request 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 24.12.

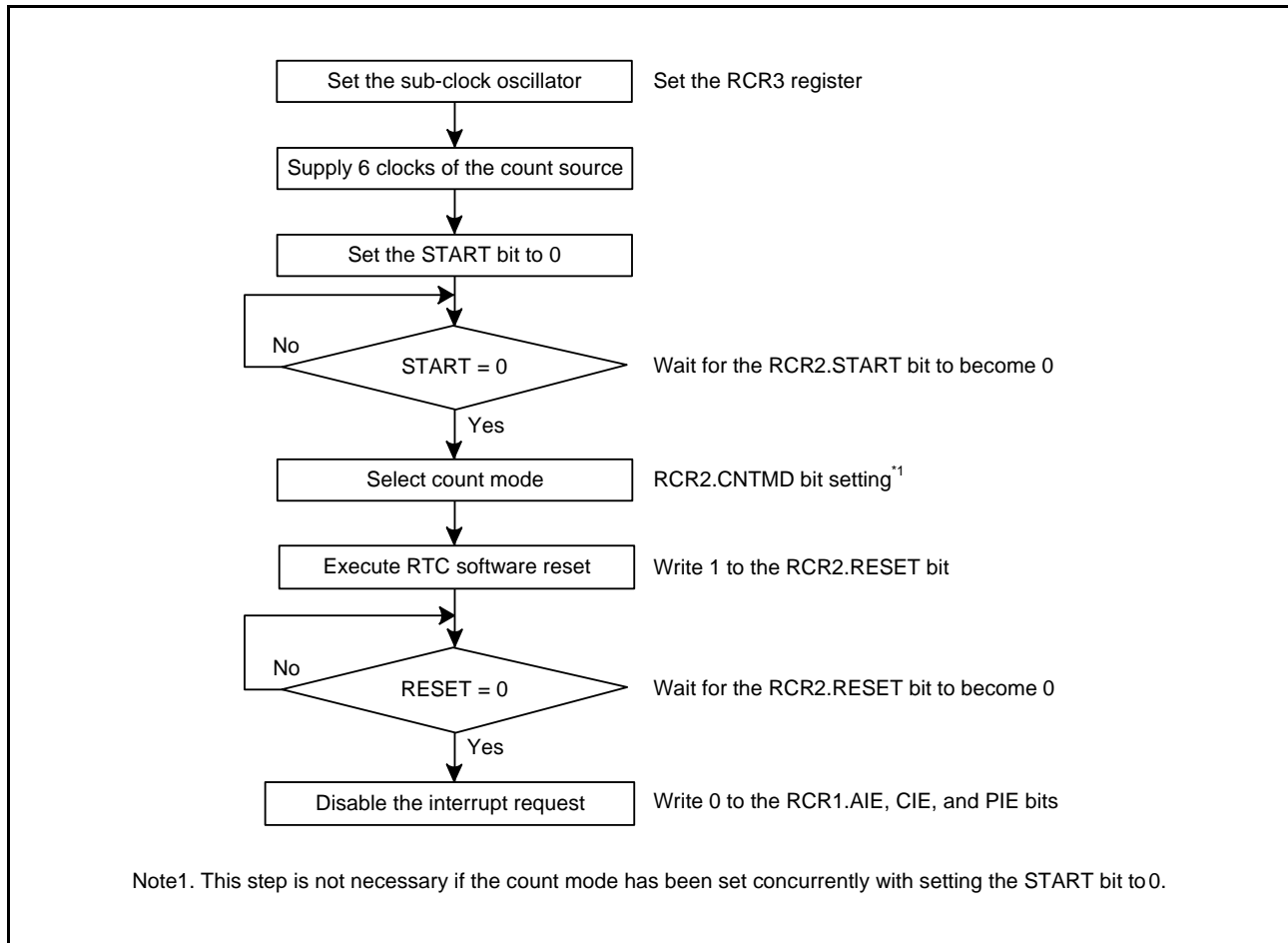


Figure 24.12 Initialization Procedure

## 25. Independent Watchdog Timer (IWDTa)

In this section, “PCLK” is used to refer to PCLKB.

### 25.1 Overview

The independent watchdog timer (IWDT) can be used to detect programs being out of control.

The user can detect when a program runs out of control if an underflow occurs, by creating a program that refreshes the IWDT counter before it underflows.

The functions of the IWDT are different from those of the WDT in the following respects.

- The divided IWDT-dedicated low-speed clock (IWDTCLK) is used as the count source (not affected by the PCLK).
- When making a transition to sleep mode, software standby mode, or deep sleep mode, the IWDTCSSTPR.SLCSTP bit can be used to select whether to stop the counter or not.

Table 25.1 lists the specifications of the IWDT and Figure 25.1 shows a block diagram of the IWDT.

**Table 25.1 IWDT Specifications**

Item	Description
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 (register start mode) by refreshing the counter (writing 00h and then FFh to the IWDTRR register).</li> </ul>
Conditions for stopping the counter	<ul style="list-style-type: none"> <li>• Reset (the down-counter and other registers return to their initial values)</li> <li>• A counter underflows or a refresh error is generated Counting restarts (In auto-start mode, counting automatically restarts after a reset or after a non-maskable interrupt request is output. In register start mode, counting restarts after refreshing.)</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>
Non-maskable interrupt sources	<ul style="list-style-type: none"> <li>• Down-counter underflows</li> <li>• When refreshing is done outside the refresh-permitted period (refresh error)</li> </ul>
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 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 timeout period of the independent watchdog timer (OFS0.IWDTTOPS[1:0] bits)</li> <li>• Selecting the window start position in the independent watchdog timer (OFS0.IWDTRPSS[1:0] bits)</li> <li>• Selecting the window end position in the independent 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, or deep sleep 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 timeout period of the independent watchdog timer (IWDTCR.TOPS[1:0] bits)</li> <li>• Selecting the window start position in the independent watchdog timer (IWDTCR.RPSS[1:0] bits)</li> <li>• Selecting the window end position in the independent watchdog timer (IWDTCR.RPES[1:0] bits)</li> <li>• Selecting the reset output or interrupt request output (IWDTCR.RSTIRQS bit)</li> <li>• Selecting the down-count stop function at transition to sleep mode, software standby mode, or deep sleep mode (IWDTCSSTPR.SLCSTP bit)</li> </ul>

Note 1. Satisfy the frequency of the peripheral module clock (PCLK)  $\geq 4 \times$  (the frequency of the count clock source after division).

To use the IWDT, the IWDT-dedicated clock (IWDTCLK) should be supplied so that the IWDT operates even if the peripheral clock (PCLK) stops. The bus interface and registers operate with PCLK, and the 14-bit counter and control circuits operate with IWDTCLK.

Figure 25.1 is a block diagram of the IWDT.

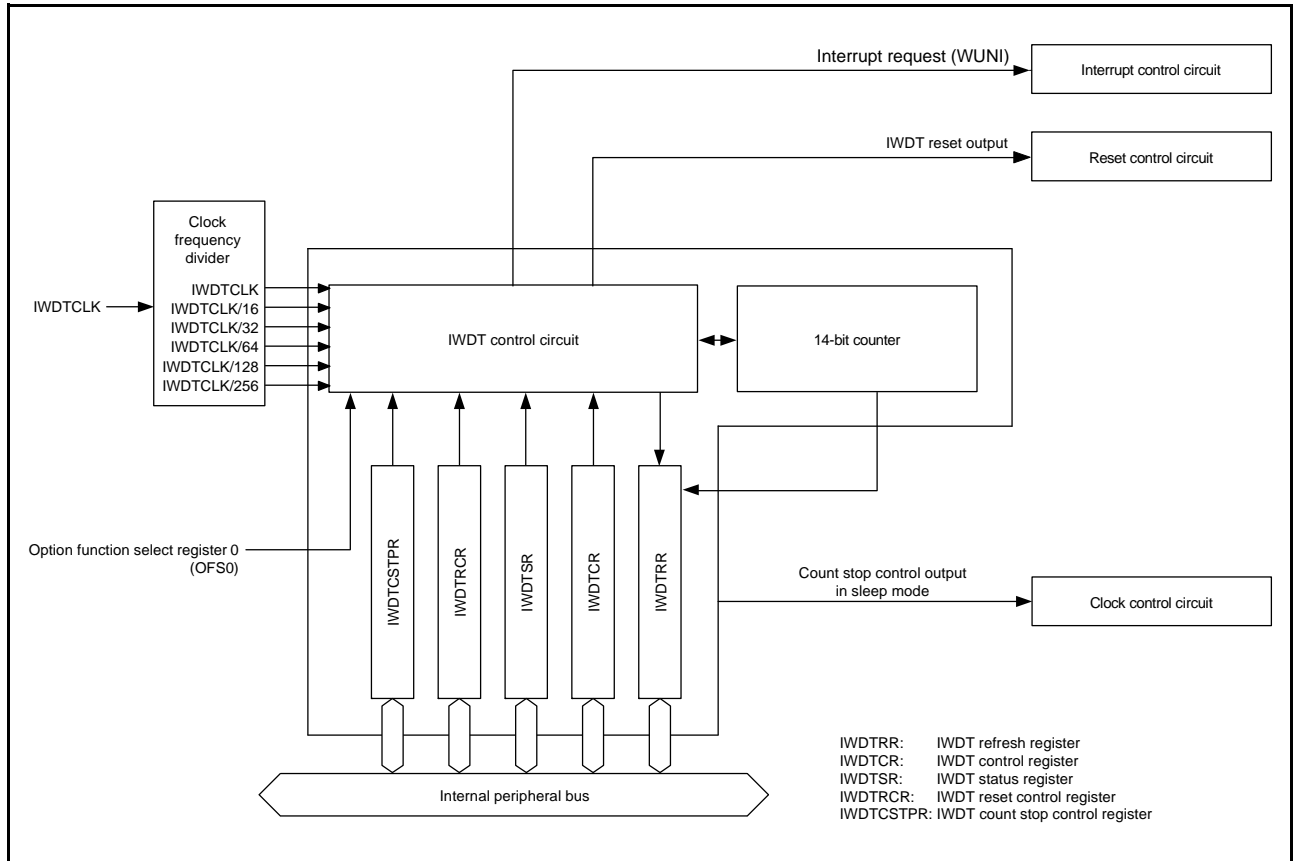
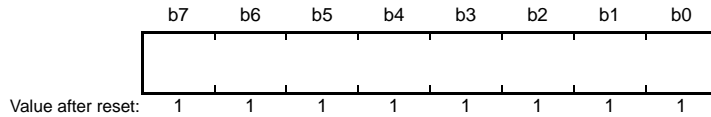


Figure 25.1 IWDT Block Diagram

## 25.2 Register Descriptions

### 25.2.1 IWDt Refresh Register (IWDTRR)

Address(es): 0008 8030h



Bit	Description	R/W
b7 to b0	The counter is refreshed by writing 00h and then writing FFh to this register.	R/W

IWDTRR refreshes the counter of the IWDt.

The counter of the IWDt is refreshed by writing 00h and then writing FFh to IWDTRR (refresh operation) within the refresh-permitted period.

After the counter has been refreshed, it starts counting down from the value selected by the IWDt timeout period select bits (OFS0.IWDTTOPS[1:0]) in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the timeout period select bits (TOPS[1:0]) 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 is FFh.

For details of the refresh operation, refer to section 25.3.3, Refresh Operation.

## 25.2.2 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]	Timeout Period Select	b1 b0 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	R/W
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7 to b4	CKS[3:0]	Clock Division Ratio Select	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 Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R/W
b11, b10	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b13, b12	RPSS[1:0]	Window Start Position Select	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R/W
b15, b14	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

There are some restrictions on writing to the IWDTCR register. For details, refer to section 25.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the settings in the IWDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the IWDTCR register can also be made in option function select register 0 (OFS0). For details, refer to section 25.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDt Registers.

**TOPS[1:0] Bits (Timeout Period Select)**

These bits select the timeout period (period until the counter underflows) from among 128, 512, 1024, or 2048 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.

After the counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDTCLK cycles) until the counter underflows.

Relations between the CKS[3:0] and TOPS[1:0] bit setting, the timeout period, and the number of IWDTCLK cycles are listed in Table 25.2.

**Table 25.2 Settings and Timeout Periods**

CKS[3:0] Bits				TOPS[1:0] Bits		Clock Division Ratio	Timeout Period (Number of Cycles)	Cycles of IWDTCLK
b7	b6	b5	b4	b1	b0			
0	0	0	0	0	0	IWDTCLK	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	IWDTCLK/16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	IWDTCLK/32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	IWDTCLK/64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	IWDTCLK/128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	IWDTCLK/256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

**CKS[3:0] Bits (Clock Division Ratio Select)**

These bits select the IWDTCLK clock division ratio from among division by 1, 16, 32, 64, 128, and 256. Combination with the TOPS[1:0] bit setting, a count period between 128 and 524288 cycles of the IWDTCLK clock can be selected for the IWDT.

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

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the counter. The window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

The counter values for the window start and end positions selected by setting the RPSS[1:0] and RPES[1:0] bits change depending on the TOPS[1:0] bit setting.

Table 25.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

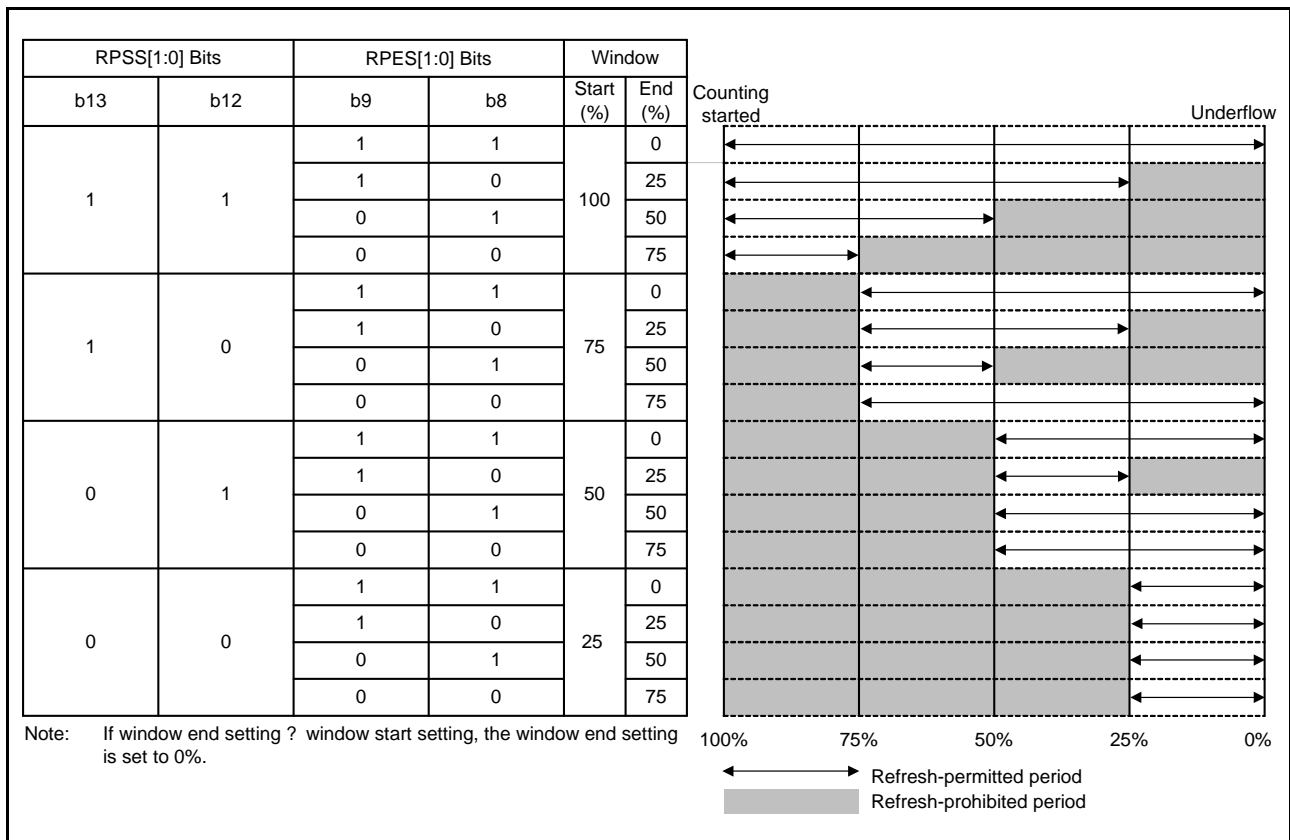
**Table 25.3 Relationship between Timeout Period and Window Start and End Counter Values**

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
b1	b0	Cycles	Counter Value	100%	75%	50%	25%
0	0	128	007Fh	007Fh	005Fh	003Fh	001Fh
0	1	512	01FFh	01FFh	017Fh	00FFh	007Fh
1	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
1	1	2048	07FFh	07FFh	05FFh	03FFh	01FFh

**RPSS[1:0] Bits (Window Start Position Select)**

These bits select a counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.

Figure 25.2 shows the relationship between of the RPSS[1:0] and RPES[1:0] bit setting and the refresh-permitted and refresh-prohibited periods.

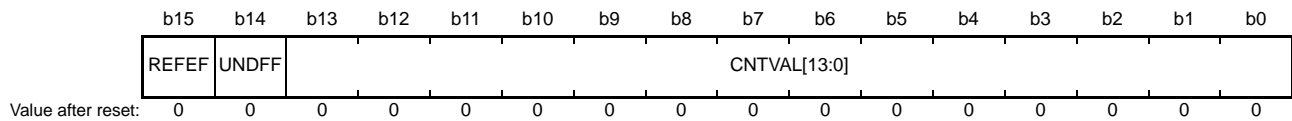


**Figure 25.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period**



### 25.2.3 IWDT Status Register (IWDTSR)

Address(es): 0008 8034h



Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Counter Value	Value counted by the 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 (Counter Value)

Read these bits to confirm the counter value of the counter, but note that the read value may differ from the actual count by a value of one count.

#### UNDFE Flag (Underflow Flag)

Read this bit to confirm whether or not an underflow has occurred in the counter.

The value 1 indicates that the counter has underflowed. The value 0 indicates that the counter has not underflowed.

Write 0 to the UNDFE flag to set the value to 0. Writing 1 has no effect.

#### REFEF Flag (Refresh Error Flag)

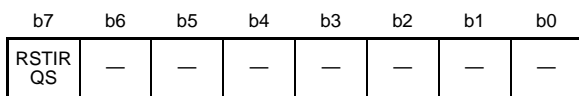
Read this bit to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period).

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.

### 25.2.4 IWDt Reset Control Register (IWDTRCR)

Address(es): 0008 8036h



Value after reset:    1    0    0    0    0    0    0    0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request output is enabled. 1: Reset output is enabled.	R/W

There are some restrictions on writing to the IWDTRCR register. For details, refer to section 25.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the IWDTRCR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTRCR register can also be made in option function select register 0. For details, refer to section 25.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDt Registers.

### 25.2.5 IWDT Count Stop Control Register (IWDTCSSTPR)

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. Writing to these bits has no effect.	R
b7	SLCSTP	Sleep-Mode Count Stop Control	0: Count stop is disabled. 1: Count is stopped at a transition to sleep mode, software standby mode, or deep sleep mode.	R/W

IWDTCSSTPR controls whether to stop the IWDT counter in a low power consumption state. There are some restrictions on writing to the IWDTCSSTPR register. For details, refer to section 25.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSSTPR Registers.

In auto-start mode, the IWDTCSSTPR register settings are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTCSSTPR register can also be made in option function select register 0 (OFS0). For details, refer to section 25.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

#### SLCSTP Bit (Sleep-Mode Count Stop Control)

This bit selects whether to stop counting at a transition to sleep mode, software standby mode, or deep sleep mode.

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

For option function select register 0 (OFS0), refer to section 25.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

## 25.3 Operation

### 25.3.1 Count Operation in Each Start Mode

Select the IWDT start mode by setting the IWDT start mode select bit (OFS0.IWDTSTRT) in 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 option function select register 0 (OFS0) is enabled, and counting automatically starts after reset.

#### 25.3.1.1 Register Start Mode

When the IWDT start mode select bit (OFS0.IWDTSTRT) in 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 the reset state is released, set the clock division ratio, window start and end positions, and timeout period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the counter stop control at transitions to low power consumption states in the IWDTCSTPR register. Then refresh the counter to start counting down from the value selected by setting the timeout period select bits (IWDTCR.TOPS[1:0]).

Thereafter, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because the counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request (WUNI). Set the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) to select either reset output or interrupt request output.

Figure 25.3 shows an example of operation under the following conditions.

- The IWDT start mode select bit (OFS0.IWDTSTRT) is 1 (register start mode)
- The IWDT reset interrupt request select bit (IWDTRCR.RSTIRQS) is 1 (reset output is enabled)
- The IWDT window start position select bits (IWDTCR.RPSS[1:0]) are 10b (75%)
- The IWDT window end position select bits (IWDTCR.RPES[1:0]) are 10b (25%)

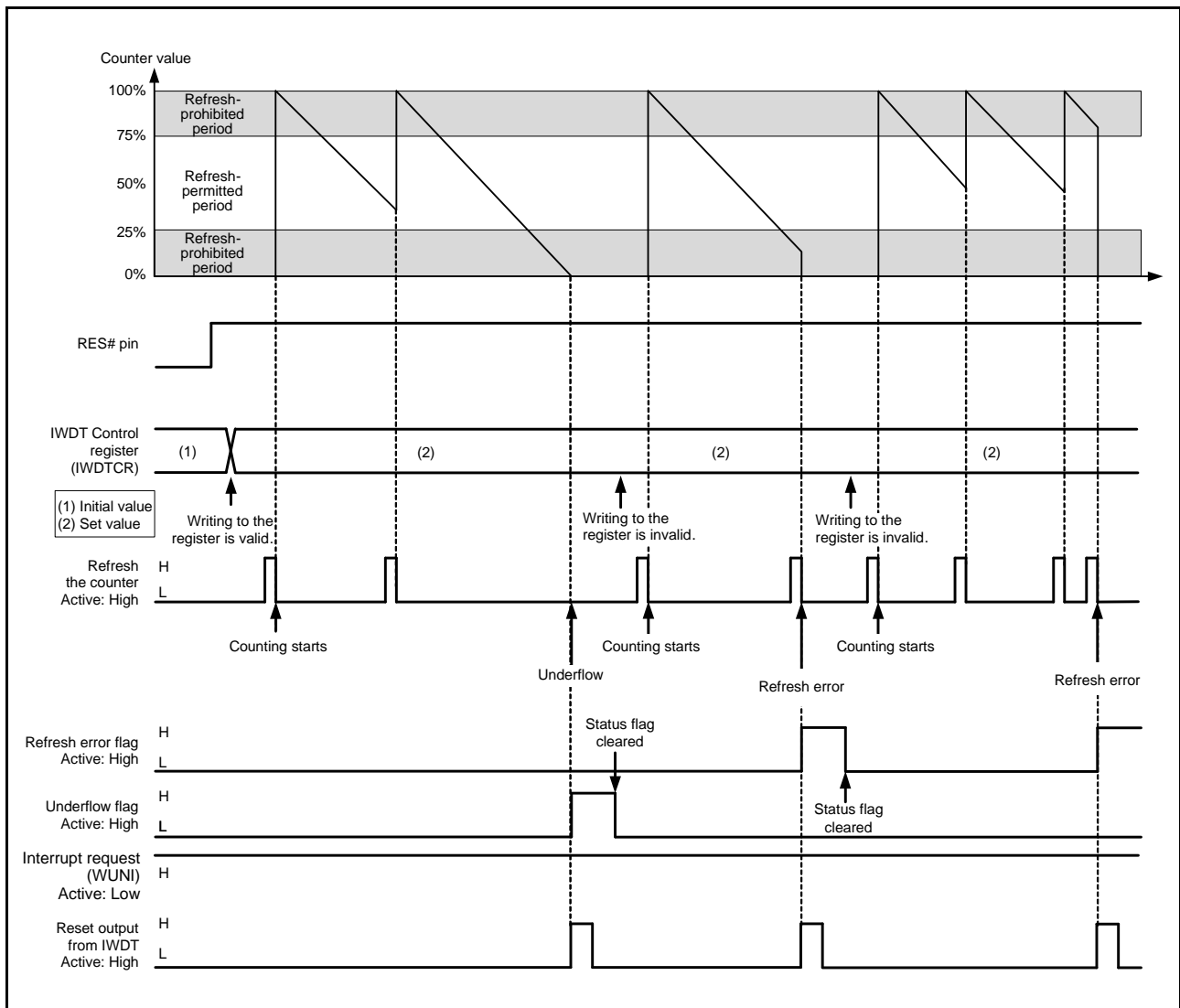


Figure 25.3 Operation Example in Register Start Mode

### 25.3.1.2 Auto-Start Mode

When the IWDT start mode select bit (OFS0.IWDTSTRT) in option function select register 0 is 0, auto-start mode is selected, and the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDCSTPR) are disabled.

Within the reset state, the clock division ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter stop control at transitions to low power consumption states should be specified in option function select register 0 (OFS0). When the reset state is released, the counter automatically starts counting down from the value selected by the IWDT timeout period select bits (OFS0.IWDTTOPS[1:0]).

After that, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because refreshing of the counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the IWDT outputs the reset signal or non-maskable interrupt request (WUNI). After the reset signal or non-maskable interrupt request (WUNI) is generated, the counter reloads the timeout period after counting for one cycle, and restarts counting. Set the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) to select either reset output or interrupt request output.

Figure 25.4 shows an example of operation under the following conditions.

- The IWDT start mode select bit (OFS0.IWDTSTRT) is 0 (auto-start mode)
- The IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) is 0 (non-maskable interrupt request output is enabled)
- The IWDT window start position select bits (OFS0.IWDTRPSS[1:0]) are 10b (75%)
- The IWDT window end position select bits (OFS0.IWDTRPES[1:0]) are 10b (25%)

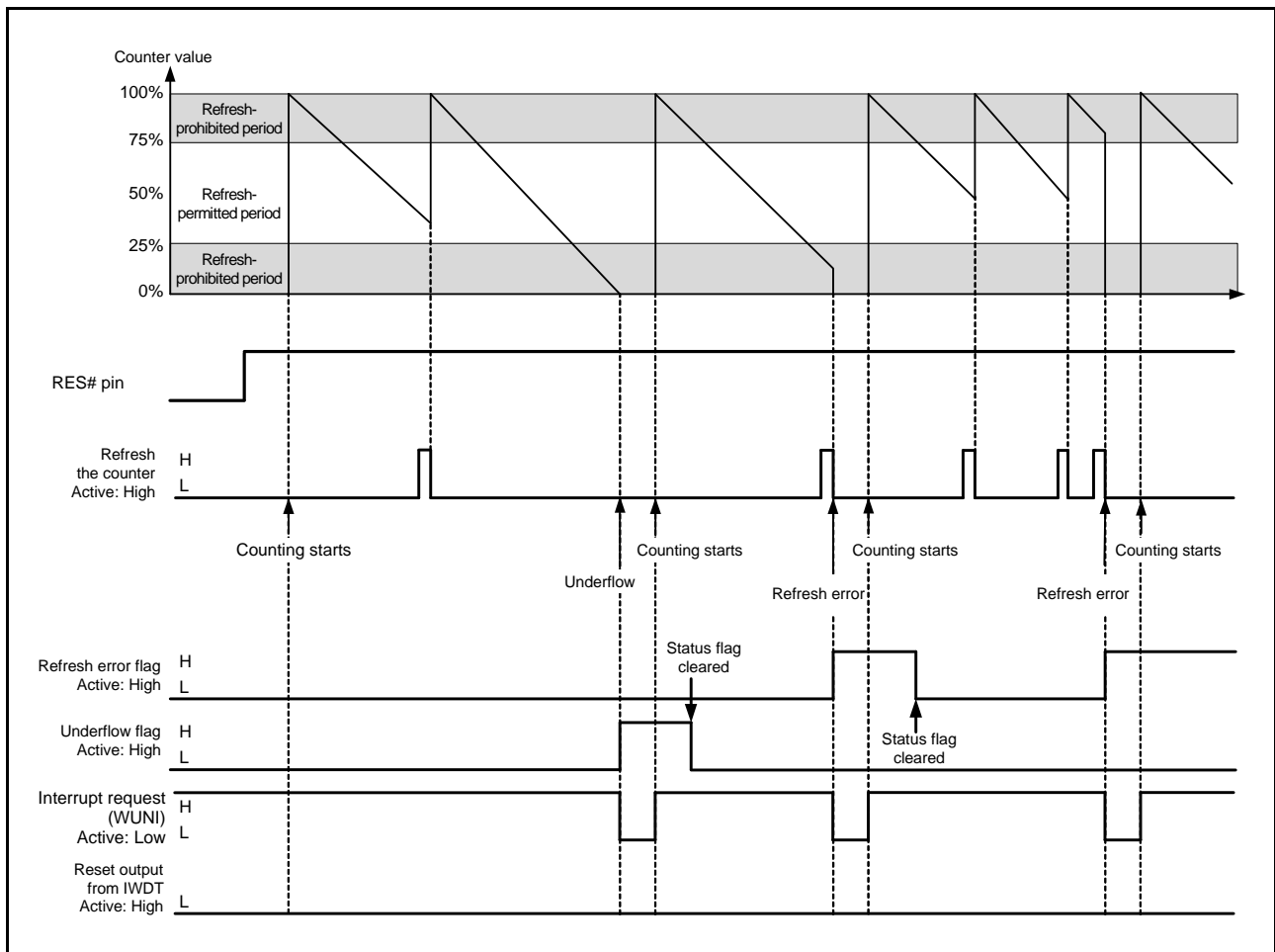


Figure 25.4 Operation Example in Auto-Start Mode

### 25.3.2 Control over Writing to the IWDTCR, IWDTRCR, and IWDTCS TPR Registers

Writing to the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), or IWDT count stop control register (IWDTCS TPR) 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 IWDTCS TPR is written to, the protection signal in the IWDT becomes 1 to protect IWDTCR, IWDTRCR, and IWDTCS TPR 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 25.5 shows control waveforms produced in response to writing to the IWDTCR register.

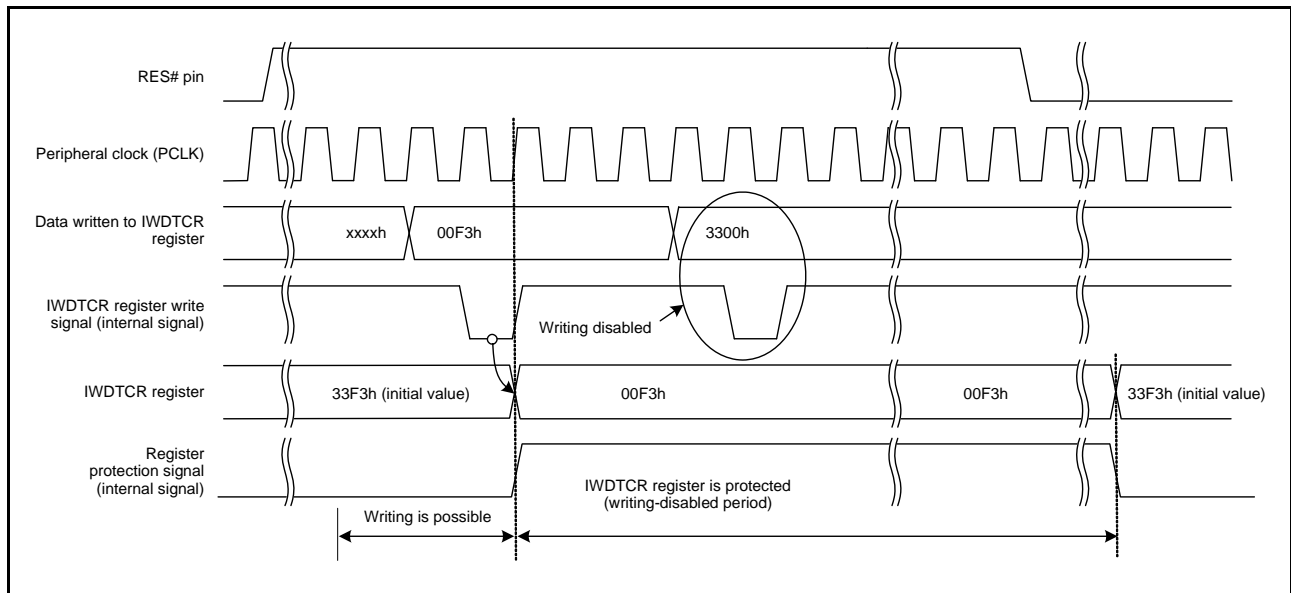


Figure 25.5 Control Waveforms Produced in Response to Writing to the IWDTCR Register



### 25.3.3 Refresh Operation

The counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the IWDT refresh register (IWDTRR). If a value other than FFh is written after 00h, the counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing 00h and then FFh to the 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 (nth time) → FFh is valid and correct refreshing will be done. Even when the first value written before 00h is not 00h, correct refreshing will be done if the operation contains the set of writing 00h → FFh. Moreover, even if a register other than 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 (nth 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 counter requires up to four cycles of the signal for counting (the clock division ratio selection bits (IWDTCR.CKS[3:0]) determine how many cycles of the IWDT-dedicated clock (IWDTCCLK) 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 counter can be checked by the counter bits (IWDTSR.CNTVAL[13:0]).

[Sample refreshing timings]

- When the window start position is set to 03FFh, 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 0403h (four-count cycles before 03FFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to 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 25.6 shows the IWDT refresh-operation waveforms when  $PCLK > IWDTCLK$  and clock division ratio =  $IWDTCLK$ .

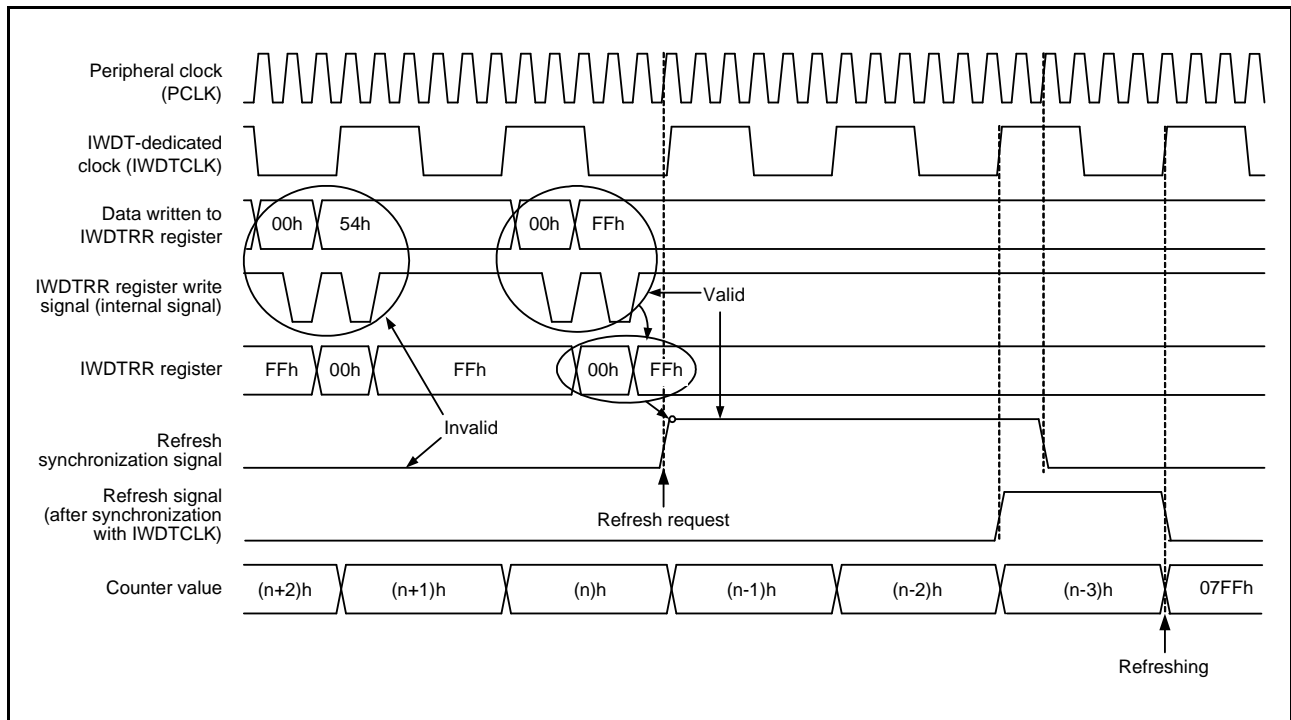


Figure 25.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

### 25.3.4 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.

After 0 is written to each flag, up to three IWDTCLK cycles and two PCLK cycles are required before the value is reflected.

### 25.3.5 Reset Output

When the reset interrupt selection bit (IWDTRCR.RSTIRQS) is set to 1 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDRSTIRQS) in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output when an underflow in the counter or a refresh error occurs.

In register start mode, the counter is initialized (all bits set to 0) and kept in that state after assertion of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing.

In auto-start mode, counting down automatically starts after the reset output.

### 25.3.6 Interrupt Sources

When the reset interrupt selection bit (IWDTRCR.RSTIRQS) is set to 0 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDRSTIRQS) in option function select register 0 (OFS0) is set to 0 in auto-start mode, an interrupt (WUNI) signal is output when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt. For details, refer to section 14, Interrupt Controller (ICUb).

**Table 25.4 IWDT Interrupt Source**

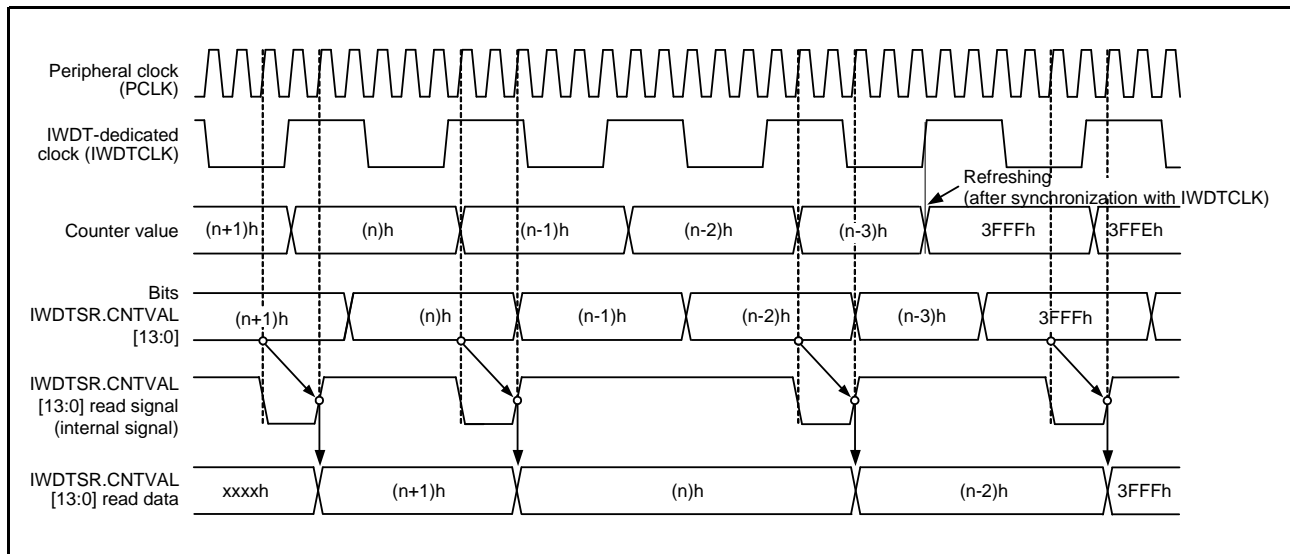
Name	Interrupt Source	DTC Activation
WUNI	Counter underflow Refresh error	Not possible

### 25.3.7 Reading the Counter Value

As the counter in IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral clock (PCLK) and stores it in the counter value bits (IWDTSR.CNTVAL[13:0]) of the IWDT status register. Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.

Reading the counter value requires multiple PCLK clock cycles (up to four clock cycles), and the read counter value may differ from the actual counter value by a value of one count.

Figure 25.7 shows the processing for reading the IWDT counter value when  $PCLK > IWDTCLK$  and clock division ratio = IWDTCLK.



**Figure 25.7 Processing for Reading IWDT Counter Value**  
 (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

### 25.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 25.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.

Do not change the OFS0 register setting during IWDT operation.

For details on option function select register 0 (OFS0), refer to section 7.2.1, Option Function Select Register 0 (OFS0).

**Table 25.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers**

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.IWDTSTRT = 0	IWDT Registers (Enabled in Register Start Mode) OFS0.IWDTSTRT = 1
Counter	Timeout period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock frequency 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	IWDRCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.IWDTSLCSTP	IWDCSTPR.SLCSTP

## 25.4 Usage Notes

### 25.4.1 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.

### 25.4.2 Clock Division Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLK)  $\geq 4 \times$  (the frequency of the count clock source after division).

## 26. USB 2.0 Host/Function Module (USBc)

### 26.1 Overview

This MCU incorporates a USB 2.0 host/function module.

The USB module is a USB controller that is equipped to operate as a host controller or function controller. The module supports full-speed and low-speed transfer as defined in Universal Serial Bus (USB) Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in USB Specification 2.0. It also supports Battery Charging Specification Revision 1.2.

The USB has buffer memory for data transfer, providing a maximum of 10 pipes. PIPE1 to PIPE9 can be assigned any endpoint number based on peripheral devices used for communication or based on the user system.

Table 26.1 shows the specifications of the USB.

**Table 26.1 USB Specifications**

Item	Specifications
Features	<ul style="list-style-type: none"> <li>• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated. Host controller, function controller, and On-The-Go (OTG) are supported (one channel)</li> <li>• The host controller and the function controller can be switched by software.</li> <li>• Self-power mode or bus power mode can be selected.</li> <li>• BC 1.2 (Battery Charging Specification Revision 1.2) is supported.</li> </ul> <hr/> <p>When the host controller is selected (only in products with 32 Kbyte or more ROM):</p> <ul style="list-style-type: none"> <li>• Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported</li> <li>• Automatic scheduling for SOF and packet transmissions</li> <li>• Programmable intervals for isochronous and interrupt transfers</li> </ul> <hr/> <p>When the function controller is selected:</p> <ul style="list-style-type: none"> <li>• Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are 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>
Pipe configuration	<ul style="list-style-type: none"> <li>• Buffer memory for USB communication is provided.</li> <li>• Up to 10 pipes can be selected (including the default control pipe).</li> <li>• PIPE1 to PIPE9 can be assigned any endpoint number.</li> </ul> <hr/> <p>Transfer conditions that can be set for each pipe:</p> <ul style="list-style-type: none"> <li>• PIPE0: Control transfer, 64-byte single buffer</li> <li>• PIPE1 and PIPE2: 64-byte double buffer can be specified for bulk transfer 256-byte double buffer for isochronous transfer</li> <li>• PIPE3 to PIPE5: Bulk transfer, 64-byte double buffer</li> <li>• PIPE6 to PIPE9: Interrupt transfer, 64-byte 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, 1) port has been read (DCLRM)</li> <li>• NAK setting function for response PID generated by end of transfer (SHTNAK)</li> <li>• On-chip pull-up and pull-down resistors of DP/DM</li> </ul>
Low power consumption function	Module stop state can be set.

Figure 26.1 shows a block diagram of the USB.

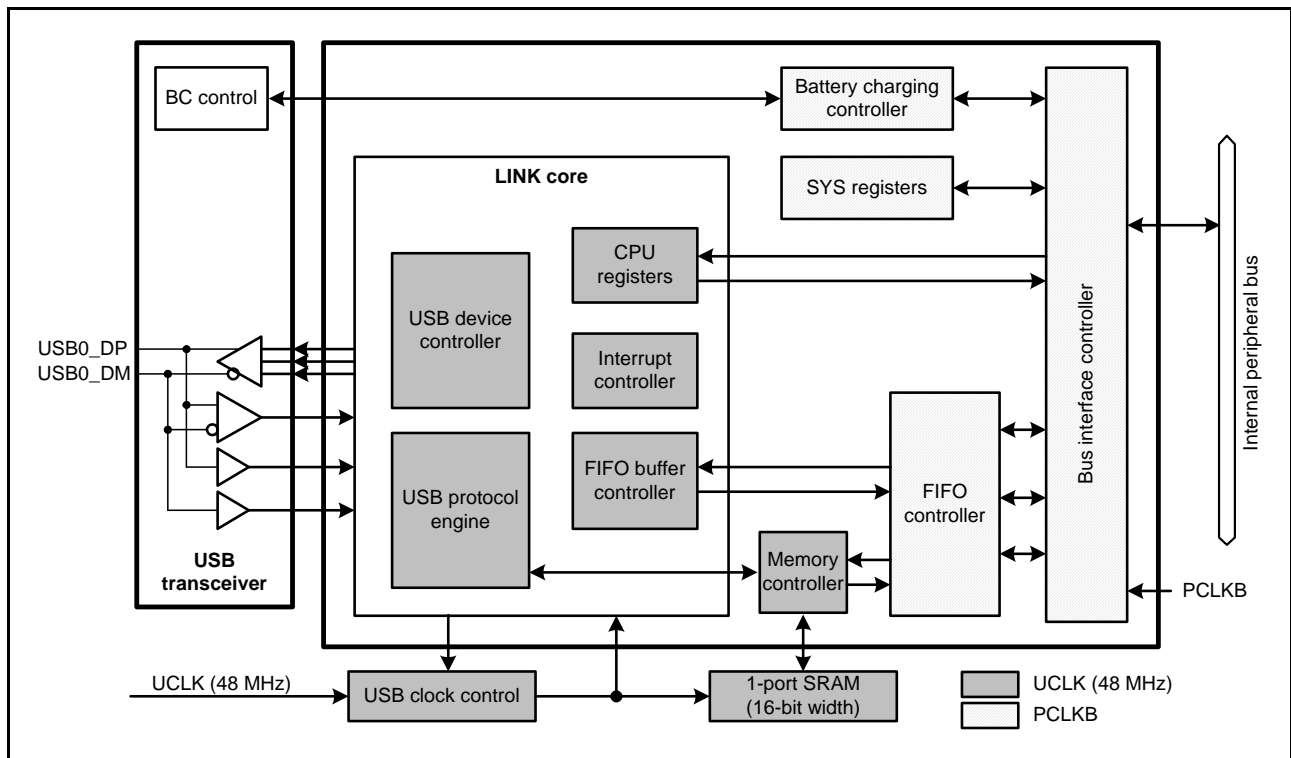


Figure 26.1 USB Block Diagram

Table 26.2 lists the I/O pins of the USB.

Table 26.2 USB Pin Configuration

Port	Pin Name	I/O	Function
USB	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. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.*1
	USB0_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB0_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB0_OVRCURA USB0_OVRCURB	Input	External overcurrent detection signals should be connected to these pins. VBUS comparator signals should be connected to these pins when the OTG power supply chip is connected.
	USB0_ID	Input	miniAB connector ID input signal should be connected to this pin during operation in OTG mode.
Common	VCC_USB	Input	USB power supply pin
	VSS_USB	Input	USB ground pin

Note 1. P16 is 5 V tolerant. When using non-5 V tolerant PC4, lower VBUS to 3.3 V and connect to the pin.

## 26.2 Register Descriptions

### 26.2.1 System Configuration Control Register (SYSCFG)

Address(es): 000A 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKE	—	CNEN	—	DCFM	DRPD	DPRPU	DMRPU	—	—	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 Operation Enable	0: USB operation is disabled. 1: USB operation is enabled.	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	DMRPU	D– Line Resistor Control*1	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b4	DPRPU	D+ Line Resistor Control*1	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b5	DRPD	D+/D– Line Resistor Control	0: Pulling down the lines is disabled. 1: Pulling down the lines is enabled.	R/W
b6	DCFM	Controller Function Select	0: Function controller is selected. 1: Host controller is selected.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	CNEN	CNEN Single End Receiver Enable	0: Single end receiver operation is disabled. 1: Single end receiver operation is enabled.	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	SCKE	USB Clock Enable*2	0: Stops supplying the clock signal to the USB. 1: Enables supplying the clock signal to the USB.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not enable the DMRPU and DPRPU bits at the same time.

Note 2. After writing 1 to the SCKE bit, read it and confirm it is set to 1.

#### USBE Bit (USB Operation Enable)

The USBE bit enables or disables operation of the USB.

Modifying the USBE bit from 1 to 0 initializes the register bits listed in Table 26.3.

This bit should be modified while the SCKE bit is 1.

When the host controller is selected, this bit should be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] bit chattering, and confirming that the USB bus state is stabilized.

**Table 26.3 Registers Initialized by Writing 0 to the SYSCFG.USBE Bit**

Selected Function	Register	Bit	Remarks
Function controller	SYSSTS0	LNST[1:0]	The value is retained when the host controller is selected.
	DVSTCTR0	RHST[2:0]	
	INTSTS0	DVSQ[2:0]	The value is retained when the host controller is selected.
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	The value is retained when the host controller is selected.
	USBVAL	WVALUE[15:0]	The value is retained when the host controller is selected.
	USBINDX	WINDEX[15:0]	The value is retained when the host controller is selected.
	USBLENG	WLENGTH[15:0]	The value is retained when the host controller is selected.
Host controller	DVSTCTR0	RHST[2:0]	
	FRMNUM	FRNM[10:0]	The value is retained when the function controller is selected.



**DMRPU Bit (D– Line Resistor Control)**

The DMRPU bit enables or disables pulling up the D– line when the function controller is selected.

When the DMRPU bit is set to 1 while the function controller is selected, the bit forces a pull-up of the D– line to notify the USB host of connection as a low-speed device. Modifying the DMRPU bit from 1 to 0 allows the USB to release the D– line, thus notifying the USB host of disconnection.

This bit should be set to 1 if the function controller is selected, and should be set to 0 if the host controller is selected.

**DPRPU Bit (D+ Line Resistor Control)**

The DPRPU bit enables or disables pulling up the D+ line when the function controller is selected.

When the DPRPU bit is set to 1 while the function controller is selected, the bit forces a pull-up of the D+ line to notify the USB host of connection as a full-speed device. Modifying the DPRPU bit from 1 to 0 allows the USB to release the D+ line, thus notifying the USB host of disconnection.

This bit should be set to 1 if the function controller is selected, and should be set to 0 if the host controller is selected.

**DRPD Bit (D+/D– Line Resistor Control)**

The DRPD bit enables or disables pulling down D+ and D– lines when the host controller is selected.

This bit should be set to 1 if the host controller is selected, and should be set to 0 if the function controller is selected.

**DCFM Bit (Controller Function Select)**

The DCFM bit selects the function of the USB.

This bit should be modified when the DMRPU, DPRPU, and DRPD bits are all 0.

**CNEN Bit (CNEN Single End Receiver Enable)**

Setting the CNEN bit to 1 allows the USB module to enable the single end receiver and set the LNST bit to monitor the status of the D+ and D– lines.

The CNEN bit is used when the USB module operates as a portable device for battery charging.

**SCKE Bit (USB Clock Enable)**

The SCKE bit stops or enables supplying 48-MHz clock signals to the USB.

When this bit is 0, only SYSCFG can be read from and written to; the other registers related to the USB cannot be read from or written to.

## 26.2.2 System Configuration Status Register 0 (SYSSTS0)

Address(es): 000A 0004h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	—	—	—	IDMON	LNST[1:0]		
Value after reset:		0 <sup>*1</sup>	0 <sup>*1</sup>	0	0	0	0	0	0	0	0	0	0	0 <sup>*1</sup>	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor	See Table 26.4.	R
b2	IDMON	External ID0 Input Pin Monitor	0: USB0_ID pin is low 1: USB0_ID pin is high	R
b5 to b3	—	Reserved	These bits are read as 0 and cannot be modified.	R
b6	HTACT	USB Host Sequencer Status Monitor	0: Host sequencer of the USB is completely stopped. 1: Host sequencer of the USB is not completely stopped.	R
b13 to b7	—	Reserved	These bits are read as 0 and cannot be modified.	R
b15, b14	OVCMON[1:0]	External USB0_OVRCURA/ USB0_OVRCURB Input Pin Monitor	The OVCMON[1] bit indicates the status of the USB0_OVRCURA pin. The OVCMON[0] bit indicates the status of the USB0_OVRCURB pin.	R

Note 1. Depends on the status of the USB0\_OVRCURA/USB0\_OVRCURB and USB0\_ID pins.

### LNST[1:0] Bits (USB Data Line Status Monitor)

The LNST[1:0] bits indicate the state of the USB data lines (D+ and D– lines). See Table 26.4.

The LNST[1:0] bits should be read after the connection processing (SYSCFG.DPRPU bit = 1) when the function controller is selected; whereas after enabling pull-down of the lines (SYSCFG.DRPD bit = 1) when the host controller is selected.

### HTACT Bit (USB Host Sequencer Status Monitor)

The HTACT bit is 0 when the host sequencer of the USB is completely stopped.

### OVCMON[1:0] Bits (External USB0\_OVRCURA/ USB0\_OVRCURB Input Pin Monitor)

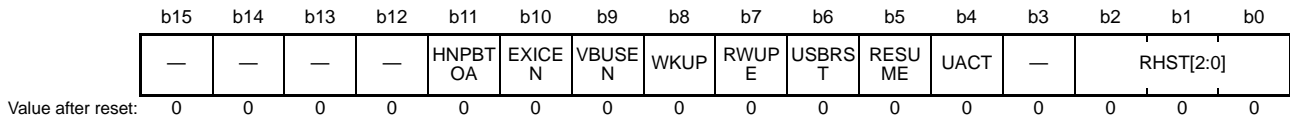
The OCVMON[1:0] bits indicate the status of overcurrent from an external power supply chip.

**Table 26.4 Status of USB Data Bus Lines (D+ Line, D– Line)**

LNST[1:0] Bits	During Low-Speed Operation	During Full-Speed Operation
00b	SE0	SE0
01b	K-State	J-State
10b	J-State	K-State
11b	SE1	SE1

### 26.2.3 Device State Control Register 0 (DVSTCTR0)

Address(es): 000A 0008h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status	<ul style="list-style-type: none"> <li>• When the host controller is selected                             <ul style="list-style-type: none"> <li>b2 b0 0 0 0: Communication speed not determined (powered state or no connection)</li> <li>1 x x: USB bus reset in progress</li> <li>0 0 1: Low-speed connection</li> <li>0 1 0: Full-speed connection</li> </ul> </li> <li>• When the function controller is selected                             <ul style="list-style-type: none"> <li>b2 b0 0 0 0: Communication speed not determined</li> <li>0 0 1: USB bus reset in progress or low-speed connection</li> <li>0 1 0: USB bus reset in progress or full-speed connection</li> </ul> </li> </ul>	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	UACT	USB Bus Enable	0: Downstream port is disabled (SOF transmission is disabled). 1: Downstream port is enabled (SOF transmission is enabled).	R/W
b5	RESUME	Resume Output	0: Resume signal is not output. 1: Resume signal is output.	R/W
b6	USBRST	USB Bus Reset Output	0: USB bus reset signal is not output. 1: USB bus reset signal is output.	R/W
b7	RWUPE	Wakeup Detection Enable	0: Downstream port wakeup is disabled. 1: Downstream port wakeup is enabled.	R/W
b8	WKUP	Wakeup Output	0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.	R/W
b9	VBUSEN	USB0_VBUSEN Output Pin Control	0: External USB0_VBUSEN pin outputs low. 1: External USB0_VBUSEN pin outputs high.	R/W
b10	EXICEN	USB0_EXICEN Output Pin Control	0: External USB0_EXICEN pin outputs low. 1: External USB0_EXICEN pin outputs high.	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	This bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

**RHST[2:0] Bits (USB Bus Reset Status)**

The RHST[2:0] bits indicate the status of the USB bus reset.

When the host controller is selected, the RHST[2:0] bits indicate 100b after the USBRST bit has been set to 1 by software.

The USB fixes the value of the RHST[2:0] bits when 0 is written to the USBRST bit by software and the USB completes SE0 driving.

When the function controller is selected, the RHST[2:0] bits indicate 010b (connection while DPRPU = 1) or 001b (disconnection while DMRPU = 1) when the USB detects the USB bus reset, and a DVST interrupt is generated.

**UACT Bit (USB Bus Enable)**

The UACT bit enables operation of the USB bus (controls the SOF packet transmission to the USB bus) when the host controller is selected.

With this bit set to 1, the USB puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception.

This module starts outputting SOF packets within one frame after 1 has been written to the UACT bit by software.

With this bit set to 0, the USB enters the idle state after outputting SOF packets.

The USB sets the UACT bit to 0 on any of the following conditions.

- A DTCH interrupt is detected during communication (while UACT = 1).
- An EOFERR interrupt is detected during communication (while UACT = 1).

Writing 1 to this bit should be done at the end of the USB reset processing (writing 0 to the USBRST bit) or at the end of the resume processing from the suspended state (writing 0 to the RESUME bit).

This bit should be set to 0 if the function controller is selected.

**RESUME Bit (Resume Output)**

The RESUME bit controls the resume signal output when the host controller is selected.

Setting the RESUME bit to 1 allows the USB to drive the port to the K-state and output the resume signal.

The USB sets the RESUME bit to 1 on detecting the remote wakeup signal while RWUPE is 1 in the USB suspended state.

The USB continues outputting K-state while the RESUME bit = 1 (until the RESUME bit is set to 0 by software). The RESUME bit should be 1 (= resume period) for the time defined by USB Specification 2.0.

This bit should be set to 1 in the suspended state.

Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).

This bit should be set to 0 if the function controller is selected.

**USBRST Bit (USB Bus Reset Output)**

The USBRST bit controls the USB bus reset signal output when the host controller is selected.

When the host controller is selected, setting this bit to 1 allows the USB to drive SE0 of the USB port to reset the USB bus.

The USB continues outputting SE0 while USBRST = 1 (until the USBRST bit is set to 1 by software). The USBRST bit should be 1 (= USB bus reset period) for the time defined by USB Specification 2.0.

Writing 1 to this bit during communication (the UACT bit = 1) or during the resume processing (the RESUME bit = 1) prevents the USB from starting the USB bus reset processing until both the UACT and RESUME bits become 0.

Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).

This bit should be set to 0 if the function controller is selected.

**RWUPE Bit (Wakeup Detection Enable)**

The RWUPE bit enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller is selected.

With this bit set to 1, on detecting the remote wakeup signal, the USB detects the resume signal (K-state for 2.5  $\mu$ s) from the downstream port device and performs the resume processing (drives the port to the K-state).

With this bit set to 0, the USB ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port.

While the RWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SYSCFG.SCKE bit should be set to 1).

This bit should be set to 0 if the function controller is selected.

**WKUP Bit (Wakeup Output)**

The WKUP bit enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller is selected.

The USB controls the output time of a remote wakeup signal. When this bit is set to 1, the USB sets this bit to 0 after outputting the 10-ms K-state.

According to USB Specification 2.0, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USB 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 (INTSTS0.DVSQ[2:0] bits = 1xxb) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while the SYSCFG.SCKE bit = 1).

This bit should be set to 0 if the host controller is selected.

**HNPBTOA Bit (Host Negotiation Protocol (HNP) Control)**

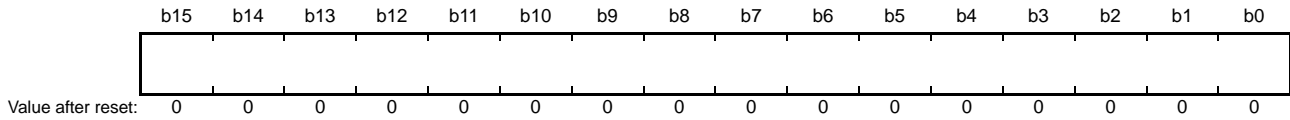
The HNPBTOA bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though the SYSCFG.DPRPU bit = 0 or SYSCFG.DCFM = 1 is set. Even if the falling edge of the D+ signal is detected at this time, no resume (RESM) interrupt is generated.

After this bit is set to 1, write 0 to this bit by software to terminate the HNP processing when connection to the host (pull-up on the target side) or timeout of the HNP processing is detected.

26.2.4 CFIFO Port Register (CFIFO)  
 D0FIFO Port Register (D0FIFO)  
 D1FIFO Port Register (D1FIFO)

(1) When the MBW bit is 1

Address(es): CFIFO 000A 0014h, D0FIFO 000A 0018h, D1FIFO 000A 001Ch



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

(2) When the MBW bit is 0

Address(es): CFIFO 000A 0014h, D0FIFO 000A 0018h, D1FIFO 000A 001Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	L[7:0]	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

**FIFO Port Bit**

Accessing the FIFO port 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 FIFO 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 bit of the FIFO port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL).

When the MBW bit is 1 (16-bit width), the data arrangement may differ from the data arrangement on the RAM depending on the value of the MDE.MDE[2:0] bits and the setting of the BIGEND bit (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, or D1FIFOSEL.BIGEND). Table 26.5 lists the endian operation in 16-bit access. Note that if the total number of transmit data bytes is odd, access the L[7:0] bits in bytes when writing the last data.

When the MBW bit is 0 (8-bit width), access the L[7:0] bits in bytes.

**Table 26.5 Endian Operation in 16-Bit Access**

<b>MDE.MDE[2:0] bits</b>	<b>CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit</b>	<b>Bits 15 to 8</b>	<b>Bits 7 to 0</b>	<b>Remarks</b>
000b (big endian)	0 (little endian)	Data in address N + 1	Data in address N	Bytes reversed
	1 (big endian)	Data in address N	Data in address N + 1	
111b (little endian)	0 (little endian)	Data in address N + 1	Data in address N	
	1 (big endian)	Data in address N	Data in address N + 1	Bytes reversed

## 26.2.5 CFIFO Port Select Register (CFIFOSEL) D0FIFO Port Select Register (D0FIFOSEL) D1FIFO Port Select Register (D1FIFOSEL)

- CFIFOSEL

Address(es): 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 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 Settings other than above are 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 <sup>1</sup>
b15	RCNT	Read Count Mode	0: The DTLN[8:0] bits (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) are cleared when all of the receive data has been read from the CFIFO. (In double buffer mode, the DTLN[8:0] 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. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DTC 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] 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.

**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): D0FIFOSEL 000A 0028h, D1FIFOSEL 000A 002Ch



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE [3:0]	FIFO Port Access Pipe Specification	b3 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 Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
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	DTC Transfer Request Enable	0: DTC transfer request is disabled. 1: DTC 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[8:0] bits (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) 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. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DTC 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 (DTC Transfer Request Enable)**

The DREQE bit enables or disables the DTC transfer request to be issued.

Before setting the DREQE bit to 1 to enable the DTC 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 using the selected pipe.

With this bit set to 1, the USB 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 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.

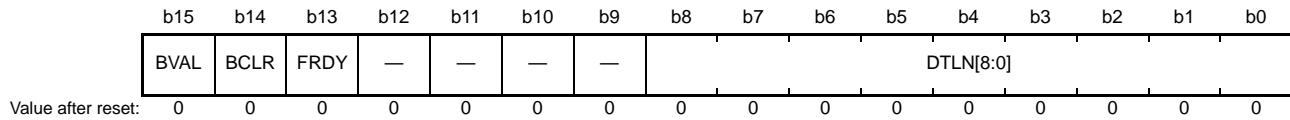
**RCNT Bit (Read Count Mode)**

The RCNT bit specifies the read mode for the value in the CFIFOCTR.DTLN bit.

When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

## 26.2.6 CFIFO Port Control Register (CFIFOCTR) D0FIFO Port Control Register (D0FIFOCTR) D1FIFO Port Control Register (D1FIFOCTR)

Address(es): CFIFOCTR 000A 0022h, D0FIFOCTR 000A 002Ah, D1FIFOCTR 000A 002Eh



Bit	Symbol	Bit Name	Description	R/W
b8 to b0	DTLN[8:0]	Receive Data Length	Indicate 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: Does not operate. 1: Clears the buffer memory on the CPU side.	R/W <sup>1</sup>
b15	BVAL	Buffer Memory Valid Flag	0: Invalid 1: Writing ended	R/W

Note 1. Only 0 can be read.

CFIFOCTR, D0FIFOCTR, and D1FIFOCTR correspond to CFIFO, D0FIFO, and D1FIFO, respectively.

### 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 bit (n = 0, 1) value as described below.

- RCNT = 0  
The USB sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU or DTC has read all the received data from a single FIFO buffer plane.  
While the PIPECFG.BFRE bit = 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 decrements the value indicated by the DTLN[8:0] 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 sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.

**FRDY Bit (FIFO Port Ready)**

The FRDY bit indicates whether the FIFO port can be accessed by the CPU or DTC.

In the following cases, the USB 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 the PIPECFG.BFRE bit = 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 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 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 flag and the BCLR bit simultaneously, the USB 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).

**BVAL Flag (Buffer Memory Valid Flag)**

The BVAL flag 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 flag to 1 in the following cases. Then, the USB switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

- To transmit a short packet, set the BVAL flag to 1 after data has been written.
- To transmit a zero-length packet, set the BVAL flag 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 sets the BVAL flag to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

Writing 1 to the BVAL flag should be done while the FRDY bit is 1 (set by the USB).

When the selected pipe is in the receiving direction, do not set the BVAL flag to 1.

## 26.2.7 Interrupt Enable Register 0 (INTENB0)

Address(es): 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*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DVSE	Device State Transition Interrupt Enable*1	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*1	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

Note 1. The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller is selected; do not set these bits to 1 to enable the corresponding interrupt output when the host controller is selected.

On detecting the interrupt corresponding to the bit in INTENB0 to which 1 has been set by software, the USB generates the USB interrupt request.

The USB 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 generates the USB interrupt request when the corresponding interrupt enable bit in INTENB0 is modified from 0 to 1 by software.

## 26.2.8 Interrupt Enable Register 1 (INTENB1)

Address(es): 000A 0032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCRE	BCHGE	—	DTCHE	ATTCH E	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	PDDETINTE0
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	PDDETINTE0	PDDETINT0 Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACKE	Setup Transaction Normal Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	SIGNE	Setup Transaction Error Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	EOFERRE	EOF Error Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCH E	Connection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DTCHE	Disconnection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHGE	USB Bus Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	OVRCRE	Overcurrent Input Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note: The bits in INTENB1 can be set to 1 only when the host controller is selected; do not set these bits to 1 to enable the corresponding interrupt output when the function controller is selected.

INTENB1 specifies the interrupt masks when the host controller is selected, and for the setup transaction.

On detecting the interrupt corresponding to the bit in INTENB1 to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in INTSTS1 when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in INTENB1 (regardless of whether the interrupt output is enabled or disabled).

While the status bit in INTSTS1 corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in INTENB1 is modified from 0 to 1 by software.

When the function controller is selected, the interrupts should not be enabled.

### 26.2.9 BRDY Interrupt Enable Register (BRDYENB)

Address(es): 000A 0036h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	PIPE9B RDYE	PIPE8B RDYE	PIPE7B RDYE	PIPE6B RDYE	PIPE5B RDYE	PIPE4B RDYE	PIPE3B RDYE	PIPE2B RDYE	PIPE1B RDYE	PIPE0B RDYE
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	PIPE0BRDYE	BRDY Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BRDYE	BRDY Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BRDYE	BRDY Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BRDYE	BRDY Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BRDYE	BRDY Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BRDYE	BRDY Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BRDYE	BRDY Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BRDYE	BRDY Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BRDYE	BRDY Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BRDYE	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

BRDYENB enables or disables the INTSTS0.BRDY bit to be set to 1 when the BRDY interrupt is detected for each pipe.

On detecting the BRDY interrupt for the pipe corresponding to the bit in BRDYENB to which 1 has been set by software, the USB sets 1 to the corresponding BRDYSTS.PIPEnBRDY bit (n = 0 to 9) and the INTSTS0.BRDY bit. If INTENB0.BRDYE = 1 at this time, the USB generates the BRDY interrupt request.

While at least one PIPEnBRDY bit indicates 1, the USB generates the BRDY interrupt request when the corresponding interrupt enable bit in BRDYENB is modified from 0 to 1 by software.



## 26.2.10 NRDY Interrupt Enable Register (NRDYENB)

Address(es): 000A 0038h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
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	PIPE0NRDYE	NRDY Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1NRDYE	NRDY Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2NRDYE	NRDY Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3NRDYE	NRDY Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4NRDYE	NRDY Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5NRDYE	NRDY Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6NRDYE	NRDY Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7NRDYE	NRDY Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8NRDYE	NRDY Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9NRDYE	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

NRDYENB enables or disables the INTSTS0.NRDY bit to be set to 1 when the NRDY interrupt is detected for each pipe.

On detecting the NRDY interrupt for the pipe corresponding to the bit in NRDYENB to which 1 has been set by software, the USB sets 1 to the corresponding NRDYSTS.PIPE<sub>n</sub>NRDYE bit (n = 0 to 9) and the INTSTS0.NRDY bit. If INTENB0.NRDYE = 1 at this time, the USB generates the NRDY interrupt request.

While at least one PIPE<sub>n</sub>NRDYE bit indicates 1, the USB generates the NRDY interrupt request when the corresponding interrupt enable bit in NRDYENB is modified from 0 to 1 by software.

## 26.2.11 BEMP Interrupt Enable Register (BEMPENB)

Address(es): 000A 003Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9B EMPE	PIPE8B EMPE	PIPE7B EMPE	PIPE6B EMPE	PIPE5B EMPE	PIPE4B EMPE	PIPE3B EMPE	PIPE2B EMPE	PIPE1B EMPE	PIPE0B EMPE
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	PIPE0BEMPE	BEMP Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BEMPE	BEMP Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BEMPE	BEMP Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BEMPE	BEMP Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BEMPE	BEMP Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BEMPE	BEMP Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BEMPE	BEMP Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BEMPE	BEMP Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BEMPE	BEMP Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BEMPE	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

BEMPENB enables or disables the INTSTS0.BEMP bit to be set to 1 when the BEMP interrupt is detected for each pipe. On detecting the BEMP interrupt for the pipe corresponding to the bit in BEMPENB to which 1 has been set by software, the USB sets 1 to the corresponding BEMPSTS.PIPEnBEMP bit (n = 0 to 9) and the INTSTS0.BEMP bit. If INTENB0.BEMPE = 1 at this time, the USB generates the BEMP interrupt request.

While at least one PIPEnBEMP bit in BEMPSTS indicates 1, the USB generates the BEMP interrupt request when the corresponding interrupt enable bit in BEMPENB is modified from 0 to 1 by software.

## 26.2.12 SOF Output Configuration Register (SOFCFG)

Address(es): 000A 003Ch

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	EDGESTS	—	—	—	—
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
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	EDGESTS	Edge Interrupt Output Status Monitor* <sup>1</sup>	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 clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select* <sup>1</sup>	0: For non-low-speed communication 1: For low-speed communication	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Confirm that this bit is 0 before stopping the clock supply to the USB module.

### 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. Confirm that this bit is 0 before stopping the clock supply to the USB.

### BRDYM Bit (BRDY Interrupt Status Clear Timing)

The BRDYM bit specifies the timing for clearing the BRDY interrupt status for each pipe.

### TRNENSEL Bit (Transaction-Enabled Time Select)

The TRNENSEL bit selects, for full-speed or low-speed communication, the transaction-enabled time in which the USB issues tokens in a frame via the port.

Set the TRNENSEL bit to 1 when a low-speed device is connected.

The TRNENSEL bit is valid only when the host controller is selected.

This bit should be set to 0 if the function controller is selected.

## 26.2.13 Interrupt Status Register 0 (INTSTS0)

Address(es): 000A 0040h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
VBINT	RESM	SOFR	DVST	CTRTR	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]		VALID	CTSQ[2:0]			
Value after reset: 0 0 0 0/1*1 0 0 0 0 0*2 0*3 0*3 0/1*3 0 0 0 0															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage	b2 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	R
b3	VALID	USB Request Reception	0: Setup packet is not received. 1: Setup packet is received.	R/W
b6 to b4	DVSQ[2:0]	Device State	b6 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	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	CTRTR	Control Transfer Stage Transition Interrupt Status*5	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*5	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.	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*6	0: VBUS interrupts are not generated. 1: VBUS interrupts are generated.	R/W*4

x: Don't care

Note 1. The value is 0 when the MCU is reset and 1 after a USB bus reset.

Note 2. The value is 1 when the USB0\_VBUS pin is high and 0 when the USB0\_VBUS pin is low.

Note 3. The value is 000b when the MCU is reset and 001b after a USB bus reset.

Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRTR, 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. The status of the RESM, DVST, and CTRTR bits are changed only when the function controller is selected. Set the corresponding interrupt enable bits to 0 (disabled) when the host controller is selected.

Note 6. A change in the status indicated by the VBINT and RESM bits can be detected even while the clock supply is stopped (the SCKE bit = 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.

**CTSQ[2:0] Bits (Control Transfer Stage)**

When the host controller is selected, the read value is invalid.

**VALID Bit (USB Request Reception)**

When the host controller is selected, the read value is invalid.

**DVSQ[2:0] Bits (Device State)**

The DVSQ[2:0] bits are initialized by a USB bus reset.

When the host controller is selected, the read value is invalid.

**BRDY Bit (Buffer Ready Interrupt Status)**

Indicates the BRDY interrupt status.

The USB sets the BRDY bit to 1 when at least one PIPE<sub>n</sub>BRDY bit (n = 0 to 9) is set to 1 among the PIPEBRDY bits. These bits correspond to the BRDYENB.PIPE<sub>n</sub>BRDYE bits (n = 0 to 9) to which 1 has been set, when the USB detects the BRDY interrupt status in at least one pipe among the pipes for which the BRDY interrupt output is enabled by software.

For the conditions for PIPE<sub>n</sub>BRDY status assertion, refer to section 26.3.3.1, BRDY Interrupt.

The USB sets the BRDY bit to 0 when 0 is written by software to all the PIPE<sub>n</sub>BRDY bits corresponding to the PIPE<sub>n</sub>BRDYE bits that have been set to 1.

The BRDY bit cannot be set to 0 even if 0 is written to this bit by software.

**NRDY Bit (Buffer Not Ready Interrupt Status)**

The USB sets the NRDY bit to 1 when at least one PIPE<sub>n</sub>NRDY bit (n = 0 to 9) is set to 1 among the PIPE<sub>n</sub>NRDY bits corresponding to the PIPE<sub>n</sub>NRDYE bits (n = 0 to 9) to which 1 has been set (when the USB detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output).

For the conditions for PIPE<sub>n</sub>NRDY status assertion, refer to section 26.3.3.2, NRDY Interrupt.

The USB sets the NRDY bit to 0 when 0 is written by software to all the PIPE<sub>n</sub>NRDY bits corresponding to the PIPE<sub>n</sub>NRDYE bits that have been set to 1.

The NRDY bit cannot be set to 0 even if 0 is written to this bit by software.

**BEMP Bit (Buffer Empty Interrupt Status)**

The USB sets the BEMP bit to 1 when at least one PIPE<sub>n</sub>BEMP bit (n = 0 to 9) is set to 1 among the PIPE<sub>n</sub>BEMP bits corresponding to the PIPE<sub>n</sub>BEMPE bits (n = 0 to 9) to which 1 has been set (when the USB detects the BEMP interrupt status in at least one pipe among the pipes for which the BEMP interrupt output is enabled by software).

For the conditions for PIPE<sub>n</sub>BEMP status assertion, refer to section 26.3.3.3, BEMP Interrupt.

The USB sets the BEMP bit to 0 when 0 is written by software to all the PIPE<sub>n</sub>BEMP bits corresponding to the PIPE<sub>n</sub>BEMPE bits that have been set to 1.

The BEMP bit cannot be set to 0 even if 0 is written to this bit by software.

**CTRT Bit (Control Transfer Stage Transition Interrupt Status)**

When the function controller is selected, the USB updates the value of the CTSQ[2:0] bits and sets the CTRT 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 detects the next control transfer stage transition.

When the host controller is selected, the read value is invalid.

**DVST Bit (Device State Transition Interrupt Status)**

When the function controller is selected, the USB updates the DVSQ[2:0] value 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 detects the next device state transition.

When the host controller is selected, the read value is invalid.

**SOFR Bit (Frame Number Refresh Interrupt Status)**

(1) When the host controller is selected

The USB sets the SOFR bit to 1 on updating the frame number when the DVSTCTR0.UACT bit has been set to 1 by software. (A frame number refresh interrupt is detected every 1 ms.)

(2) When the function controller is selected

The USB sets the SOFR bit to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.)

The USB 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)**

When the function controller is selected, the USB 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] = 1xxb).

When the host controller is selected, the read value is invalid.

**VBINT Bit (VBUS Interrupt Status)**

The USB 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 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.

## 26.2.14 Interrupt Status Register 1 (INTSTS1)

Address(es): 000A 0042h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVR CR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFER R	SIGN	SACK	—	—	—	PDD ET INT0
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	PDD ET INT0	PDD ET0 Detection Interrupt Status	0: PDD ET0 detection interrupts are not generated. 1: PDD ET0 detection interrupts are generated.	R/W *1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status	0: SACK interrupts are not generated. 1: SACK interrupts are generated.	R/W *1
b5	SIGN	Setup Transaction Error Interrupt Status	0: SIGN interrupts are not generated. 1: SIGN interrupts are generated.	R/W *1
b6	EOFERR	EOF Error Detection Interrupt Status	0: EOFERR interrupts are not generated. 1: EOFERR interrupts are generated.	R/W *1
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCH	ATTCH Interrupt Status	0: ATTCH interrupts are not generated. 1: ATTCH interrupts are generated.	R/W *1
b12	DTCH	USB Disconnection Detection Interrupt Status	0: DTCH interrupts are not generated. 1: DTCH interrupts are generated.	R/W *1
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHG	USB Bus Change Interrupt Status*2	0: BCHG interrupts are not generated. 1: BCHG interrupts are generated.	R/W *1
b15	OVR CR	Overcurrent Input Change Interrupt Status*2	0: OVR CR interrupts are not generated. 1: OVR CR interrupts are generated.	R/W *1

Note 1. To clear the status indicated by the bits in INTSTS1, write 0 only to the bits to be cleared; write 1 to the other bits.

Note 2. A change in the status indicated by the OVRCR or BCHG bit can be detected even while the clock supply is stopped (while the SYSCFG.SCKE bit = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after setting the SYSCFG.SCKE bit to 1.

No interrupts other than those indicated by the BCHG and OVRCR bits can be detected while the clock supply is stopped (while the SYSCFG.SCKE bit = 0).

INTSTS1 is used to confirm the status of each interrupt when the host controller is selected.

The various status change interrupts indicated by the bits in INTSTS1 should be enabled only when the host controller is selected.

### PDD ET INT0 Bit (PDD ET0 Detection Interrupt Status)

Indicates the status of the portable device detection interrupt when the host controller is selected.

This bit is set to 1 when the USB module detects when a level change (high to low or low to high) occurs in the input value to the VDPDET pin of the USB physical layer transceiver (PHY). The USB module sets the PDD  
ET  
INT0 bit to indicate the VDPDET input value.

When the PDD  
ET  
INT0 interrupt is generated, use software to repeat reading the PDD  
ET  
INT0 bit until the same value is read three or more times, and eliminate chattering.

**SACK Bit (Setup Transaction Normal Response Interrupt Status)**

Indicates the status of the setup transaction normal response interrupt when the host controller is selected.

The USB detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by the USB, and sets the SACK bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SACK interrupt.

When the function controller is selected, the read value is invalid.

**SIGN Bit (Setup Transaction Error Interrupt Status)**

Indicates the status of the setup transaction error interrupt when the host controller is selected.

The USB detects the SIGN interrupt when ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by this module, and sets the SIGN bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SIGN interrupt.

Specifically, the USB detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions.

- Timeout is detected by the USB when the peripheral device has returned no response.
- A damaged ACK packet is received.
- A handshake other than ACK (NAK, NYET, or STALL) is received.

When the function controller is selected, the read value is invalid.

**EOFERR Bit (EOF Error Detection Interrupt Status)**

Indicates the status of the EOFERR interrupt when the host controller is selected.

The USB detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by USB Specification 2.0, and sets the EOFERR bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the EOFERR interrupt.

After detecting the EOFERR interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the USB port should be terminated by software and perform re-enumeration of the USB port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

**ATTCH Bit (ATTCH Interrupt Status)**

Indicates the status of the ATTCH interrupt when the host controller is selected.

The USB detects the ATTCH interrupt on detecting J-state or K-state of the full-speed or low-speed signal level for 2.5  $\mu$ s, and sets the ATTCH bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

Specifically, the USB detects the ATTCH interrupt on any of the following conditions.

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5  $\mu$ s.
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5  $\mu$ s.

When the function controller is selected, the read value is invalid.



**DTCH Bit (USB Disconnection Detection Interrupt Status)**

Indicates the status of the USB disconnection detection interrupt when the host controller is selected.

The USB detects the DTCH interrupt on detecting USB bus disconnection, and sets the DTCH bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

The USB detects bus disconnection based on USB Specification 2.0.

After detecting the DTCH interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All the pipes in which communications are currently carried out for the USB port should be terminated by software and make a transition to the wait state for bus connection to the USB port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

**BCHG Bit (USB Bus Change Interrupt Status)**

Indicates the status of the USB bus change interrupt.

The USB detects the BCHG interrupt when a change in the full-speed or low-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets the BCHG bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

The USB sets the LNST[1:0] bits to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST[1:0] bits until the same value is read three or more times, and eliminate chattering.

A change in the USB bus state can be detected even while the internal clock supply is stopped.

When the function controller is selected, the read value is invalid.

**OVRCCR Bit (Overcurrent Input Change Interrupt Status)**

Indicates the status of the USB0\_OVRCURA and USB0\_OVRCURB input pin change interrupt.

The USB detects the OVRCCR interrupt when a change (high to low or low to high) occurs in at least one of the input values to the USB0\_OVRCURA and USB0\_OVRCURB pins, and sets the OVRCCR bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

## 26.2.15 BRDY Interrupt Status Register (BRDYSTS)

Address(es): 000A 0046h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9B RDY	PIPE8B RDY	PIPE7B RDY	PIPE6B RDY	PIPE5B RDY	PIPE4B RDY	PIPE3B RDY	PIPE2B RDY	PIPE1B RDY	PIPE0B RDY
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.

## 26.2.16 NRDY Interrupt Status Register (NRDYSTS)

Address(es): 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.

## 26.2.17 BEMP Interrupt Status Register (BEMPSTS)

Address(es): 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.

### 26.2.18 Frame Number Register (FRMNUM)

Address(es): 000A 004Ch



Bit	Symbol	Bit Name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number	Latest 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. To clear the status, write 0 only to the bits to be cleared; write 1 to the other bits.

#### FRNM[10:0] Bits (Frame Number)

These bits indicate the latest frame number for the USB after the issuing of an SOF packet every 1 ms or writing to the FRNM[10:0] bits at the SOF packet reception.

The CRCE bit can be set to 0 by writing 0 to the CRCE bit by software.

#### CRCE Bit (Receive Data Error)

Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer.

The CRCE bit can be set to 0 by writing 0 to the CRCE bit by software. Here, 1 should be written to the other bits in FRMNUM.

On detecting a CRC error, the USB generates 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.

The OVRN bit can be set to 0 by writing 0 to the OVRN bit by software. Here, 1 should be written to the other bits in FRMNUM.

(1) When the host controller is selected

The USB sets the OVRN bit to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the time to issue an OUT token comes before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the time to issue an IN token comes when no FIFO buffer planes are empty.

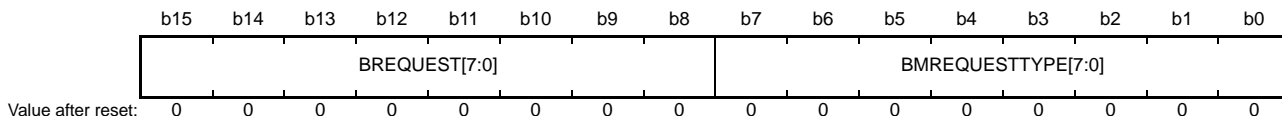
(2) When the function controller is selected

The USB 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.

### 26.2.19 USB Request Type Register (USBREQ)

Address(es): 000A 0054h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	These bits store the USB request bmRequestType value.	R/W *1
b15 to b8	BREQUEST[7:0]	Request	These bits store the USB request bRequest value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

USBREQ stores setup requests for control transfers.

When the function controller is selected, the values of bRequest and bmRequestType that have been received are stored.

When the host controller is selected, the values of bRequest and bmRequestType to be transmitted are set.

USBREQ is initialized by a USB bus reset.

#### BMREQUESTTYPE[7:0] Bits (Request Type)

These bits hold the value of the bmRequestType field of a USB request.

- When the host controller is selected:  
Set these bits to the value of the USB request data in setup transactions for transmission. Do not overwrite the value of the BMREQUESTTYPE[7:0] bits while the DCPCTR.SUREQ bit = 1.
- When the function controller is selected:  
These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

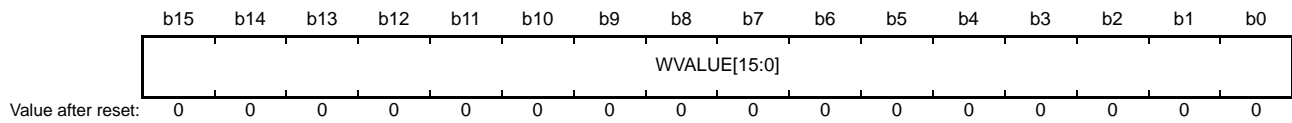
#### BREQUEST[7:0] Bits (Request)

These bits store bRequest value of the USB request.

- When the host controller is selected:  
Set these bits to the value of the USB request data in setup transactions for transmission. Do not overwrite the value of the BREQUEST[7:0] bits while the DCPCTR.SUREQ bit = 1.
- When the function controller is selected:  
These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

## 26.2.20 USB Request Value Register (USBVAL)

Address(es): 000A 0056h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WVALUE[15:0]	Value	These bits store the USB request wValue value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

When the function controller is selected, the value of wValue that has been received is stored in USBVAL. When the host controller is selected, the value of wValue to be transmitted is set.

USBVAL is initialized by a USB bus reset.

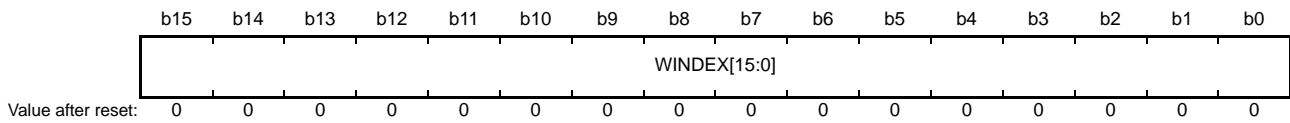
### WVALUE[15:0] Bits (Value)

These bits store wRequest value of the USB request.

- When the host controller is selected:  
Set these bits to the value of the wValue field in USB requests of setup transactions for transmission. Do not overwrite the value of the WVALUE[15:0] bits while the DCPCTR.SUREQ bit = 1.
- When the function controller is selected:  
These bits indicate the value of the wValue field in USB requests received in setup transactions for reception. Writing to the WVALUE[15:0] bits has no effect.

### 26.2.21 USB Request Index Register (USBINDX)

Address(es): 000A 0058h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WINDEX[15:0]	Index	These bits store the USB request wIndex value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

USBINDX stores setup requests for control transfers.

When the function controller is selected, the value of wIndex that has been received is stored. When the host controller is selected, the value of wIndex to be transmitted is set.

USBINDX is initialized by a USB bus reset.

#### WINDEX[15:0] Bits (Index)

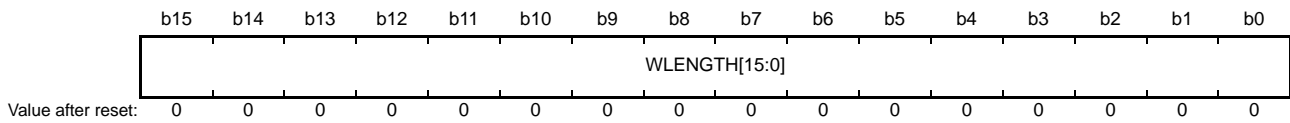
These bits hold the value of the wIndex field of a USB request.

- When the host controller is selected:  
Set these bits to the value of the wIndex field in USB requests of setup transactions for transmission. Do not overwrite the value of the WINDEX[15:0] bits while the DCPCTR.SUREQ bit = 1.
- When the function controller is selected:  
These bits indicate the value of the wIndex field in USB requests received in setup transactions for reception. Writing to the WINDEX[15:0] bits has no effect.



## 26.2.22 USB Request Length Register (USBLENG)

Address(es): 000A 005Ah



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WLENGTH[15:0]	Length	These bits store the USB request wLength value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

USBLENG stores setup requests for control transfers.

When the function controller is selected, the value of wLength that has been received is stored. When the host controller is selected, the value of wLength to be transmitted is set.

USBLENG is initialized by a USB bus reset.

### WLENGTH[15:0] Bits (Length)

These bits hold the value of the wLength field of a USB request.

- When the host controller is selected:  
Set these bits to the value of the wLength field in USB requests of setup transactions for transmission. Do not overwrite the value of the WLENGTH[15:0] bits while the DCPCTR.SUREQ bit = 1.
- When the function controller is selected:  
These bits indicate the value of the wLength field in USB requests received in setup transactions for reception. Writing to the WLENGTH[15:0] bits has no effect.

### 26.2.23 DCP Configuration Register (DCPCFG)

Address(es): 000A 005Ch

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	SHTNA K	—	—	DIR	—	—	—	—
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
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DIR	Transfer Direction* <sup>1</sup>	0: Data receiving direction 1: Data 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* <sup>1</sup>	0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify this bit while PID is NAK. Before modifying this bit after modifying the DCPCTR.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, checking the PBUSY bit through software is not necessary.

#### DIR Bit (Transfer Direction)

When the host controller is selected, the DIR bit sets the transfer direction of the data stage and status stage.

When the function controller is selected, the DIR bit should be set to 0.

#### 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.

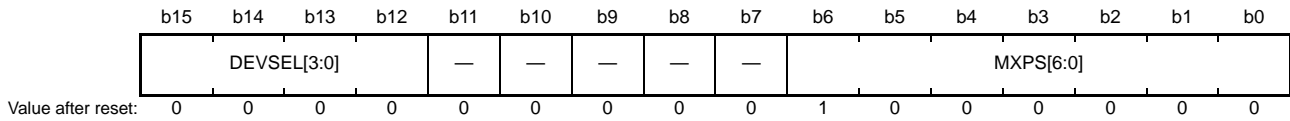
The SHTNAK bit is valid when the selected pipe is in the receiving direction.

When the SHTNAK bit is set to 1, the USB modifies the DCPCTR.PID[1:0] bits for the DCP to NAK on determining the end of the transfer. The USB determines that the transfer has ended on the following condition.

- A short packet (including a zero-length packet) is successfully received.

### 26.2.24 DCP Maximum Packet Size Register (DCPMAXP)

Address(es): 000A 005Eh



Bit	Symbol	Bit Name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size*1	These bits set the maximum amount of data (maximum packet size) in payloads for the DCP. b6                      b0 0 0 0 1 0 0 0: 8 bytes 0 0 1 0 0 0 0: 16 bytes 0 0 1 1 0 0 0: 24 bytes 0 1 0 0 0 0 0: 32 bytes 0 1 0 1 0 0 0: 40 bytes 0 1 1 0 0 0 0: 48 bytes 0 1 1 0 0 0 0: 56 bytes 1 0 0 0 0 0 0: 64 bytes 1 0 0 1 0 0 0: 72 bytes 1 0 1 0 0 0 0: 80 bytes 1 0 1 1 0 0 0: 88 bytes 1 1 0 0 0 0 0: 96 bytes 1 1 0 1 0 0 0: 104 bytes 1 1 1 0 0 0 0: 112 bytes 1 1 1 1 0 0 0: 120 bytes Settings other than above are prohibited.	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*2	b15                      b12 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. Modify the MXPS[6:0] bits while PID is NAK. Before modifying these bits after modifying the DCPCTR.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, checking the PBUSY bit through software is not necessary. After modifying the MXPS[6:0] bits and the DCP has been set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit the port control register to 1.

Note 2. Modify the DEVSEL[3:0] bits while PID is NAK and the DCPCTR.SUREQ bit is 0. To modify these bits after modifying the DCPCTR.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, checking the PBUSY bit through software is not necessary.

#### MXPS[6:0] Bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum amount of data (maximum packet size) in payloads for the DCP. The initial value of the bits is 40h (64 bytes).

Ensure that the setting of the MXPS[6:0] bits is in compliance with USB Specification 2.0.

Do not write to the FIFO buffer or set PID = BUF while the setting of the MXPS[6:0] bits is 0.

#### DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the address of the peripheral device which is the communication target during control transfer.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

## 26.2.25 DCP Control Register (DCPCTR)

Address(es): 000A 0060h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BSTS	SUREQ	—	—	SUREQ CLR	—	—	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	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	0: Invalid 1: Specifies DATA0.	R/W*1
b10, b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	SUREQCLR	SUREQ Bit Clear	0: Invalid 1: Clears the SUREQ bit to 0.	R/W
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	SUREQ	Setup Token Transmission	0: Invalid 1: Transmits the setup packet.	R/W
b15	BSTS	Buffer Status	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read.

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 PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

### PID[1:0] Bits (Response PID)

The PID[1:0] bits control the response type of the USB during control transfer.

(1) When the host controller is selected

Modify the setting of the PID[1:0] bits from NAK to BUF using the following procedure.

- When the transmitting direction is set  
Write all the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID is NAK, and then write 01b (BUF response). After PID has been set to BUF, the USB executes the OUT transaction.
- When the receiving direction is set  
Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, the USB executes the IN transaction.

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB sets PID to STALL (11b) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits has been set to BUF by software.
- The USB sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times.
- The USB also sets PID to STALL (11b) on receiving the STALL handshake.

(2) When the function controller is selected

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB modifies the PID[1:0] bits to NAK on receiving the setup packet. Here, the USB sets the INTSTS0.VALID bit to 1. The setting of the PID[1:0] bits cannot be modified until the VALID bit is set to 0 by software.
- The USB sets PID to STALL (11b) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits have been set to BUF by software.
- The USB sets PID to STALL (1xb) on detecting the control transfer sequence error.
- The USB sets PID to NAK on detecting the USB bus reset.

The USB does not check the setting of the PID[1:0] bits while the SET\_ADDRESS request is processed.

The PID[1:0] bits are initialized by a USB bus reset.

#### **CCPL Bit (Control Transfer End Enable)**

When the function controller is selected, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.

When the CCPL bit is set to 1 by software while the corresponding PID[1:0] bits are set to BUF, the USB completes the control transfer status stage.

During control read transfer, the USB 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 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 modifies the CCPL bit from 1 to 0 on receiving a new setup packet.

1 cannot be written to the CCPL bit by software while the INTSTS0.VALID bit is 1.

The CCPL bit is initialized by a USB bus reset.

When the host controller is selected, be sure to write 0 to the CCPL bit.

#### **PBUSY Bit (Pipe Busy)**

The PBUSY bit indicates whether DCP is used or not for the transaction when USB changes the PID[1:0] bits from BUF to NAK.

The USB 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 PID has been set to NAK by software allows checking whether modification of the pipe settings is possible.

For details, refer to section 26.3.4.1, 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 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 mismatch occurs during the transfer in the receiving direction.

When the function controller is selected, the USB sets the SQMON bit to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet.

When the function controller is selected, the USB 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 (Sequence 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 (Sequence 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 indicates 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

**SUREQCLR Bit (SUREQ Bit Clear)**

When the host controller is selected, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The SUREQCLR bit indicates 0.

Set the SUREQCLR bit to 1 through software when communication has stopped with the SUREQ bit being 1 during the setup transaction. However, for normal setup transactions, the USB automatically clears the SUREQ bit to 0 upon completion of the transaction; therefore, clearing the SUREQ bit through software is not necessary.

Controlling the SUREQ bit through the SUREQCLR bit must be done while the DVSTCTR0.UACT bit is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.

When the function controller is selected, be sure to write 0 to the SUREQCLR bit.

**SUREQ Bit (Setup Token Transmission)**

The USB transmits the setup packet by setting the SUREQ bit to 1 when the host controller is selected.

After completing the setup transaction process, the USB generates either the SACK or SIGN interrupt and sets the SUREQ bit to 0.

The USB also sets the SUREQ bit to 0 when software sets the SUREQCLR bit to 1.

Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the desired USB request in the setup transaction. Before setting this bit to 1, check that the PID[1:0] bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not modify the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is completed (the SUREQ bit = 1).

Write 1 to the SUREQ bit only when transmitting the setup token; for other purposes, write 0.

When the function controller is selected, be sure to write 0 to the SUREQ bit.

**BSTS Bit (Buffer Status)**

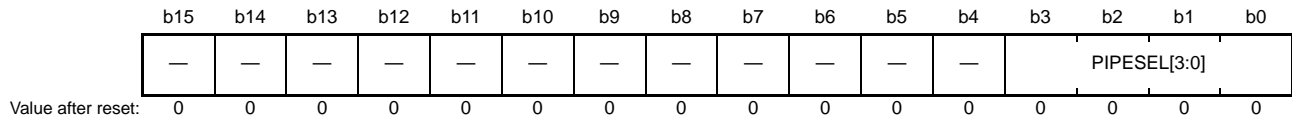
Indicates whether DCP FIFO buffer access is enabled or disabled.

The meaning of the BSTS bit depends on the setting of ISEL bit in the port select register as shown below.

- When the ISEL bit = 0, the BSTS bit indicates whether the received data can be read from the buffer.
- When the ISEL bit = 1, the BSTS bit indicates whether the data to be transmitted can be written to the buffer.

## 26.2.26 Pipe Window Select Register (PIPESEL)

Address(es): 000A 0064h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PIPESEL[3:0]	Pipe Window Select	b3    b0 0 0 0 0: No pipe selected 0 0 0 1: PIPE1 0 0 1 0: PIPE2 0 0 1 1: PIPE3 0 1 0 0: PIPE4 0 1 0 1: PIPE5 0 1 1 0: PIPE6 0 1 1 1: PIPE7 1 0 0 0: PIPE8 1 0 0 1: PIPE9 Settings other than above are prohibited.	R/W
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 the PIPESEL register, 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 the PIPESEL register.

### 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, and PIPEPERI. Writing to these bits is invalid.

## 26.2.27 Pipe Configuration Register (PIPECFG)

Address(es): 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 0000b means an 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 assignment continued at the end of transfer 1: Pipe assignment 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</li> <li>b15 b14</li> <li>0 0: Pipe not used</li> <li>0 1: Bulk transfer</li> <li>1 0: Setting prohibited</li> <li>1 1: Isochronous transfer</li> </ul> <ul style="list-style-type: none"> <li>• PIPE3 to PIPE5</li> <li>b15 b14</li> <li>0 0: Pipe not used</li> <li>0 1: Bulk transfer</li> <li>1 0: Setting prohibited</li> <li>1 1: Setting prohibited</li> </ul> <ul style="list-style-type: none"> <li>• PIPE6 to PIPE9</li> <li>b15 b14</li> <li>0 0: Pipe not used</li> <li>0 1: Setting prohibited</li> <li>1 0: Interrupt transfer</li> <li>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, checking the 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 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, checking the PIPEnCTR.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.

PIPECFG specifies the transfer type, buffer memory access direction, and endpoint numbers for PIPE1 to PIPE9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.



**EPNUM[3:0] Bits (Endpoint Number)**

The EPNUM[3:0] bits specify the endpoint number for the selected pipe.

Setting 0000b means an unused pipe.

Do not make the settings such that the combination of the settings of the DIR and EPNUM[3:0] bits should be the same for two or more pipes (EPNUM[3:0] bits = 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 the DIR bit has been set to 0 by software, the USB uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB 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 the SHTNAK bit has been set to 1 by software for the selected pipe in the receiving direction, the USB modifies the PIPEnCTR.PID[1:0] bits corresponding to the selected pipe to NAK on determining the end of the transfer. The USB 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)**

The BFRE bit specifies the BRDY interrupt generation timing from the USB to the CPU with respect to the selected pipe.

When the BFRE bit has been set to 1 by software and the selected pipe is in the receiving direction, the USB 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, 1 should be written to the BCLR bit in the port control register by software. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.

When the BFRE bit has been set to 1 by software and the selected pipe is in the transmitting direction, the USB does not generate the BRDY interrupt.

For details, refer to section 26.3.3.1, BRDY Interrupt.

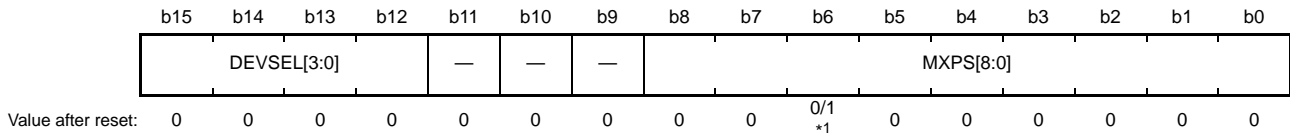
**TYPE[1:0] Bits (Transfer Type)**

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), set the TYPE[1:0] bits to a value other than 00b.

### 26.2.28 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): 000A 006Ch



Bit	Symbol	Bit Name	Description	R/W																					
b8 to b0	MXPS[8:0]	Maximum Packet Size*2	<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																					
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W																					
b15 to b12	DEVSEL[3:0]	Device Select*3	<table border="0"> <tr> <td>b3</td> <td>b0</td> <td></td> </tr> <tr> <td>0 0 0 0:</td> <td>Address 0000</td> <td></td> </tr> <tr> <td>0 0 0 1:</td> <td>Address 0001</td> <td></td> </tr> <tr> <td>0 0 1 0:</td> <td>Address 0010</td> <td></td> </tr> <tr> <td>0 0 1 1:</td> <td>Address 0011</td> <td></td> </tr> <tr> <td>0 1 0 0:</td> <td>Address 0100</td> <td></td> </tr> <tr> <td>0 1 0 1:</td> <td>Address 0101</td> <td></td> </tr> </table> Settings other than above are prohibited.	b3	b0		0 0 0 0:	Address 0000		0 0 0 1:	Address 0001		0 0 1 0:	Address 0010		0 0 1 1:	Address 0011		0 1 0 0:	Address 0100		0 1 0 1:	Address 0101		R/W
b3	b0																								
0 0 0 0:	Address 0000																								
0 0 0 1:	Address 0001																								
0 0 1 0:	Address 0010																								
0 0 1 1:	Address 0011																								
0 1 0 0:	Address 0100																								
0 1 0 1:	Address 0101																								

- Note 1. The value of these bits is 0000h when no pipe is selected with the PIPESEL.PIPSEL[3:0] bits and 0040h when a pipe is selected.
- Note 2. Modify the MXPS[8:0] 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 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, checking the PBUSY bit through software is not necessary.
- Note 3. Modify the DEVSEL[3:0] bits while PID is NAK. To modify 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, checking the PBUSY bit through software is not necessary.

PIPEMAXP specifies the maximum packet size for PIPE1 to PIPE9.

#### MXPS[8:0] Bits (Maximum Packet Size)

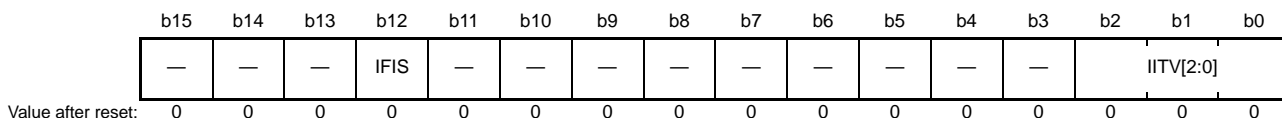
The MXPS[8:0] bits specify the maximum data payload (maximum packet size) for the selected pipe. These bits should be set to the appropriate value for each transfer type based on USB Specification 2.0. Note that the maximum value of PIPE1 and PIPE2 is 256. While MXPS[8:0] = 000h, do not write to the FIFO buffer or do not set the PID[1:0] bits to 01b (BUF).

#### DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the USB device address of the peripheral device which is the communication target. The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2. When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

### 26.2.29 Pipe Cycle Control Register (PIPEPERI)

Address(es): 000A 006Eh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	IITV[2:0] *1	Interval Error Detection Interval	Specify the interval error detection timing for the selected pipe in terms of frames, which is expressed as nth 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

Note 1. Modify the IITV[2:0] bits while PID is NAK. To modify 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, checking the PBUSY bit through software is not necessary.

PIPEPERI selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfer, and sets the interval error detection interval for PIPE1 to PIPE9.

#### 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.

#### IFIS Bit (Isochronous IN Buffer Flush)

Specifies whether to flush the buffer when the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe) is used for isochronous IN transfers.

When the function controller is selected and the selected pipe is for isochronous IN transfers, the USB automatically clears the FIFO buffer when the USB fails to receive the IN token from the USB host within the interval set by the IITV[2:0] bits in terms of frames.

In double buffer mode (the PIPECFG.DBLB bit = 1), the USB only clears the data in the plane used earlier.

The USB clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USB has expected to receive the IN token. Even if the SOF packet is damaged, the USB also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation function.

When the host controller is selected, set the IITV[2:0] bits to 000b.

When the selected pipe is not for isochronous transfer, set the IITV[2:0] bits to 000b.

### 26.2.30 PIPE<sub>n</sub> Control Registers (PIPE<sub>n</sub>CTR) (n = 1 to 9)

- PIPE<sub>n</sub>CTR (n = 1 to 5)

Address(es): PIPE1CTR 000A 0070h, PIPE2CTR 000A 0072h, PIPE3CTR 000A 0074h, PIPE4CTR 000A 0076h, PIPE5CTR 000A 0078h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Bit	Symbol	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	Sequence Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set* <sup>2</sup>	0: Write disabled 1: Specifies DATA1.	R/W* <sup>1</sup>
b8	SQCLR	Sequence Toggle Bit Clear* <sup>2</sup>	0: Write disabled 1: Specifies DATA0.	R/W* <sup>1</sup>
b9	ACLRM	Auto Buffer Clear Mode* <sup>3</sup>	0: Disabled 1: Enabled (all buffers are initialized)	R/W
b10	ATREPM	Auto Response Mode* <sup>2</sup>	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.

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, checking the PBUSY bit through software is not necessary.

Note 3. Modify the ACLRM bit while PID[1:0] 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, checking the PBUSY bit through software is not necessary.

PIPE<sub>n</sub>CTR can be set regardless of the pipe selection in the PIPESEL register.

### 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 26.6 and Table 26.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB 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. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets PID to NAK on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe has been set to 1 by software.
- The USB sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets PID to NAK on detecting a USB bus reset when the function controller is selected.
- The USB sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets PID to STALL (11b) on receiving the STALL handshake when the host controller is selected.

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 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 PID to NAK has been set to NAK by software allows checking whether modification of the pipe settings is possible.

For details, refer to section 26.3.4.1, Pipe Control Register Switching Procedures.

### SQMON Bit (Sequence 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 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 mismatch occurs during the transfer in the receiving direction.

### SQSET Bit (Sequence 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 to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

**SQCLR Bit (Sequence Toggle Bit Clear)**

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.

Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB 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 26.8 shows 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 function controller is selected and 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 responds to the token from the USB host as described below.

(1) When the relevant pipe is for bulk IN transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 1)  
When the ATREPM bit = 1 and PID = BUF, the USB transmits a zero-length packet in response to the IN token.

The USB updates (allows toggling of) the sequence toggle bit (DATA-PID) each time the USB 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 does not generate the BRDY or BEMP interrupt.

(2) When the relevant pipe is for bulk OUT transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 0)  
When the ATREPM bit = 1 and PID = BUF, the USB 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.

When the host controller is selected, 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 (the PIPECFG.DIR bit = 1), the USB sets the INBUFM bit to 1 when the CPU or DTC completes writing data to at least one FIFO buffer plane.

The USB sets the INBUFM bit to 0 when the USB completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (the PIPECFG.DBLB bit = 1), the USB sets the INBUFM bit to 0 when the USB completes transmitting the data from the two FIFO buffer planes before the CPU or DTC 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 (the PIPECFG.DIR bit = 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 bits as shown in Table 26.9.

**Table 26.6 Operation of USB depending on PID[1:0] Bits Setting (When Host Controller is Selected)**

Bits PID[1:0]	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01b (BUF)	Bulk or interrupt	Operation does not depend on the setting.	Issues tokens while the DVSTCTR0.UACT bit is 1 and the FIFO buffer corresponding to the relevant pipe is ready for transmission and reception. Does not issue tokens while the DVSTCTR0.UACT bit is 0 or the FIFO buffer corresponding to the relevant pipe is not ready for transmission or reception.
	Isochronous	Operation does not depend on the setting.	Issues tokens irrespective of the status of the FIFO buffer corresponding to the relevant pipe.
10b (STALL) or 11b (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.

**Table 26.7 Operation of USB depending on PID[1:0] Bits Setting (When Function Controller is Selected)**

Bits PID[1:0]	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK 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.
01b (BUF)	Bulk	Receiving direction (DIR bit = 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.
	Interrupt	Receiving direction (DIR bit = 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.
	Bulk or interrupt	Transmitting direction (DIR bit = 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 bit = 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.
	Isochronous	Transmitting direction (DIR bit = 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.
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 26.8 Information Cleared by USB 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 26.9 Operation of BSTS Bit**

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Bit Function
0	0	0	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	The received data can be read from the FIFO buffer. The BCLR bit in the port control register has been set to 1 by software after the received data has been completely read from the FIFO buffer.
		1	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
1	0	0	The transmit data can be written to the FIFO buffer. 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): PIPE6CTR 000A 007Ah, PIPE7CTR 000A 007Ch, PIPE8CTR 000A 007Eh, PIPE9CTR 000A 0080h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	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	Sequence Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W *1
b8	SQCLR	Sequence 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. Only 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, 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, 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 26.6 and Table 26.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the setting of the PID[1:0] bits.

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. However, if the PID bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets PID to NAK on detecting a USB bus reset when the function controller is selected.
- The USB sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets PID to STALL (11b) on receiving the STALL handshake when the host controller is selected.

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 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 PID has been set to NAK by software allows checking whether modification of the pipe settings is possible.

### **SQMON Bit (Sequence 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 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 mismatch occurs during the transfer in the receiving direction.

### **SQSET Bit (Sequence 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 through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

### **SQCLR Bit (Sequence Toggle Bit Clear)**

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.

Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB 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 26.10 shows 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 PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 26.9.

**Table 26.10 Information Cleared by USB by Setting the ACLRM Bit = 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	The interval count value when the selected pipe is for interrupt transfer and the host controller is selected	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	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

### 26.2.31 PIPEn Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5)

Address(es): PIPE1TRE 000A 0090h, PIPE2TRE 000A 0094h, PIPE3TRE 000A 0098h, PIPE4TRE 000A 009Ch,  
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, 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 to control hardware as described below on having received the number of packets equal to the setting of the TRNCNT[15:0] bits.

- While the PIPECFG.SHTNAK bit is 1, the USB modifies the PID bits to NAK for the corresponding pipe on having received the number of packets equal to the setting of the TRNCNT[15:0] bits.
- While the PIPECFG.BFRE bit is 1, the USB asserts the BRDY interrupt on having received the number of packets equal to the setting of the TRNCNT[15:0] bits and then reading the last received data.

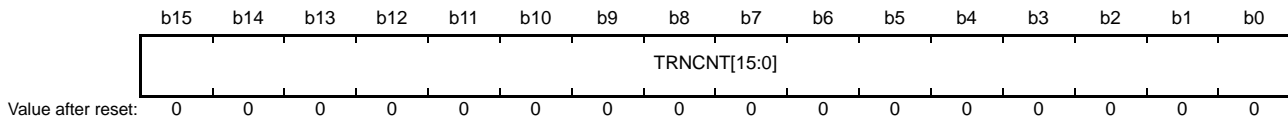
For the pipe in the transmitting direction, set the TRENB bit to 0.

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.

### 26.2.32 PIPE<sub>n</sub> Transaction Counter Register (PIPE<sub>n</sub>TRN) (n = 1 to 5)

Address(es): PIPE1TRN 000A 0092h, PIPE2TRN 000A 0096h, PIPE3TRN 000A 009Ah, PIPE4TRN 000A 009Eh,  
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 total of packets (number of transactions) to be received in corresponding PIPE.</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 increments the value of the TRNCNT[15:0] bits by one when all of the following conditions are satisfied on receiving the packet.

- The PIPE<sub>n</sub>TRE.TRENB bit = 1
- (TRNCNT[15:0] set value  $\neq$  current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPE<sub>n</sub>MAXP.MXPS[8:0] bits.

The USB sets the value of the TRNCNT[15:0] bits to 0 when any of the following conditions are satisfied.

- All of the following conditions are satisfied.
  - The 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 PIPE<sub>n</sub>MAXP.MXPS[8:0] bits.
- All of the following conditions are satisfied.
  - The PIPE<sub>n</sub>TRE.TRENB bit = 1
  - The USB has received a short packet.
- All of the following conditions are satisfied.
  - The PIPE<sub>n</sub>TRE.TRENB bit = 1
  - The PIPE<sub>n</sub>TRE.TRCLR bit has been set to 1 by software.

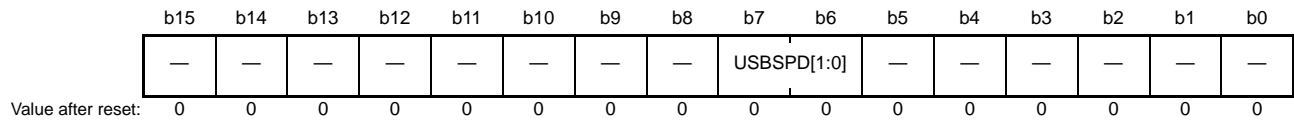
For the pipe in the transmitting direction, set the TRNCNT[15:0] bits to 0.

When the transaction counter is not used, set the TRNCNT[15:0] bits to 0.

Setting the number of transactions to be transferred to the TRNCNT[15:0] bits is only enabled when the PIPE<sub>n</sub>TRE.TRENB bit is 0. To modify the number of transactions to be transferred, set the TRCLR bit to 1 (to clear the current counter value) before setting the PIPE<sub>n</sub>TRE.TRENB bit to 1.

### 26.2.33 Device Address n Configuration Register (DEVADDn) (n = 0 to 5)

Address(es): DEVADD0 000A 00D0h, DEVADD1 000A 00D2h, DEVADD2 000A 00D4h, DEVADD3 000A 00D6h,  
DEVADD4 000A 00D8h, DEVADD5 000A 00DAh



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
b7, b6	USBSPD[1:0]	Transfer Speed of Communication Target Device	b7 b6 0 0: DEVADDn is not used 0 1: Low-speed 1 0: Full-speed 1 1: Setting prohibited	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DEVADDn specifies the transfer speed of the peripheral device which is the communication target for PIPE0 to PIPE9. When the host controller is selected, the bits in DEVADDn should be set before starting communication using each pipe. The bits in DEVADDn should be modified while no valid pipes are using the settings of the bits. Valid pipes refer to the pipes satisfying both of the following conditions:

- DEVADDn is selected by the DEVSEL[3:0] bits.
- The PID[1:0] bits are set to BUF for the selected pipe or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1.

#### USBSPD[1:0] Bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the communication target peripheral device.

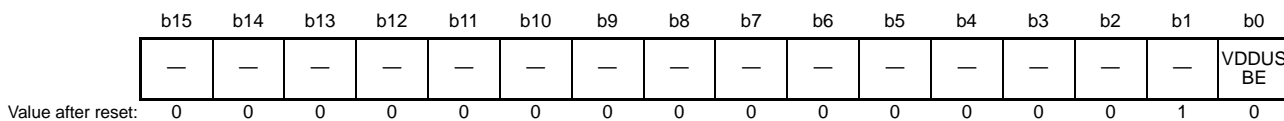
Set these bits to 01b when a low-speed device is connected, whereas set them to 10b when a full-speed device is connected.

When the host controller is selected, the USB refers to the setting of the USBSPD[1:0] bits to generate packets.

When the function controller is selected, set these bits to 00b.

### 26.2.34 USB Module Control Register (USBMC)

Address(es): 000A 00CCh



Bit	Symbol	Bit Name	Description	R/W
b0	VDDUSBE	USB Power Supply Circuit On/Off Control	0: USB power supply circuit off 1: USB power supply circuit on	R/W
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### VDDUSBE Bit (USB Power Supply Circuit On/Off Control)

The USB power supply circuit generates the reference voltage for battery charging. Set this bit to 1 when using the battery charging function.

## 26.2.35 BC Control Register 0 (USBBCCTRL0)

Address(es): 000A 00B0h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	PDDETSTS0	CHGDETSTS0	BATCHGE0	—	VDMSRCE0	IDPSINKE0	VDPSRCE0	IDMSINKE0	IDPSRCE0	RPDME0
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	RPDME0	D– Pin Pull-Down Control	0: Pull-down off 1: Pull-down on	R/W
b1	IDPSRCE0	D+ Pin IDPSRC Output Control	0: Stop 1: 10 $\mu$ A output	R/W
b2	IDMSINKE0	D– Pin 0.6 V Input Detection (Comparator and Sink) Control	0: Detection off 1: Detection on (comparator and sink current on)	R/W
b3	VDPSRCE0	D+ Pin VDPSRC (0.6 V) Output Control	0: Stop 1: 0.6 V output	R/W
b4	IDPSINKE0	D+ Pin 0.6 V Input Detection (Comparator and Sink) Control	0: Detection off 1: Detection on (comparator and sink current on)	R/W
b5	VDMSRCE0	D– Pin VDMSRC (0.6 V) Output Control	0: Stop 1: 0.6 V output	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	BATCHGE0	BC (Battery Charger) Function Ch0 General Enable Control	0: Disabled 1: Enabled	R/W
b8	CHGDETSTS0	D– Pin 0.6 V Input Detection Status*1	0: Not detected 1: Detected	R
b9	PDDETSTS0	D+ Pin 0.6 V Input Detection Status*2	0: Not detected 1: Detected	R
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Valid when IDMSINKE0 = 1.

Note 2. Valid when IDPSINKE0 = 1.

### RPDME0 Bit (D– Pin Pull-Down Control)

When using the battery charging function, set this bit to 1 to control the pull-down resistor of the D– pin.

### IDPSRCE0 Bit (D+ Pin IDPSRC Output Control)

With this bit set to 1, when the function controller is selected, current output is enabled upon detection of the connection of the data pin and the D+ pin is pulled up.

### IDMSINKE0 Bit (D– Pin 0.6 V Input Detection (Comparator and Sink) Control)

With this bit set to 1, when the function controller is selected, the USB module detects whether VDMSRC (0.6 V) that is output from the host to D– upon primary detection is connected, or whether VDPSRC (0.6 V) that is output from the function to D+ is connected to the function's D– via the host.

### VDPSRCE0 Bit (D+ Pin VDPSRC (0.6 V) Output Control)

With this bit set to 1, when the function controller is selected, output is enabled upon primary detection and VDPSRC (0.6 V) is applied to D+.



**IDPSINKE0 Bit (D+ Pin 0.6 V Input Detection (Comparator and Sink) Control)**

With this bit set to 1, when the function controller selected, the USB module detects whether VDMSRC (0.6 V) that is output from the function to D– is connected to the function's D+ (DCP) via the host. When the host controller is selected, the USB module detects whether VDPSRC (0.6 V) that is output from the function to D+ upon primary detection is connected.

**VDMSRCE0 Bit (D– Pin VDMSRC (0.6 V) Output Control)**

With this bit set to 1, when the function controller selected, output is enabled upon secondary detection and VDMSRC (0.6 V) is applied to D–. When the host controller is selected, output is enabled upon primary detection and VDMSRC (0.6 V) is applied to D–.

**CHGDETSTS0 Flag (D– Pin 0.6 V Input Detection Status)**

When the host controller is selected, this flag is set to 1 if the USB module detects whether VDMSRC (0.6 V) that is output from the host to D– during primary detection is connected, or whether VDPSRC (0.6 V) that is output from the function to D+ is connected to the function's D– via the host.

**PDDTSTS0 Flag (D+ Pin 0.6 V Input Detection Status)**

When the function controller is selected, this flag is set to 1 if the USB module detects whether VDMSRC (0.6 V) that is output from the function to D– during secondary detection is connected to the function's D+ (DCP) via the host. When the host controller is selected, this bit is set to 1 if the USB module detects whether VDPSRC (0.6 V) that is output from the function to D+ during primary detection is connected.

## 26.3 Operation

### 26.3.1 System Control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

#### 26.3.1.1 Setting Data to the USB Related Register

Setting the SYSCFG.USBE bit to 1 after starting the clock supply to the USB (SYSCFG.SCKE bit = 1) enables and starts USB operation.

#### 26.3.1.2 Controller Function Selection

For the USB, the host or function controller can be selected using the SYSCFG.DCFM bit. Note that the DCFM bit should be modified in the initial settings immediately after a reset is released or when pulling up of the D+ line and pulling down of the D+ and D- lines are disabled (the SYSCFG.DPRPU bit = 0 and DRPD bit = 0).

#### 26.3.1.3 Controlling USB Data Bus Resistors

The USB has pull-up and pull-down resistors for the D+ and D- lines. Pull up or pull down these lines by setting the SYSCFG.DPRPU and DRPD bits.

When the function controller is selected, confirm that connection to the USB host is made, then set the SYSCFG.DPRPU bit to 1 and pull up the D+ line (during full-speed) and D- line (during low-speed).

When the SYSCFG.DPRPU bit is set to 1 during communication with the PC, the USB module disables the pull-up resistor of the USB data line, thus notifying the USB host of disconnection.

When the host controller is selected, set the SYSCFG.DRPD bit and pull down the D+ and D- lines.

**Table 26.11 Controlling USB Data Bus Resistors**

Settings		Controlling USB Data Bus Resistors			
DRPD	DPRPU	DMRPU	D-	D+	Remarks
0	0	0	Open	Open	When not used
0	1	0	Open	Pull-up	When operating as the function controller (full-speed)
0	0	1	Pull-up	Open	When operating as the function controller (low-speed)
1	0	0	Pull-down	Pull-down	When operating as the host controller
Other than above			—	—	Setting prohibited

### 26.3.1.4 Example of USB External Connection Circuit

Figure 26.2 shows an example of OTG connection of the USB connector in the self-powered state.

The USB controls the signals for enabling a pull-up resistor for the D+ signal and pull-down resistors for the D+ and D- signals. These signals can be pulled up or down using the SYSCFG.DPRPU and SYSCFG.DRPD bits.

When the function controller is selected and the DPRPU bit is set to 0 during communication with the host controller, the pull-up resistor of the USB data line is disabled, making it possible to notify the USB host of the device disconnection.

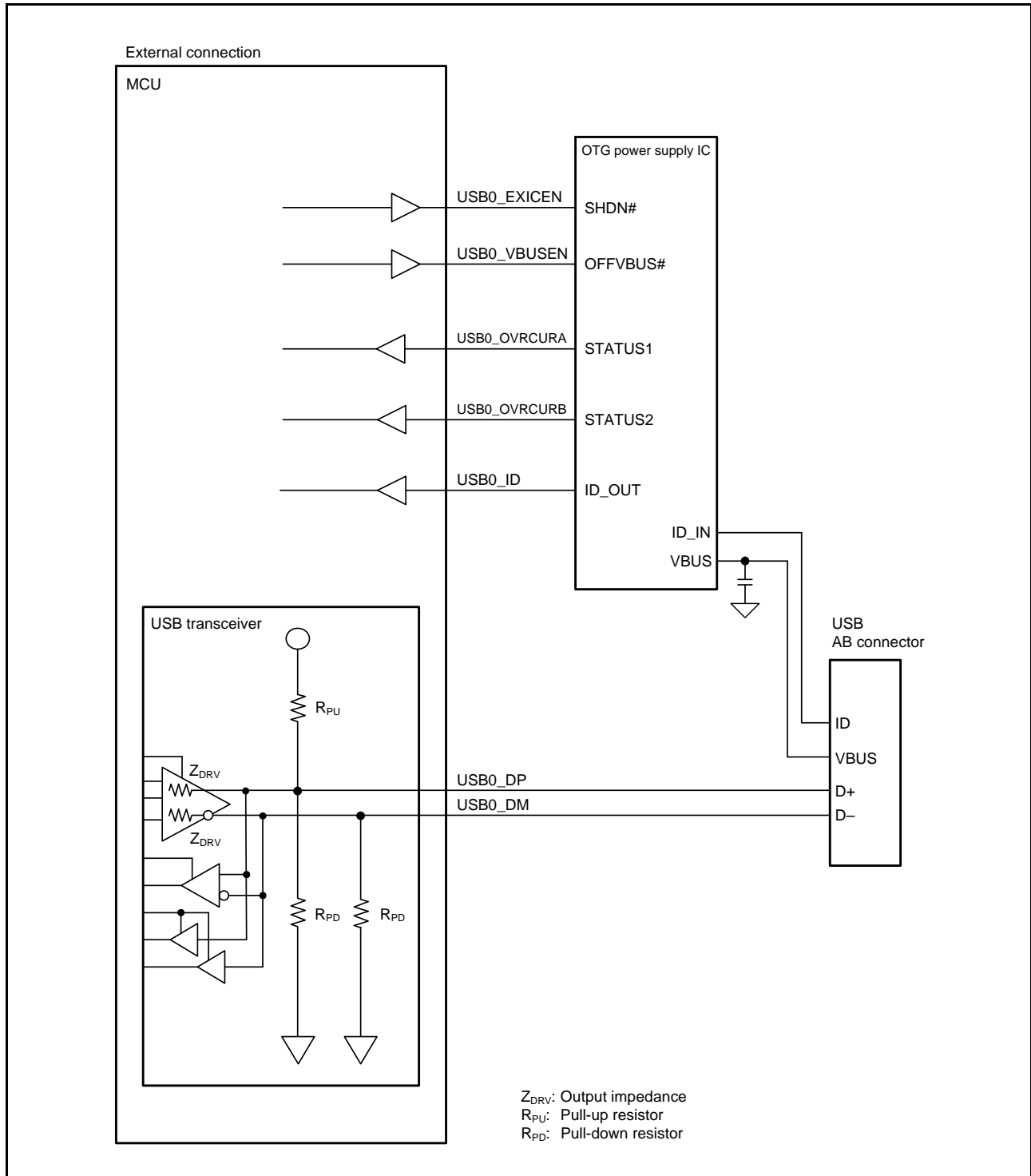


Figure 26.2 Sample OTG Connection of USB Connector in Self-Powered State

Figure 26.3 shows an example of functional connection of the USB connector in the self-powered state.

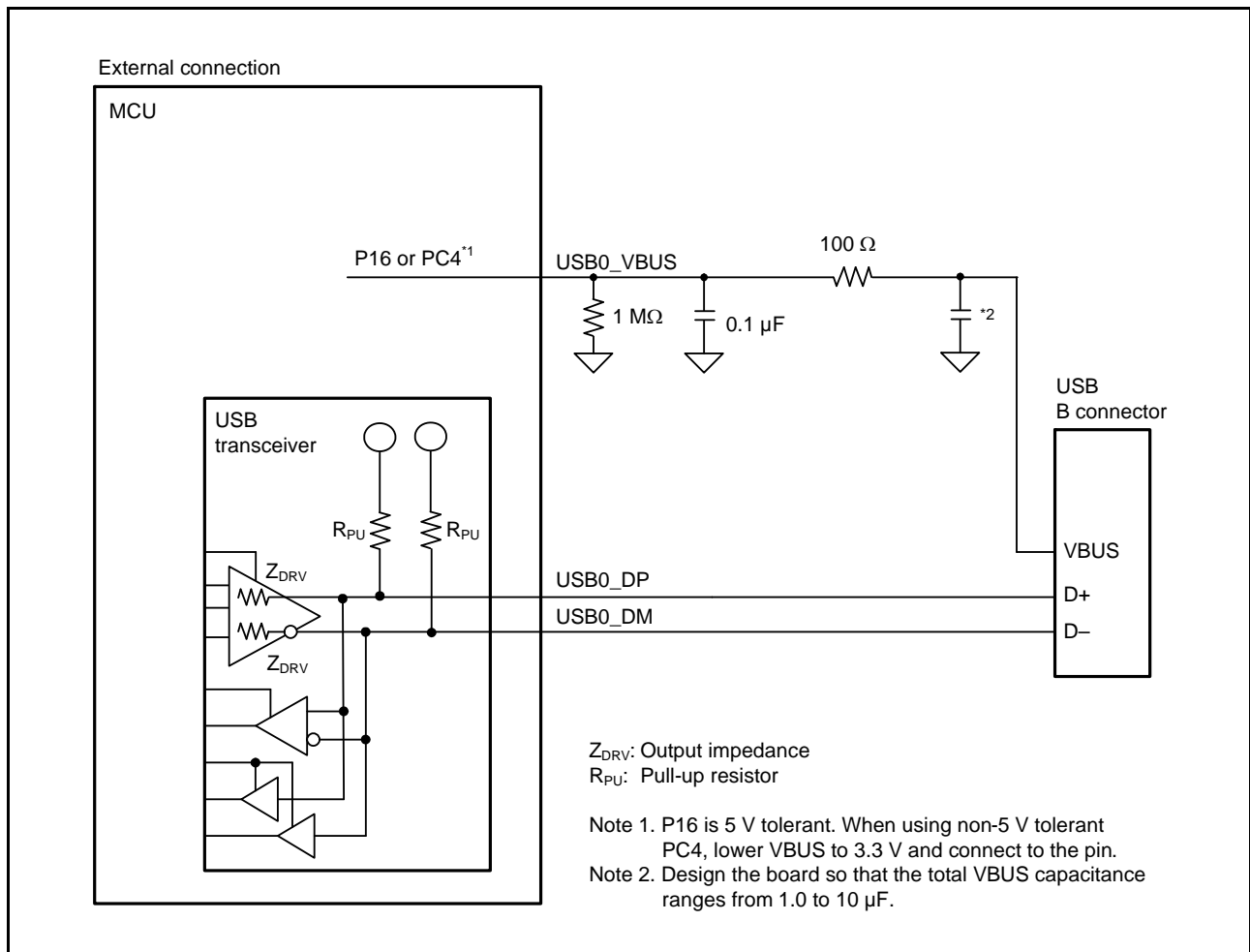


Figure 26.3 Functional Connection of USB Connector in Self-Powered State

Figure 26.4 shows an example of functional connection of the USB connector with Battery Charging Rev 1.2 supported.

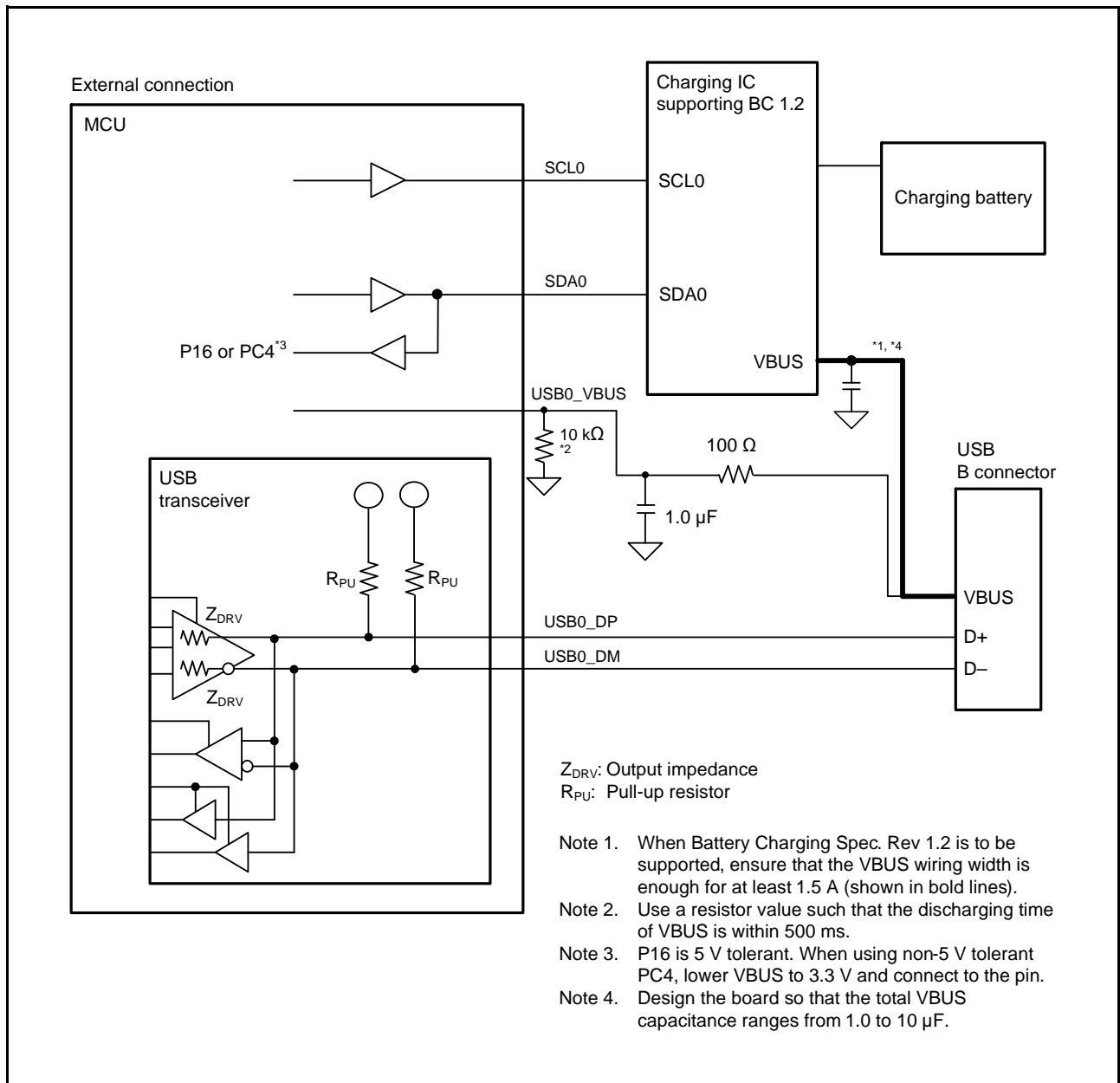


Figure 26.4 Functional Connection Sample of USB Connector with Battery Charging Rev 1.2 Supported

Figure 26.5 shows an example of host connection of the USB connector.

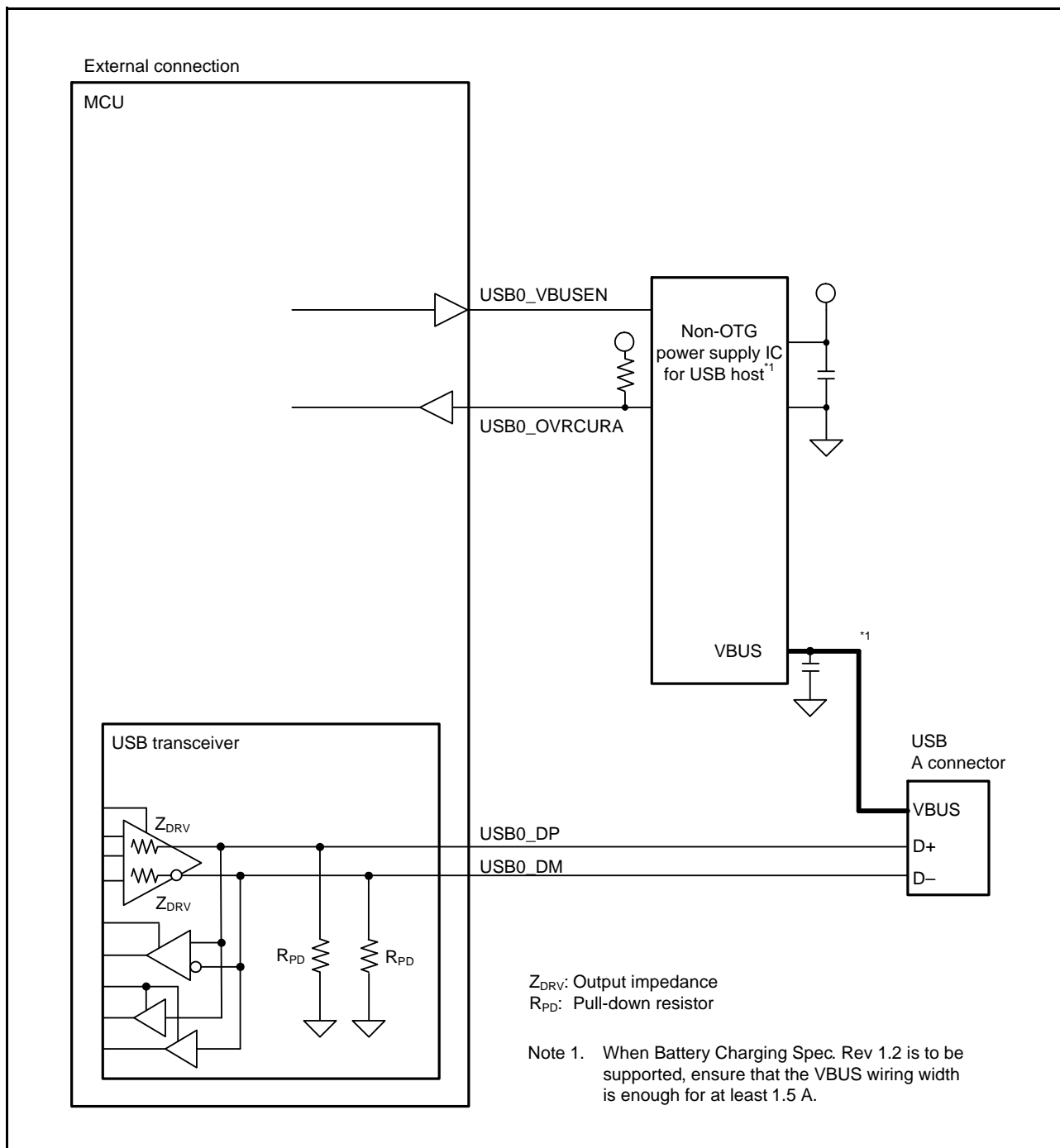
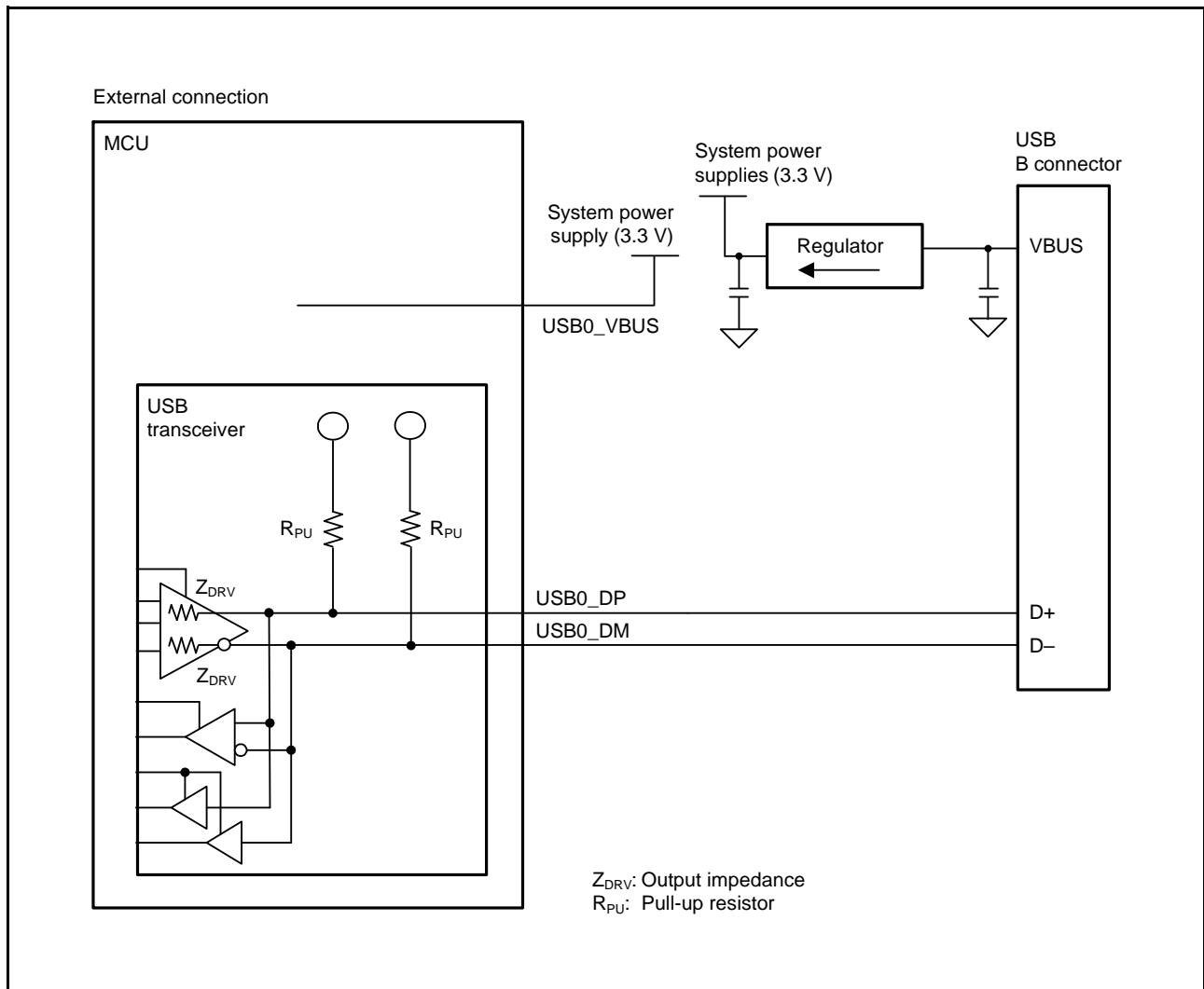


Figure 26.5 Sample Host Connection of USB Connector

Figure 26.6 shows an example of functional connection of the USB connector in bus powered state.



**Figure 26.6 Functional Connection Sample of USB Connector in Bus Powered State**

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

### 26.3.2 Interrupt Sources

Table 26.12 lists the interrupt sources in the USB.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, a USB interrupt request is issued the Interrupt Controller (ICU) and an USB interrupt will be generated.

**Table 26.12 Interrupt Sources**

Bit to be Set	Name	Interrupt Source	Function That Generates the Interrupt	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>	Host/function <sup>*1</sup>	INTSTS0.VBSTS
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>	Function	—
SOFR	Frame number update interrupt	<ul style="list-style-type: none"> <li>[Host controller is selected]</li> <li>When an SOF packet with a different frame number has been transmitted</li> <li>[Function controller is selected]</li> <li>When an SOF packet with a different frame number has been received</li> </ul>	Host/function	—
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>	Function	INTSTS0.DVSQ[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>	Function	INTSTS0.CTSQ[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>	Host/function	BEMPSTS.PIPEnBEMP
NRDY	Buffer not ready interrupt	<ul style="list-style-type: none"> <li>[Host controller is selected]</li> <li>When STALL has been received from the peripheral device for the issued token</li> <li>When a response has not been received correctly from the peripheral device for the issued token (no response was returned three consecutive times or a packet reception error occurred three consecutive times)</li> <li>When an overrun/underrun occurred during isochronous transfer</li> <li>[Function controller is selected]</li> <li>When NAK has been returned for an IN or OUT token while the PID bit = 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>	Host/function	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>	Host/function	BRDYSTS.PIPEnBRDY
OVRRCR	Overcurrent input change interrupt	<ul style="list-style-type: none"> <li>When a change in the state of the USB0_OVRCURA or USB0_OVRCURB input pin has been detected (low to high or high to low)</li> </ul>	Host	INTSTS1.OVRRCR
BCHG	Bus change interrupt	<ul style="list-style-type: none"> <li>When a change of USB bus state has been detected</li> </ul>	Host/function	SYSTS0.LNST[1:0]
DTCH	Disconnection detection during full-speed operation	<ul style="list-style-type: none"> <li>When disconnection of a peripheral device has been detected in full-speed operation</li> </ul>	Host	DVSTCTR0.RHST[2:0]
ATTCH	Device connection detection	<ul style="list-style-type: none"> <li>When J-state or K-state is detected on the USB port for 2.5 μs. Used for checking whether a peripheral device is connected.</li> </ul>	Host	—
EOFERR	EOF error detection	<ul style="list-style-type: none"> <li>When an EOF error of a peripheral device has been detected</li> </ul>	Host	—
SACK	Normal setup operation	<ul style="list-style-type: none"> <li>When the normal response (ACK) for the setup transaction has been received</li> </ul>	Host	—
SIGN	Setup error	<ul style="list-style-type: none"> <li>When a setup transaction error (no response or ACK packet corruption) was detected three consecutive times</li> </ul>	Host	—
PDDTINT0	Portable device detection interrupt	<ul style="list-style-type: none"> <li>When connection of the portable device has been detected</li> </ul>	Host	INTSTS1.PDDTINT0

Note 1. Though this interrupt can be generated while the host function is selected, it is not usually used with the host function.



Figure 26.7 shows the circuits related to the interrupts in the USB.

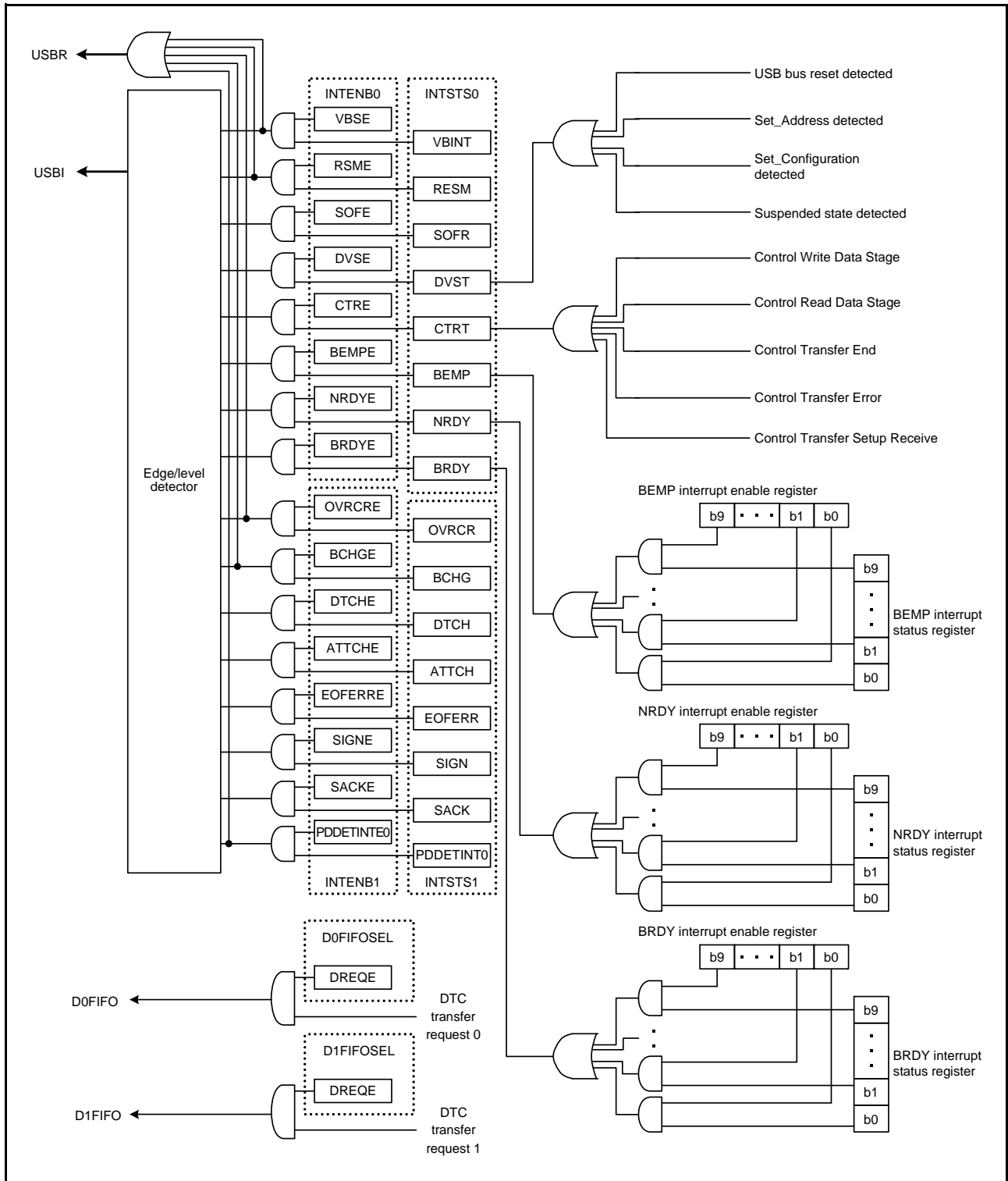


Figure 26.7 Circuits Related to Interrupts in USB

Table 26.13 shows the interrupts generated in the USB0.

**Table 26.13 USB Interrupts**

Interrupt Name	Interrupt Status Flag	DTC Activation	Priority
D0FIFO	DTC transfer request 0	Possible	High
D1FIFO	DTC transfer request 1	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, overcurrent input change interrupt, bus change interrupt, disconnection detection during full-speed operation, device connection detection, EOF error detection, normal setup operation, setup error, and portable device detection interrupt	Not possible	
USBR	VBUS interrupt, resume interrupt, overcurrent input change interrupt, bus change interrupt, and portable device detection interrupt	Not possible	—

### 26.3.3 Interrupt Descriptions

#### 26.3.3.1 BRDY Interrupt

The BRDY interrupt is generated when either of the host controller or function controller is selected. The following shows the conditions under which the USB sets 1 to a corresponding bit in BRDYSTS. Under this condition, the USB 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 the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, the USB generates an internal BRDY interrupt request trigger and sets 1 to the BRDYSTS.PIPEnBRDY bit corresponding to the pertinent pipe.

##### (a) For the pipe in the transmitting direction:

- When the DIR bit is changed from 0 to 1 by software.
- 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 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).

## (b) 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 BSTS bit is read as 0).  
No request trigger is generated for the transaction in which DATA-PID mismatch 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.

When the function controller is selected, the BRDY interrupt is not generated in the status stage of control transfers. The PIPEBRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding PIPEBRDY bit through software. In this case, 1s should be written to the PIPEBRDY bits for the other pipes.  
Clear the BRDY status before accessing the FIFO buffer.

## (2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1

With these settings, the USB 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 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 PIPEn transaction counter register (PIPEnTRN) is used and the number of packets specified by the PIPEnTRN.TRNCNT[15:0] bits are completely received.

When the pertinent data is completely read after any of the above conditions has been satisfied, the USB 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 when the FRDY bit in the FIFO port control register is 1 and the DTLN[8:0] bits are 0. 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 does not detect a BRDY interrupt for the pipe in the transmitting direction.

The PIPEBRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding BRDYSTS.PIPEBRDY bit through software. In this case, 1s 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 the SOFCFG.BRDYM Bit = 1 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS bit setting for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

(a) 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.

(b) For the pipe in the receiving direction:

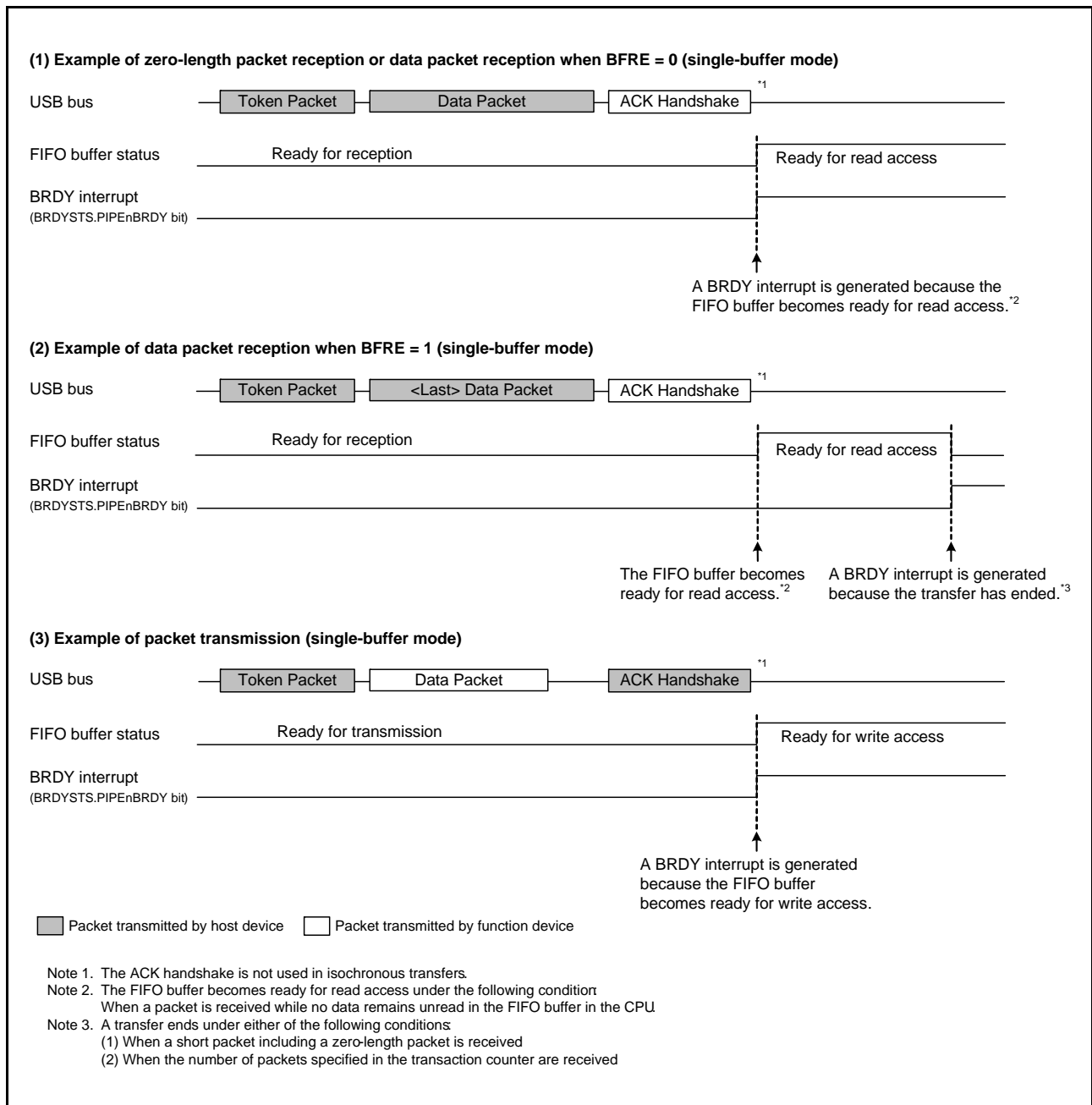
The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access).

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 BCLR = 1 is written through software.

With this setting, the PIPEnBRDY bit cannot be set to 0 through software.

When the SOFCFG.BRDYM bit is set to 1, all PIPECFG.BFRE bits (for all pipes) should be set to 0.

Figure 26.8 shows the timing of BRDY interrupt generation.



**Figure 26.8 Timing of BRDY Interrupt Generation**

The condition that USB clears the INTSTS0.BRDY bit depends on the SOFCFG.BRDYM bit setting. Table 26.14 shows the condition for clearing the BRDY bit.

**Table 26.14 Condition for Clearing BRDY Bit**

BRDYM Bit	Condition for Clearing BRDY Bit
0	The USB sets the BRDY bit to 0 when all bits in BRDYSTS have been set to 0 by software.
1	The USB sets the BRDY bit to 0 when the BSTS bits for all piles have become 0.

### 26.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, the USB sets the corresponding PIPEnNRDY bit in NRDYSTS to 1. If the corresponding bit in NRDYENB has been set to 1 by software, the USB sets the INTSTS0.NRDY bit to 1 and generates a USB interrupt.

The following describes the conditions on which the USB generates the internal NRDY interrupt request for a given pipe.

Note that the internal NRDY interrupt request is not generated during setup transaction execution when the host controller is selected. During setup transactions when the host controller is selected, the SACK or SIGN interrupt is detected.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer when the function controller is selected.

#### (1) When Host Controller is Selected

##### (a) For the pipe in the transmitting direction:

On any of the following conditions, the USB detects an NRDY interrupt.

- For the pipe for isochronous transfers, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer.

In this case, the USB transmits a zero-length packet following the OUT token and sets the bit corresponding to the NRDYSTS.PIPEnNRDY bit and the FRMNUM.OVRN bit to 1.

- During communications other than setup transactions using the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.

In this case, the USB sets the bit corresponding to the PIPEnNRDY bit to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to NAK.

- During communications other than setup transactions, when the STALL handshake is received from the peripheral device.

In this case, the USB sets the bit corresponding to the PIPEnNRDY bit to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to STALL (11b).

##### (b) For the pipe in the receiving direction:

- For the pipe for isochronous transfers, when the time to issue an IN token comes while there is no space available in the FIFO buffer.

In this case, the USB discards the received data for the IN token and sets the PIPEnNRDY bit corresponding to the pipe and the OVRN bit to 1.

When a packet error is detected in the received data for the IN token, the USB also sets the FRMNUM.CRCE bit to 1.

- For the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by the USB (when timeout is detected before detection of the DATA packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.

In this case, the USB sets the PIPEnNRDY bit corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to NAK.

- For the pipe for isochronous transfers, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device.

In this case, the USB sets the PIPEnNRDY bit corresponding to the pipe to 1. (The setting of the PID[1:0] bits of the pipe is not modified.)

- For the pipe for isochronous transfers, when a CRC error or a bit stuffing error is detected in the received data packet.  
In this case, the USB sets the PIPEnNRDY bit corresponding to the pipe and the CRCE bit to 1.
- When the STALL handshake is received.  
In this case, the USB sets the PIPEnNRDY bit corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to STALL.

## (2) When Function Controller is Selected

### (a) 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 generates a NRDY interrupt request at the reception of the IN token and sets the NRDYSTS.PIPEnNRDY bit to 1.  
For the pipe for the isochronous transfers in which an interrupt is generated, the USB transmits a zero-length packet and sets the FRMNUM.OVRN bit to 1.

### (b) 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 generates a NRDY interrupt request at the reception of the OUT token and sets the PIPEnNRDY bit to 1 and OVRN bit to 1.  
For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY bit to 1.  
However, during re-transmission (due to DATA-PID mismatch), 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 generates a NRDY interrupt request when SOF is received, and sets the PIPEnNRDY bit to 1.

Figure 26.9 shows the timing of NRDY interrupt generation when the function controller is selected.

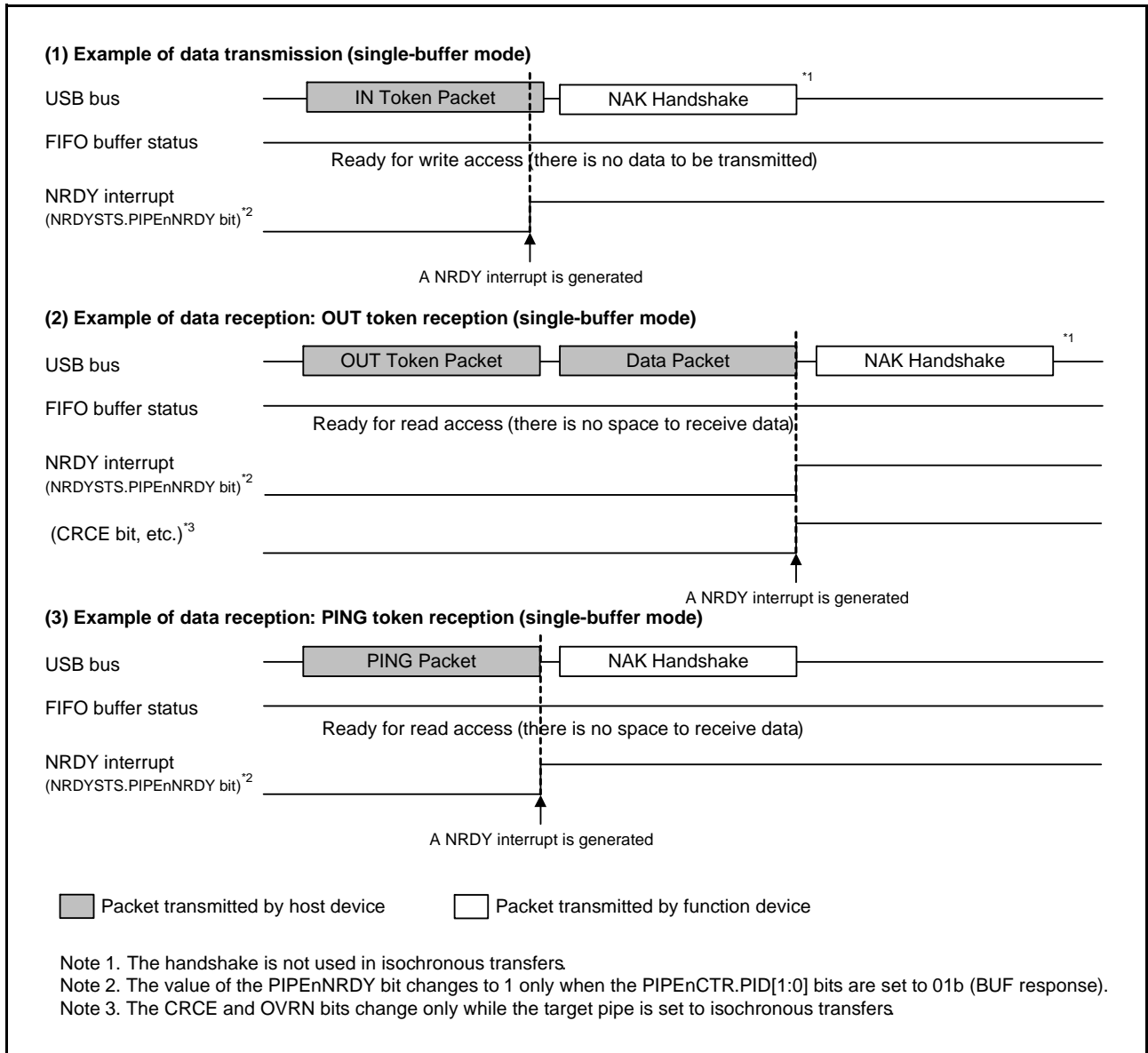


Figure 26.9 Timing of NRDY Interrupt Generation (When Function Controller is Selected)



### 26.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by software, the USB sets the corresponding BEMPSTS.PIPEnBEMP bit to 1. If the corresponding bit in BEMPENB has been set to 1 by software, the USB sets the INTSTS0.BEMP bit to 1 and generates a USB interrupt.

The following describes the conditions on which the USB generates an internal BEMP interrupt request.

**(1) For the pipe in the transmitting direction:**

When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).

In single buffer 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 the CPU or DTC 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 while the function controller is selected.

**(2) For the pipe in the receiving direction:**

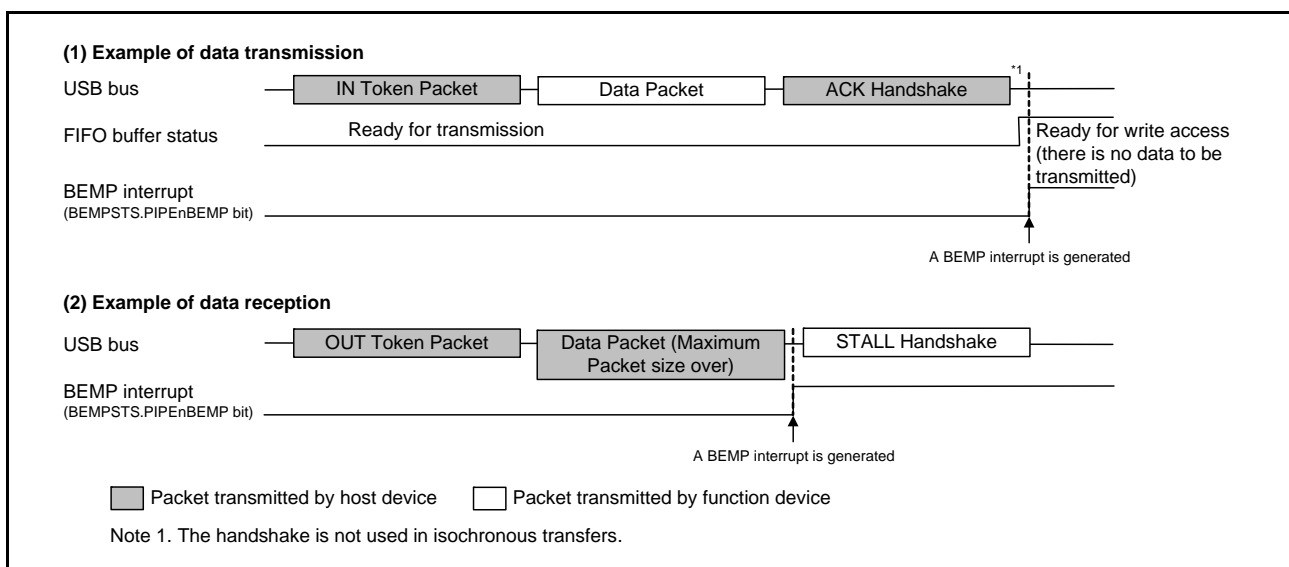
When the successfully-received data packet size exceeds the specified maximum packet size.

In this case, the USB 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 returns no response when used as the host controller, and returns STALL response when used as the function controller.

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 26.10 shows the timing of BEMP interrupt generation when the function controller is selected.



**Figure 26.10 Timing of BEMP Interrupt Generation (When Function Controller is Selected)**

### 26.3.3.4 Device State Transition Interrupt

Figure 26.11 is a diagram of device state transitions in the USB. The USB 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 to which a transition was made can be confirmed using the INTSTS0.DVSQ[2:0] bits.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

Device state can be controlled only when the function controller is selected. The device state transition interrupts can also be generated only when the function controller is selected.

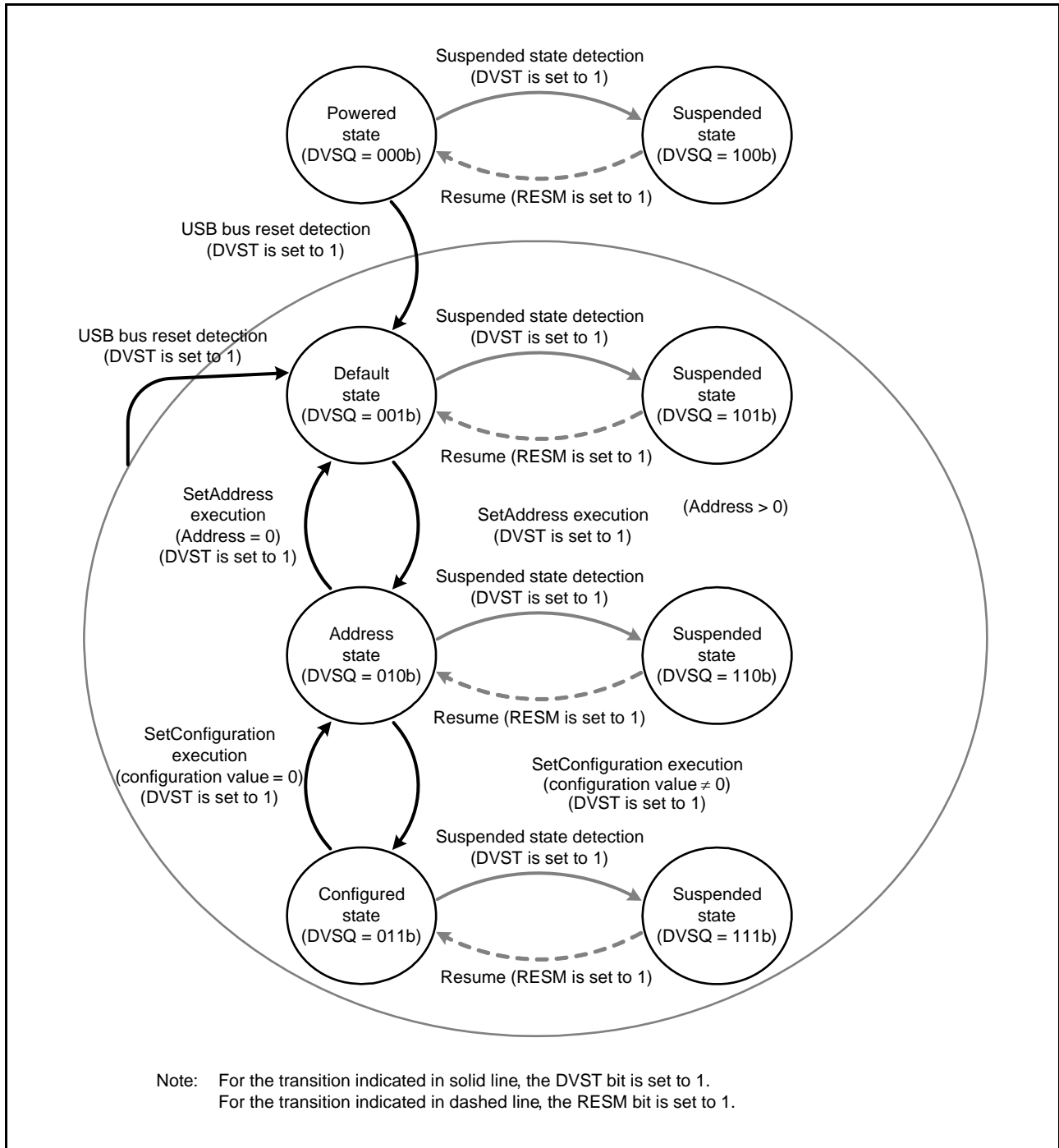


Figure 26.11 Device State Transitions

### 26.3.3.5 Control Transfer Stage Transition Interrupt

Figure 26.12 is a diagram of control transfer stage transitions in the USB. The USB 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 using the INTSTS0.CTSQ[2:0] bits.

Control transfer stage transition interrupts are generated only when the function controller is selected.

The control transfer sequence errors are listed below. If an error occurs, the DCPCTR.PID[1:0] bits 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 (INTSTS0.CTRT = 1), CTSQ[2:0] = 110b value is retained until the CTRT bit = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ[2:0] = 110b 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 retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

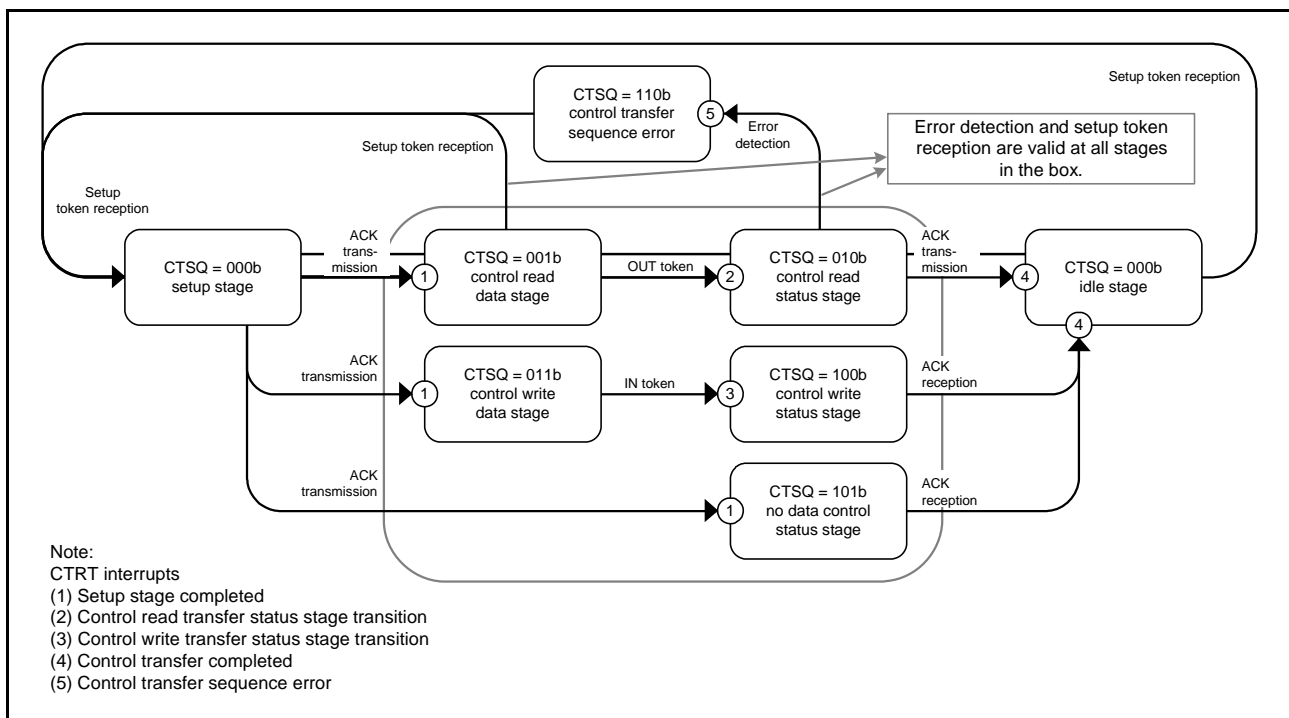


Figure 26.12 Control Transfer Stage Transitions

### 26.3.3.6 Frame Update Interrupt

With the host controller selected, an interrupt is generated at the timing when the frame number is updated. With the function controller selected, an SOFR interrupt is generated when the frame number is updated.

When the function controller is selected, the USB updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

### 26.3.3.7 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. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the USB0\_VBUS pin level.

### 26.3.3.8 Resume Interrupt

When the function controller is selected, 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.

When the host controller is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

### 26.3.3.9 OVRCCR Interrupt

An OVRCCR interrupt is generated when the USB0\_OVRCURA or USB0\_OVRCURB pin level has changed. The levels of the USB0\_OVRCURA and USB0\_OVRCURB pins can be checked with the SYSSTS0.OVCMON[1:0] bits. The external power supply IC can check whether overcurrent has been detected using the OVRCCR interrupt.

For OTG connection, whether a change has been detected in the VBUS comparator can be checked using the OVRCCR interrupt.

### 26.3.3.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether the peripheral device is connected and can also be used to detect a remote wakeup when the host controller is selected. The BCHG interrupt is generated regardless of whether the host controller or function controller is selected.

### 26.3.3.11 DTCH Interrupt

A DTCH interrupt is generated when disconnection of the USB bus is detected while the host controller is selected. The USB detects bus disconnection based on USB Specification 2.0.

After detecting a DTCH interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

### 26.3.3.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral device with the host controller selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

### 26.3.3.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times with the host controller selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

### 26.3.3.14 ATTCH Interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5  $\mu$ s with the host controller selected. To be more specific, an ATTCH interrupt is detected on any of the following conditions.

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5  $\mu$ s.
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5  $\mu$ s.

### 26.3.3.15 EOFERR Interrupt

An EOFERR interrupt is generated when it is detected that communication is not completed at the EOF2 timing prescribed in USB Specification 2.0.

After detecting an EOFERR interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and perform re-enumeration of the pertinent port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

### 26.3.3.16 Portable Device Detection Interrupt

A portable device detection interrupt is generated when the USB module detects a level change (high to low or low to high) in the PDDDET output from the USB-PHY. When a portable device detection interrupt is generated, use software to repeat reading the PDDDETSTS bit until the same value is read three or more times, and perform debouncing.

### 26.3.4 Pipe Control

Table 26.15 lists the pipe settings for the USB. With USB data transfer, data transfer is carried out using the pipe that the software has associated with the endpoint. The USB 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 26.15 Pipe Settings**

Register Name	Bit Name	Setting	Remarks
DCPCFG PIPECFG	TYPE	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	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	DEVSEL	Selects a device	Referenced only when the host controller is selected.
	MXPS	Maximum packet size	Compliant with USB Specification 2.0.
PIPEPERI	IFIS	Buffer flush	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE9: Cannot be set
	IITV	Interval counter	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE5: Cannot be set PIPE6 to PIPE9: Can be set (only when the host controller has been selected)
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.
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller has been selected.
	SUREQCLR	SUREQ clear	Can be set only for the DCP. Can be controlled only when the host controller has been selected.
	ATREPM	Auto response mode	PIPE1 to PIPE5: Can be set Can be set only when the function controller has been selected.
	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	Response PID	Refer to section 26.3.4.6, Response PID.
PIPEnTRE	TRENB	Transaction counter enable	PIPE1 to PIPE5: Can be set
	TRCLR	Current transaction counter clear	PIPE1 to PIPE5: Can be set
PIPEnTRN	TRNCNT	Transaction counter	PIPE1 to PIPE5: Can be set

### 26.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is prohibited (PID = NAK).

The following shows the registers and bits that should not be modified when USB communication is enabled (PID = BUF).

- Bits in DCPCFG and 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 PBUSY bit is set 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, DOFIFOSEL, and D1FIFOSEL.

Registers that should not be set when the CURPIPE[3:0] bits are set:

- Bits in DCPCFG and 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.

### 26.3.4.2 Transfer Types

The PIPECFG.TYPE[1:0] 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.

### 26.3.4.3 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 endpoint 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.

#### 26.3.4.4 Maximum Packet Size Setting

The DCPMAXP.MXPS[6:0] bits and the 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 USB Specification 2.0. 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.

#### 26.3.4.5 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 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 corresponding PIPEnCTR.PID[1:0] bits are 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 PIPEnTRN differs depending on the setting of the PIPEnTRE.TRENB bit.

- The TRENB bit = 0: The specified transaction counter value can be read.
- The TRENB bit = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the 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.



### 26.3.4.6 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 operation with various response PID settings:

#### (1) Response PID settings when the host controller is selected:

The response PID is used to specify the execution of transactions.

- NAK setting: Using pipes is disabled. No transaction is executed.
- BUF setting: Transactions are executed based on the status of the buffer memory.  
For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.  
For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
- STALL setting: Using pipes is disabled. No transaction is executed.

Note: Setup transactions for the DCP are set with the DCPCTR.SUREQ bit.

#### (2) Response PID settings when the function controller is selected:

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] bits setting, and the USB request is stored in the register.

The USB may write to the PID[1:0] bits, depending on the results of the transaction as described below.

#### (3) When the host controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID = NAK is set and issuing of tokens is automatically stopped:  
When a transfer other than isochronous transfer has been performed and an NRDY interrupt is generated.  
(For details, refer to section 26.3.3.2, NRDY Interrupt.)  
- If a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.  
- If the transaction counting ends when the SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID = STALL is set and issuing of tokens is automatically stopped:  
When STALL is received in response to the transmitted token.  
When the size of the receive data packet exceeds the maximum packet size.

#### (4) When the function controller has been selected and 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.
- 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).

#### 26.3.4.7 Data PID Sequence Bit

The USB 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 function controller has been selected and control transfer is used, the USB automatically sets the sequence bit when a stage transition is made. DATA1 is returned when the setup stage is ended. The sequence bit is not referenced and PID = DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller has been selected and control transfer is used, the sequence bit should be set by software at a stage transition. For the ClearFeature request transmission or reception, the data PID sequence bit should be set by software regardless of whether the host controller or function controller is selected.

#### 26.3.4.8 Response PID = NAK Function

The USB 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 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.

#### 26.3.4.9 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 (the PIPECFG.DIR bit = 0), OUT-NAK mode is entered, and during an IN transfer (the DIR bit = 1), null auto response mode is entered.

#### 26.3.4.10 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.

### 26.3.4.11 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, the PIPEnCTR.INBUFM = 0 should be confirmed because the mode can be set only when the buffer is empty. If the 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).

## 26.3.5 FIFO Buffer Memory

### 26.3.5.1 FIFO Buffer Memory

The USB has FIFO buffer memory for data transfer. The memory area used for each pipe is managed by the USB. The FIFO buffer memory has two states depending on whether the access right is assigned to the system (CPU side) or the USB (SIE side).

#### (1) Buffer Status

Table 26.16 and Table 26.17 show the buffer status in the USB. The buffer memory status can be confirmed using the BSTS bit in DCPCTR and the INBUFM bit in PIPEnCTR. The transfer direction for the buffer memory can be specified using either the PIPECFG.DIR bit or the CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM bit is valid for PIPE0 to PIPE5 in the transmitting direction.

When a transmitting pipe uses the double buffer configuration, software can read the BSTS bit to monitor the buffer memory status on the CPU side and the 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 or DTC is slow, software can use the INBUFM bit to confirm the end of transmission.

**Table 26.16 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 26.17 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.

### 26.3.5.2 FIFO Buffer Clearing

Table 26.18 shows the clearing of the FIFO buffer memory by the USB. The buffer memory can be cleared using the BCLR, DnFIFOSEL.DCLRM, and PIPEnCTR.ACLRM bit in the port control register.

Either a single or double buffer configuration can be selected for PIPE1 to PIPE5, using the PIPECFG.DBLB bit.

**Table 26.18 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, 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 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 ACLRM = 1 and ACLRM = 0.

### 26.3.5.3 FIFO Port Functions

Table 26.19 shows the settings for the FIFO port functions of the USB. 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 reading, 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 26.19 FIFO Port Function Settings**

Register Name	Bit Name	Description
CFIFOSEL, DnFIFOSEL (n = 0, 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 DTC transfers (only for DnFIFO).
	MBW	FIFO port access bit width.
	BIGEND	Selects FIFO port endian.
	ISEL	FIFO port access direction (only for DCP).
	CURPIPE	Selects the current pipe.
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the buffer memory.
	BCLR	Clears the buffer memory on the CPU side.
	DTLN	Checks the length of receive data.

#### (1) FIFO Port Selection

Table 26.20 shows 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 FRDY bit in a 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 PIPECFG.DIR bit. Only for the DCP, the ISEL bit in the port select register determines the direction.

**Table 26.20 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
	DTC 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.

## 26.3.6 Control Transfers Using 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.

### 26.3.6.1 Control Transfers When the Host Controller is Selected

#### (1) Setup Stage

USQREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the DCPCTR.SUREQ bit transmits the specified data for setup transactions. Upon completion of the transaction, the SUREQ bit is set to 0. The above USB request registers should not be modified while SUREQ = 1.

After the attached state of the connected function device is detected, the first setup transaction for the device should be issued by using the sequence described above with the DCPMAXP.DEVSEL[3:0] bits set to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.

After the connected function device is shifted to the Address state, setup transactions should be issued by using the sequence described above with the assigned USB address set in the DEVSEL[3:0] bits and the bits in DEVADDn corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in DEVADD2; when PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in DEVADD5.

When the setup transaction data has been sent, an interrupt request is generated according to the response received from the peripheral device (SIGN or SACK bit in INTSTS1), by means of which the result of the setup transactions can be confirmed.

A data packet of DATA0 (USB request) is transmitted as the data packet for a setup transaction regardless of the setting of the DCPCTR.SQMON bit.

#### (2) Data Stage

Data is transferred using the DCP buffer memory.

The access direction of the DCP buffer memory should be specified using the CFIFOSEL.ISEL bit. The transfer direction should be specified using the DCPCFG.DIR bit.

For the first data packet of the data stage, the data PID should be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and the PID bits = BUF. Completion of data transfer is detected using the BRDY or BEMP interrupt.

For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, software should control so as to send a zero-length packet at the end.

#### (3) Status Stage

Zero-length packet data is transferred in the direction opposite to that in the data stage. As in the data stage, data is transferred using the DCP buffer memory. Transactions are done in the same manner as the data stage.

For the data packets of the status stage, the data PID should be set to DATA1 using the DCPCTR.SQSET bit.

For reception of a zero-length packet, the received data length should be confirmed using the CFIFOCTR.DTLN[8:0] bits after a BRDY interrupt is generated, and the buffer memory should then be cleared using the CFIFOCTR.BCLR bit.

### 26.3.6.2 Control Transfers When the Function Controller is Selected

#### (1) Setup Stage

The USB sends an ACK response for a correct setup packet targeted to the USB. The operation of the USB in the setup stage is described below.

When receiving a new setup packet, the USB 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 stores the USB request parameters in USBREQ, USBVAL, USBINDEX, and USBLENG.

Response processing with respect to the control transfer should be carried out after setting the VALID bit = 0. In the VALID bit = 1 state, PID = BUF cannot be set, and the data stage cannot be terminated.

Using the function of the VALID bit, the USB 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 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, see Figure 26.12.

#### (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 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  
A zero-length packet is received from the USB host and an ACK response is sent.
- For control write transfers and no-data control transfers  
A zero-length packet is transmitted and an ACK response is received from the USB host.

#### (4) Control Transfer Auto Response Function

The USB 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.

- bmRequestType is not 00h: Any transfer other than a control write transfer
- wIndex is not 00h: Request error
- wIndex is not 00h: Any transfer other than a no-data control transfer
- wValue is larger than 7Fh: Request error
- INTSTS0.DVSQ[2:0] are 011b (Configured state): Control transfer of a device state error

For all requests other than the SET\_ADDRESS request, a response is required from the corresponding software.

### 26.3.7 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: refer to section 26.3.3.1, (2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1)
- Transaction count function (PIPE<sub>n</sub>TRE.TRENB, TRCLR, and PIPE<sub>n</sub>TRN.TRNCNT[15:0] bits: refer to section 26.3.4.5, Transaction Counter (For PIPE1 to PIPE5 in Reading Direction))
- Response PID = NAK function (PIPECFG.SHTNAK bit: refer to section 26.3.4.8, Response PID = NAK Function)
- Auto response mode (PIPE<sub>n</sub>CTR.ATREPM bit: refer to section 26.3.4.9, Auto Response Mode)

### 26.3.8 Interrupt Transfers (PIPE6 to PIPE9)

When the function controller is selected, the USB carries out interrupt transfers in accordance with the timing controlled by the host controller.

When the host controller is selected, the timing of issuing a token can be specified using the interval counter.

#### 26.3.8.1 Interval Counter during Interrupt Transfers When the Host Controller is Selected

For interrupt transfers, intervals between transactions are set in the PIPEPERI.IITV[2:0] bits. The USB controller issues interrupt transfer tokens based on the specified intervals.

##### (1) Counter Initialization

The interval counter is initialized when the MCU is reset or when the PIPE<sub>n</sub>CTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

Note that the interval counter is not initialized in the following case.

- USB bus reset or USB suspended  
The IITV[2:0] bits are not initialized. Setting 1 to the UACT bit in DVSTCTR0 starts counting from the value before entering the USB bus reset state or USB suspended state.

##### (2) Operation When Transmission/Reception is Impossible at Token Issuance Timing

The USB cannot issue tokens even at token issuance timing in the following cases. In such a case, the USB attempts transactions at the subsequent interval.

- When the PID is set to NAK or STALL.
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the transmitting (OUT) direction.



### 26.3.9 Isochronous Transfers (PIPE1 and PIPE2)

The USB 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 (IDLX function)
- Isochronous IN transfer buffer flush function (specified by the PIPEPERI.IFIS bit)

#### 26.3.9.1 Error Detection in Isochronous Transfers

The USB has a function for detecting the error information described below, so that when errors occur in isochronous transfers, they can be controlled by software. Table 26.21 and Table 26.22 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 the host controller is selected  
When the buffer memory is full at the token sending timing in the IN (receiving) direction.  
When there is no data to be sent in the buffer memory at the token sending timing in the OUT (transmitting) direction.
- When the function controller is selected  
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 when the function controller is selected.

- During an isochronous IN transfer, an IN token could not be received in the interval frame.
- During an isochronous OUT transfer, an OUT token could not be received in the interval frame.

**Table 26.21 Error Detection When a Token is Received**

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	No interrupts generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the FRMNUM.OVRN bit to 1 in both cases when the host controller is selected and function controller is selected. When the function controller is selected, 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 when the function controller is selected. It is not generated when the host controller is selected.

**Table 26.22 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 to 1 bit in both cases when the host controller is selected and the function controller is selected.
3	Maximum packet size exceeded errors	A BEMP interrupt is generated to set the PID[1:0] bits to STALL in both cases when the host controller is selected and the function controller is selected.

### 26.3.9.2 DATA-PID

When the function controller is selected, the USB 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

### 26.3.9.3 Interval Counter

The isochronous transfer interval can be set using the PIPEPERI.IITV[2:0] bits. The interval counter enables the functions shown in Table 26.23 when the function controller is selected. When the host controller is selected, the USB generates the token issuance timing. When the host controller is selected, the interval counter operation is the same as that in the interrupt transfer.

**Table 26.23 Interval Counter Function When the Function Controller is Selected**

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 When the Function Controller is Selected

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

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.

- When the PID[1:0] bits are set to NAK or STALL  
The interval timer does not stop. The USB attempts transactions at the subsequent interval.
- When the USB bus is reset or USB is suspended  
The 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 Host Controller is Selected

The USB controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings.

Specifically, the USB issues a token for a selected pipe once every  $2^{IITV}$  frames.

The USB starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits have been set to BUF by software.

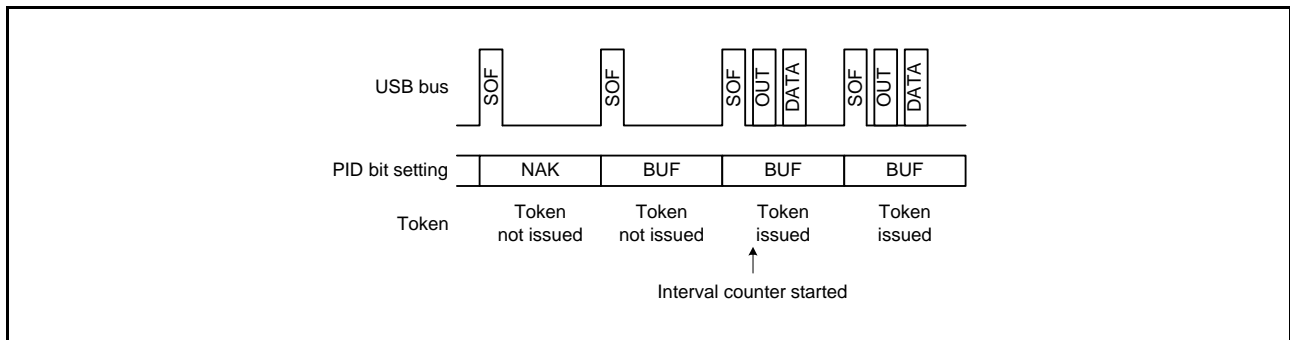


Figure 26.13 Token Issuance When IITV = 0

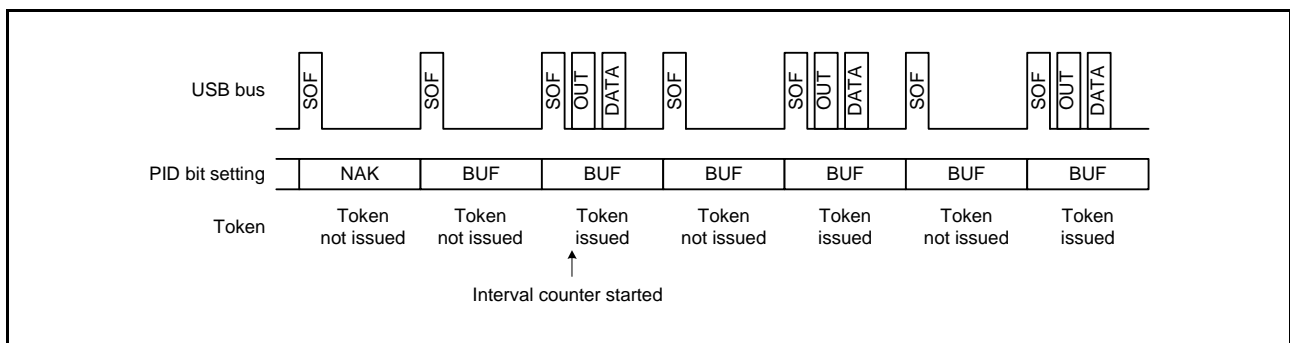


Figure 26.14 Token Issuance When IITV = 1

When the selected pipe is set for isochronous transfers, the USB carries out the following operation in addition to controlling the token issuance interval. The USB issues a token even when the NRDY interrupt generation condition is satisfied.

(a) When the selected pipe is for isochronous IN transfers

The USB generates an NRDY interrupt when the USB issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).

The USB sets the FRMNUM.OVRN bit to 1 generating an NRDY interrupt when the time to issue an IN token comes while the USB cannot receive data because the FIFO buffer is full (due to the fact that the CPU or DTC is too slow to read data from the FIFO buffer).

(b) When the selected pipe is for isochronous OUT transfers

The USB sets the OVRN bit to 1 generating an NRDY interrupt and transmitting a zero-length packet when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer (because the CPU or DTC is too slow to write data to the FIFO buffer).

The token issuance interval is reset on any of the following conditions.

- When the USB is reset through a reset pin (The IITV[2:0] bits are also set to 0).
- When the PIPEnCTR.ACLRM bit has been set to 1 by software.

(3) Interval Counting and Transfer Control When the Function Controller is Selected

(a) When the selected pipe is for isochronous OUT transfers

The USB generates an NRDY interrupt when the USB fails to receive a data packet within the interval set by the PIPEPERI.IITV[2:0] bits.

The USB also generates an NRDY interrupt when the USB 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 bit is set to a value other than 0, the USB 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 does not generate an NRDY interrupt on receiving an SOF packet.

The timing to start interval counting depends on the setting of IITV[2:0] bits as shown below.

- When the 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.

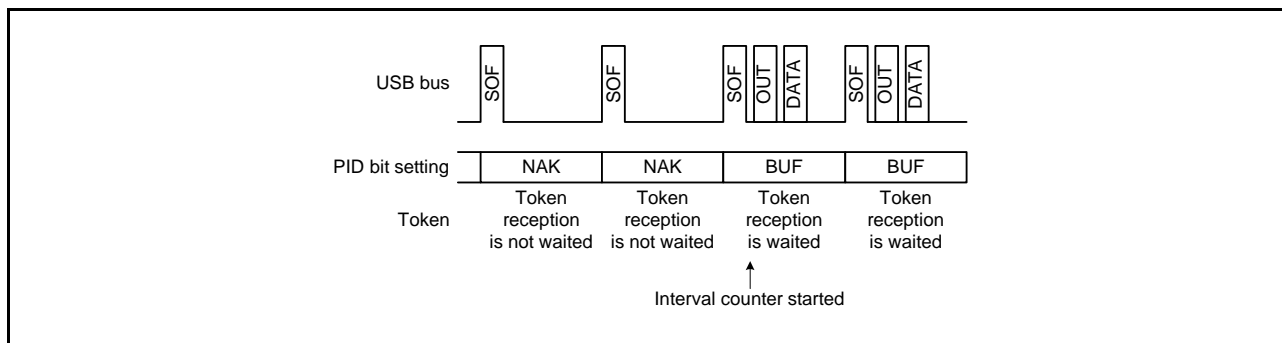
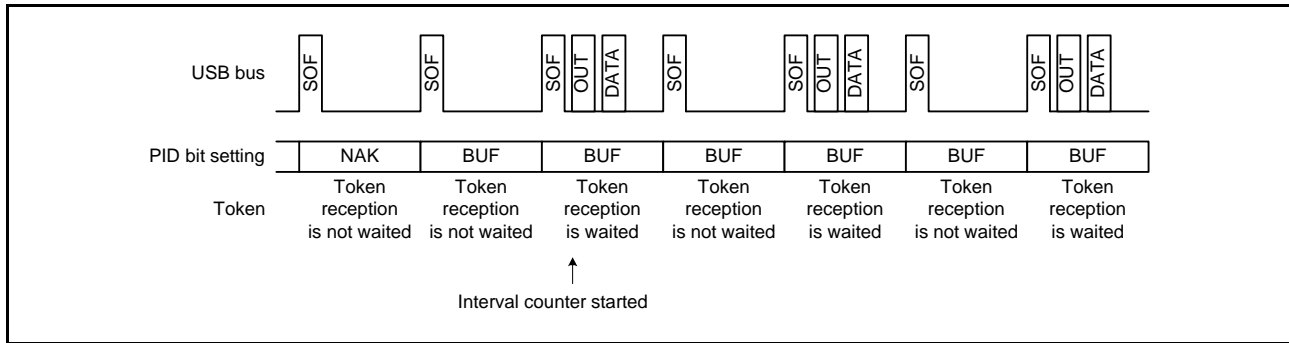


Figure 26.15 Relationship between Frames and Expected Token Reception When IITV = 0

- When the IITV  $\neq 0$ : 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 26.16 Relationship between Frames and Expected Token Reception When IITV  $\neq 0$**

(b) When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit should be 1 for this use. When IFIS = 0, the USB transmits a data packet in response to the received IN token irrespective of the setting of the PIPEPERI.IITV[2:0] bits.

When IFIS = 1, the USB clears the FIFO buffer when the USB 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 also clears the FIFO buffer when the USB 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 setting of the IITV[2:0] bits (similar to the timing during OUT transfers).

The interval is counted on any of the following conditions in function controller mode.

- When a hardware-reset is applied to the USB (here, the IITV[2:0] bits are also set to 000b).
- When the PIPEnCTR.ACLRM bit is set to 1 by software.
- When the USB detects a USB bus reset.

(4) Setup of Data to be Transmitted Using Isochronous Transfer When the Function Controller is Selected

With isochronous data transmission using the USB in the function controller, 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, 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 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 26.17 shows an example of transmission using the isochronous transfer transmission data setup function with the USB when IITV = 0 (every frame) has been set.

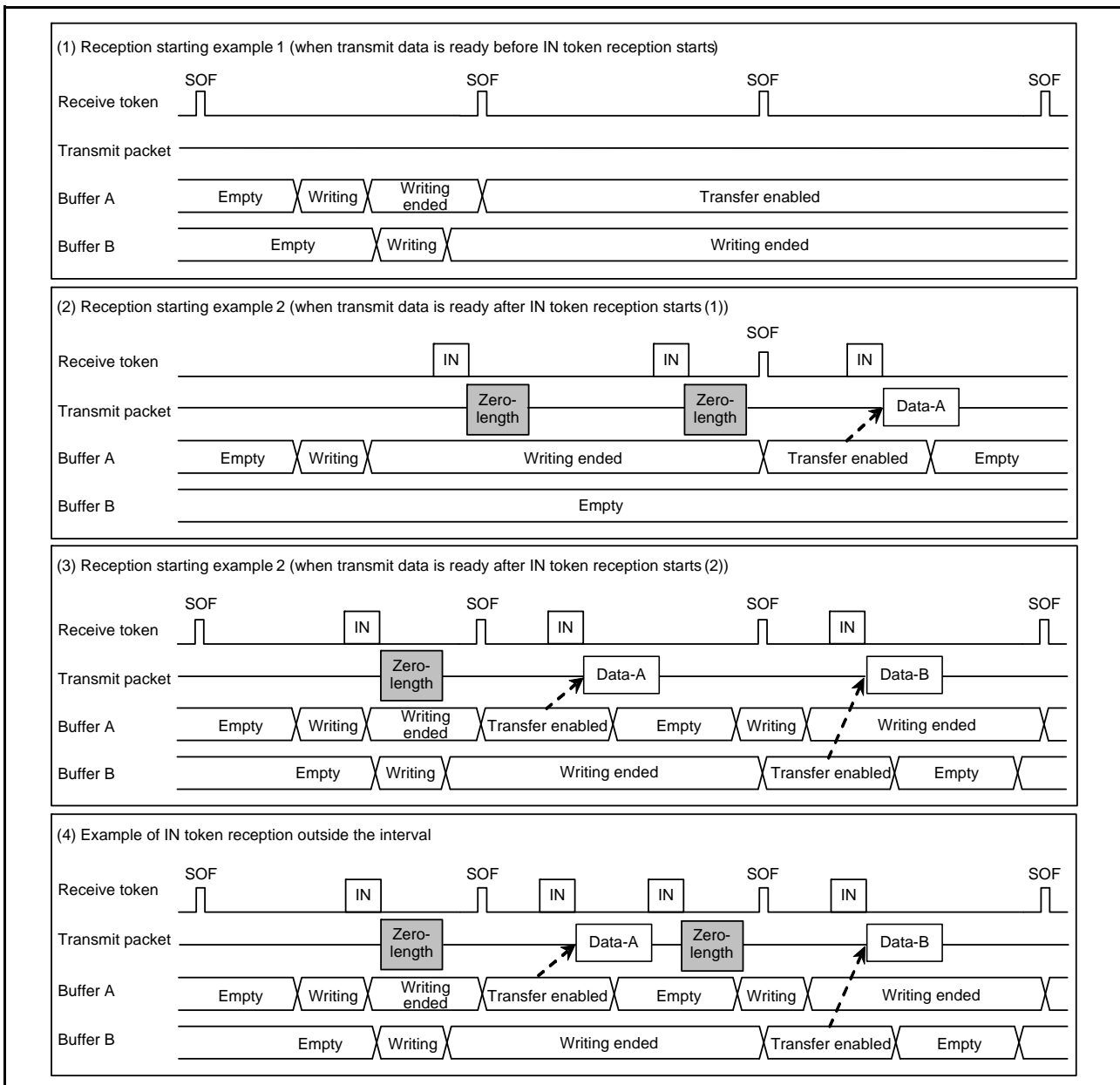


Figure 26.17 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmission Buffer Flush When the Function Controller is Selected

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 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 setting of the PIPEPERI.IITV[2:0] bits.

- When the IITV = 0  
The buffer flush operation starts from the next frame after the pipe becomes valid.
- When the IITV ≠ 0  
The buffer flush operation is carried out after the first successful transaction.

Figure 26.18 shows an example of the buffer flush function in the USB. When an unanticipated token is received before the interval frame, the USB sends the write data or a zero-length packet as an underrun error according to the data setup state.

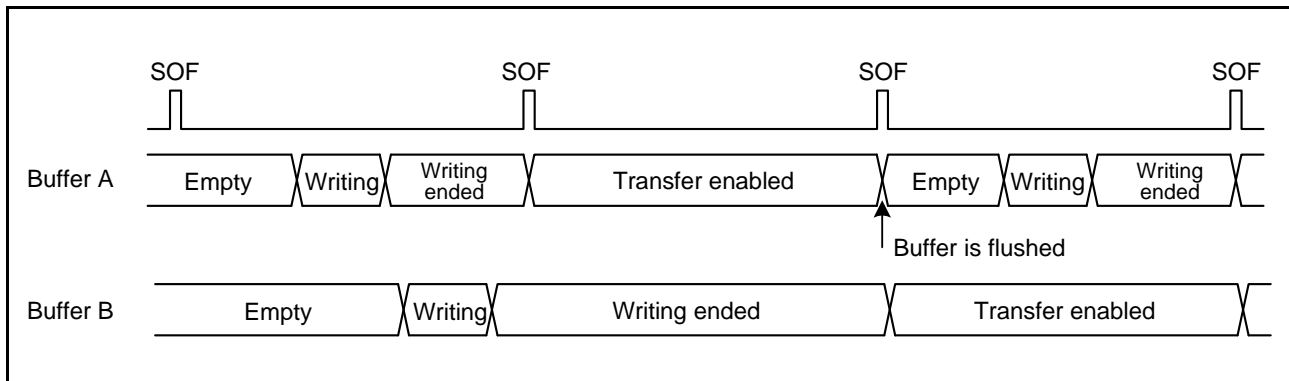


Figure 26.18 Example of Buffer Flush Operation

Figure 26.19 shows an example of interval error occurrence in the USB. 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 FRMNUM.OVRN bit 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.

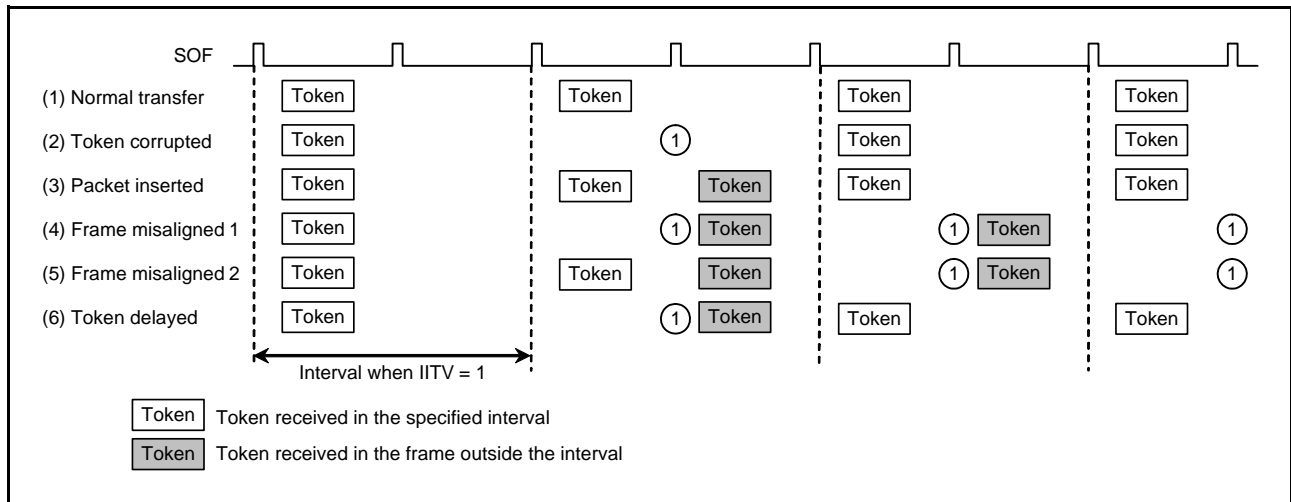


Figure 26.19 Example of Interval Error Occurrence When IITV = 1



### 26.3.10 SOF Interpolation Function

When the function controller is selected and if data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB 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.

- MCU 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 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, the FRMNUM.FRNM[10:0] bits are not updated.

## 26.3.11 Pipe Schedule

### 26.3.11.1 Conditions for Generating a Transaction

When the host controller is selected and the DVSTCTR0.UACT bit has been set to 1, the USB generates a transaction under the conditions shown in Table 26.24.

**Table 26.24 Conditions for Generating a Transaction**

Transaction	Conditions for Generation				
	DIR	PID	IITV0	Buffer State	SUREQ
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Note 1. Symbols (—) in the table indicate that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.

Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.

Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

### 26.3.11.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USB. After the USB sends an SOF, the transfer is carried out in the sequence described below.

#### 1. Execution of periodic transfers

A pipe is searched in the order of PIPE1 → PIPE2 → PIPE6 → PIPE7 → PIPE8 → PIPE9, and then, if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.

#### 2. Setup transactions for control transfers

The DCP is checked, and if a setup transaction is possible, it is sent.

#### 3. Execution of bulk transfers, control transfer data stages, and control transfer status stages

A pipe is searched in the order of DCP → PIPE1 → PIPE2 → PIPE3 → PIPE4 → PIPE5, and then, if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.

When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

### 26.3.11.3 Enabling USB Communication

Setting the DVSTCTR0.UACT bit to 1 initiates SOF transmission and transaction generation is enabled.

Setting the UACT bit to 0 stops SOF transmission and a suspend state is entered. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF is sent.

## 26.4 Usage Notes

### 26.4.1 Setting the Module Stop Function

Operation of the USB module can be disabled or enabled using module stop control register B (MSTPCRB). The setting after a reset is for operation of the USB module to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

## 26.5 Battery Charging Detection Processing

It is possible to control the processing for data contact detection (D+ line contact check), primary detection (charger detection), and secondary detection (charger verification), which are defined in the battery charging specification.

The following describes required operations for a function device and a host device, individually.

### 26.5.1 Processing When Function Controller is Selected

The following processing is required when operating the USB module as a portable device for battery charging.

- (1) Detect when the data lines (D+ and D-) have made contact and start the processing for primary detection.
- (2) After primary detection starts, wait 40 ms for masking, and then check the D- voltage level to confirm the primary detection result.
- (3) If the charger is detected during primary detection, also start secondary detection.
- (4) After secondary detection starts, wait 40 ms for masking, and then check the D+ voltage level to confirm the secondary detection result.

For step (1), after VBUS is detected using the VBIT interrupt and the VBSTS bit, wait for 300 to 900 ms by software, and then set the VDPSRCE and IDMSINKE bits in the BCCTRL register. Or set the IDPSRCE bit, and after a change from high to low on the D+ line is detected using the LNST bits, clear the IDPSRCE bit and set the VDPSRCE and IDMSINKE bits. Set the VDPSRCE and IDMSINKE bits at the same time.\*1

For step (2), set the VDPSRCE and IDMSINKE bits and wait 40 ms by software, and then use the CHGDETSTS bit to verify the primary detection result.\*2

For step (3), if the CHGDETSTS bit is set in step (2), verify that the charger is detected, and then clear the VDPSRCE and IDMSINKE bits and set the VDMSRCE and IDPSINKE bits.

For step (4), set the VDMSRCE and IDPSINKE bits and wait for 40 ms by software, and then use the PDDTSTS bit to verify the secondary detection result.

The following shows the process flow.

Note 1. The battery charging specification describes two implementation methods of the process flow for data contact detection (D+/D- line contact check). One of the methods is to detect a change to logic low due to the pull-down resistor of the host device when the D+ and D- lines have made contact with the target while the D+ line is held at logic high by applying a current of 7 to 13  $\mu$ A on the D+ line. The other method is to wait for 300 to 900 ms after VBUS is detected.

Note 2. During primary detection, when the voltage on the D- line is detected to be 0.25 to 0.4 V or above and 0.8 to 2.0 V or below, the target device is recognized as the host device for battery charging (charging downstream port). When using a PHY in which the 0CHGDETSTS bit only indicates that the voltage on the D- line is 0.25 to 0.4 V or above, add the processing to check that the voltage on D- line is 0.8 V to 2.0 V or below using the LNST bits, as necessary.

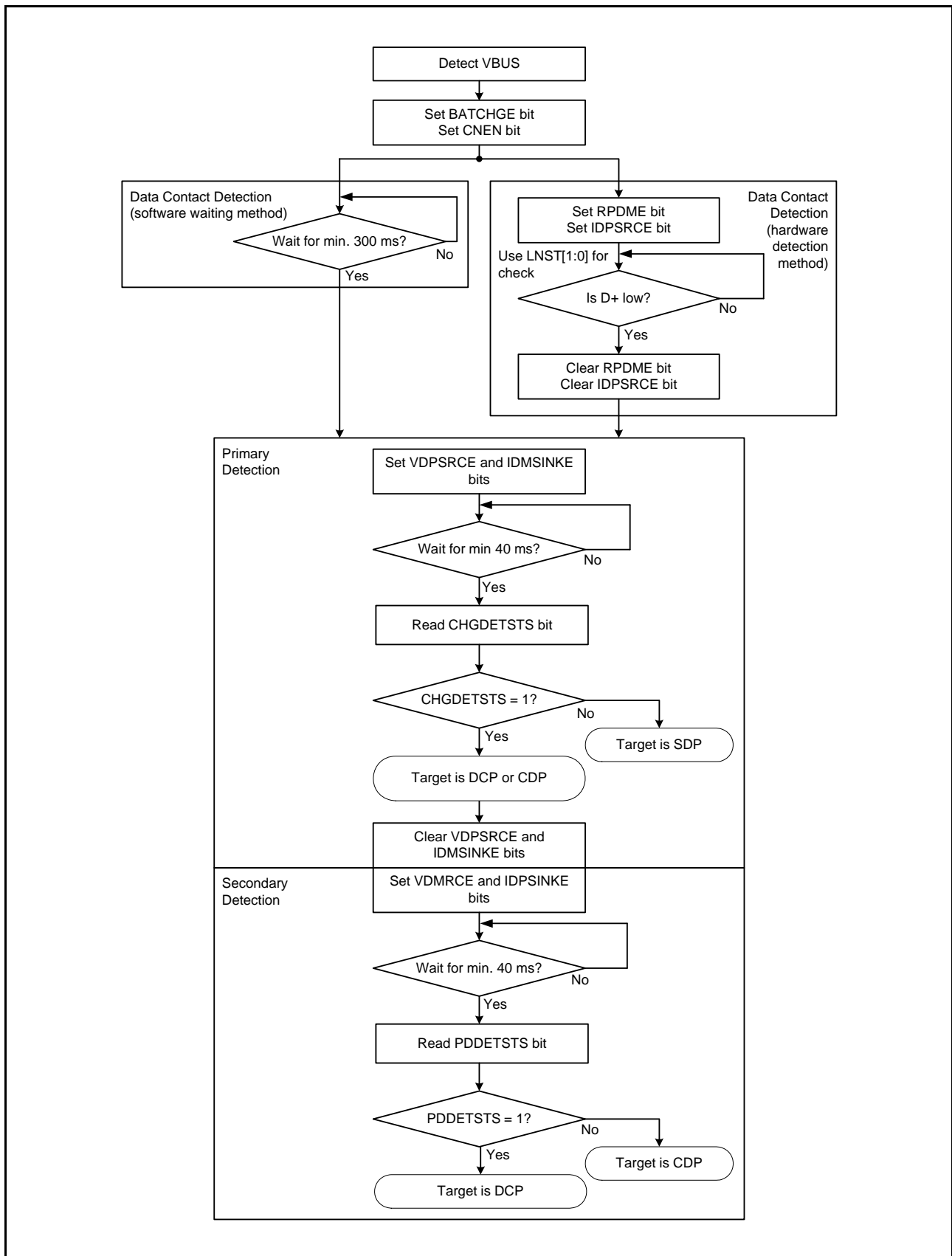


Figure 26.20 Process Flow for Operating as Portable Device

### 26.5.2 Processing When Host Controller is Selected

The following processing is required when operating the USB module as a charging downstream port for battery charging.

- (1) Start driving the VBUS.
- (2) Enable the portable device detection circuit.
- (3) Monitor the portable device detection signal, and start driving the D– line if the detection signal is high.
- (4) Detect when the portable device detection signal is low level and stop driving the D– line.

Or, the following processing can also be used in accordance with the battery charging specification.

- (A) After disconnection is detected, start driving the D– line within 200 ms.
- (B) After connection is detected, stop driving the D– line within 10 ms.

The D– line must be driven to allow the portable device to detect the primary detection described in section 26.5.1, Processing When Function Controller is Selected. The above steps (1) to (4) apply when the portable device detection function is provided by hardware. This method is to drive the D– line when the portable device is detected. Steps (A) and (B) apply when the portable device function is not provided or available by hardware. Regardless of detection of the portable device, the D– line is driven in the disconnected state and the line is not driven in the connected state. In the battery charging specification, either of these methods can be used.

For steps (3) and (4), after a change in the portable device detection signal is detected using the PDDDETINT interrupt, the current signal state can be confirmed by reading the PDDDETSTS bit.

Steps (A) and (B) can be performed only in a software timer.

The following show the process flow for steps (1) to (4) and the process flow for steps (A) to (B), respectively.

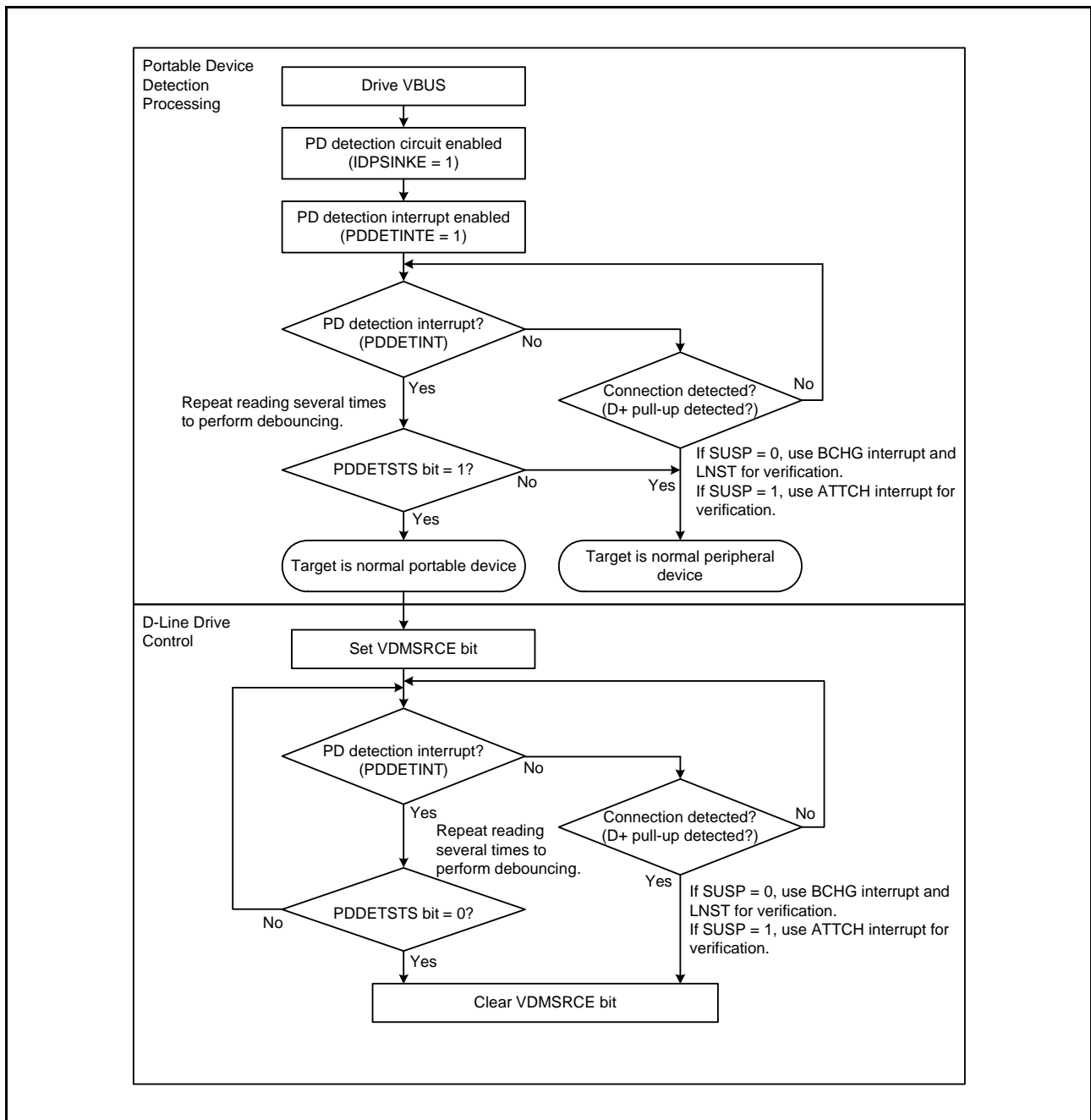


Figure 26.21 Process Flow for Operating as Charging Downstream Port (Steps (1) to (4))

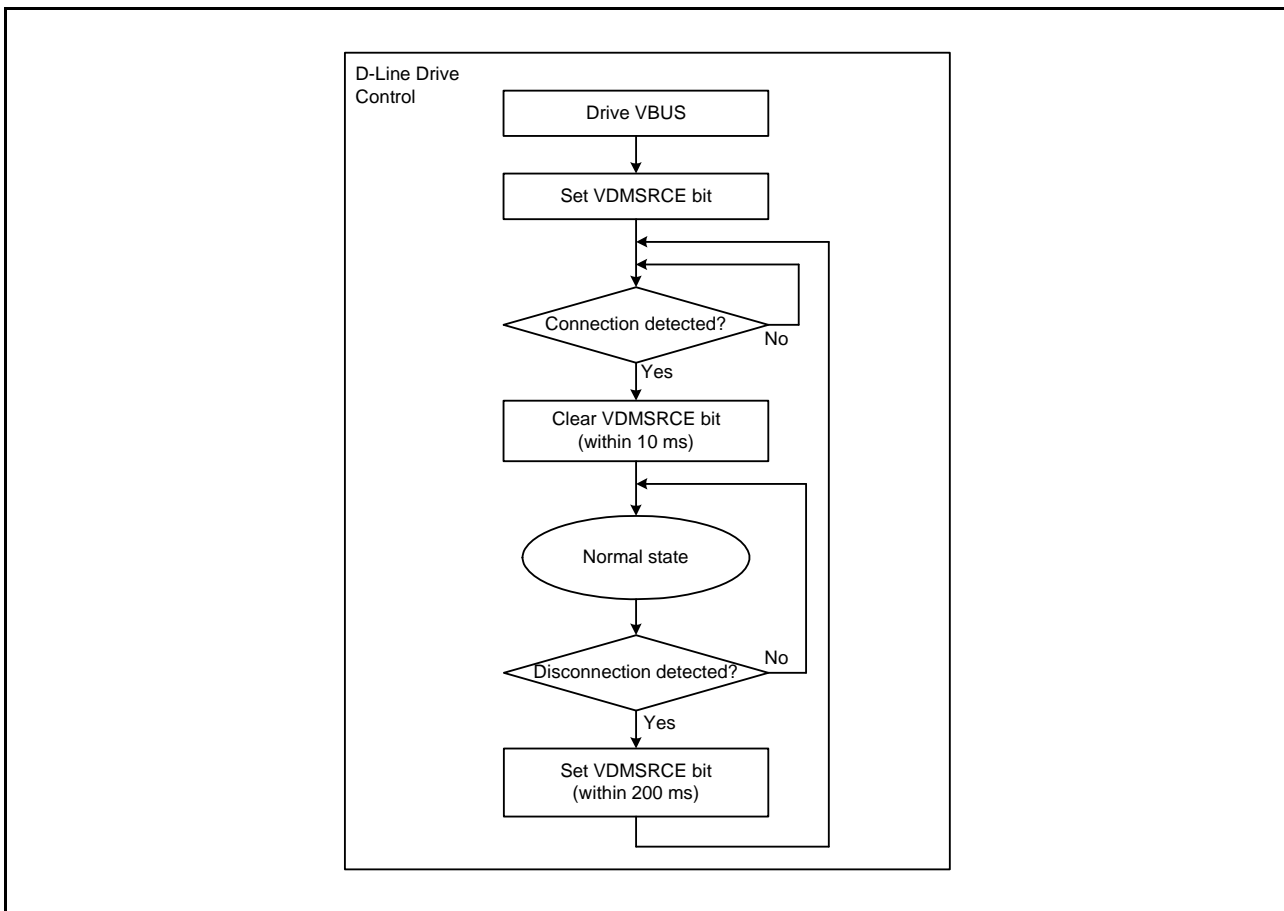


Figure 26.22 Process Flow for Operating as Charging Downstream Port (Steps (A) to (B))



## 27. Serial Communications Interface (SCIE, SCIF)

This MCU has eight independent serial communications interface (SCI) channels. The SCI consists of the SCIE module (SCI0, SCI1, SCI2, SCI5, SCI6, SCI8, and SCI9) and the SCIF module (SCI12).

The SCIE module (SCI0, SCI1, SCI2, SCI5, SCI6, SCI8, and SCI9) can handle both asynchronous and clock synchronous serial communications.

Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). As an extended function in asynchronous communications mode, the SCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards). The SCI is also supports simple SPI interfaces, and simple I<sup>2</sup>C bus interfaces when configured for single-master systems.

The SCIF module includes the functions of the SCIE module, and supports an extended serial communication protocol formed of Start Frames and Information Frames.

In this section, "PCLK" is used to refer to PCLKB.

### 27.1 Overview

Table 27.1 lists the specifications of the SCIE module, Table 27.2 lists the specifications of the SCIF module, and Table 27.3 lists the specifications of the individual SCI channels.

Figure 27.1, Figure 27.2, and Figure 27.3 show block diagrams of the SCIE module, and Figure 27.4 shows the block diagram of the SCIF module.

**Table 27.1 SCIE Specifications (1/2)**

Item	Description	
Serial communication modes	<ul style="list-style-type: none"> <li>Asynchronous</li> <li>Clock synchronous</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 the on-chip baud rate generator.	
Full-duplex communications	Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.	
I/O pins	See Table 27.4 to Table 27.6.	
Data transfer	Selectable as LSB first or MSB first transfer* <sup>1</sup>	
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)	
Low power consumption 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 parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTS# and RTS# pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the MTU can be used.(SCI1, SCI5)
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.

**Table 27.1 SCIE Specifications (2/2)**

Item	Description	
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTS# and RTS# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I <sup>2</sup> C mode	Transfer format	I <sup>2</sup> C bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (see section 27.2.9, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Event link function (supported by SCI5 only)	Error (receive error or error signal detection) event output	
	Receive data full event output	
	Transmit data empty event output	
	Transmit end event output	

Note 1. In simple I<sup>2</sup>C mode, only MSB first is available.

**Table 27.2 SCIF Specifications (1/2)**

Item	Description
Serial communication modes	<ul style="list-style-type: none"> <li>• Asynchronous</li> <li>• Clock synchronous</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 the on-chip baud rate generator.
Full-duplex communications	Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins	See Table 27.4 to Table 27.7.
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)
Low power consumption function	Module stop state can be set.

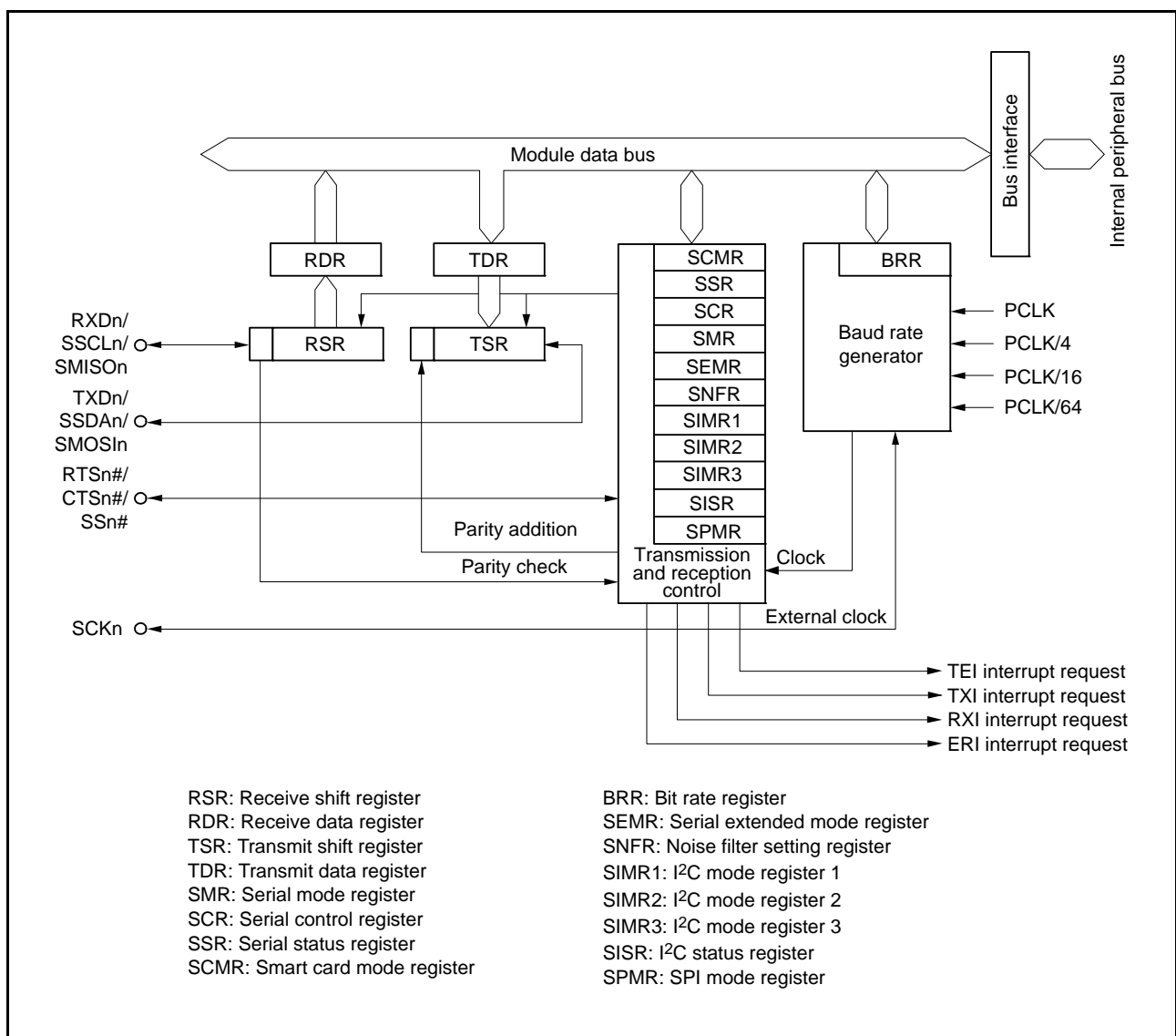
**Table 27.2 SCIf Specifications (2/2)**

Item	Description	
Asynchronous mode	Data length	7 or 8 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the MTU can be used. (SCI12)
	Multi-processor communications function	Serial communication among multiple processors
Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.	
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I <sup>2</sup> C mode	Transfer format	I <sup>2</sup> C bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (see section 27.2.9, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Extended serial mode	Start Frame transmission	<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 the RXDX12 signal</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 SCIE 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 27.3 Functions of SCI Channels**

Item	SCI0, SCI2, SCI6, SCI8, SCI9	SCI1	SCI5	SCI12
Asynchronous mode	Available	Available	Available	Available
Clock synchronous mode	Available	Available	Available	Available
Smart card interface mode	Available	Available	Available	Available
Simple I <sup>2</sup> C mode	Available	Available	Available	Available
Simple SPI mode	Available	Available	Available	Available
Extended serial mode	Not available	Not available	Not available	Available
MTU clock input	Not available	Available	Available	Available
Event link function	Not available	Not available	Available	Not available



**Figure 27.1 Block Diagram of SCIe (SCI0, SCI2, SCI6, SCI8, and SCI9)**

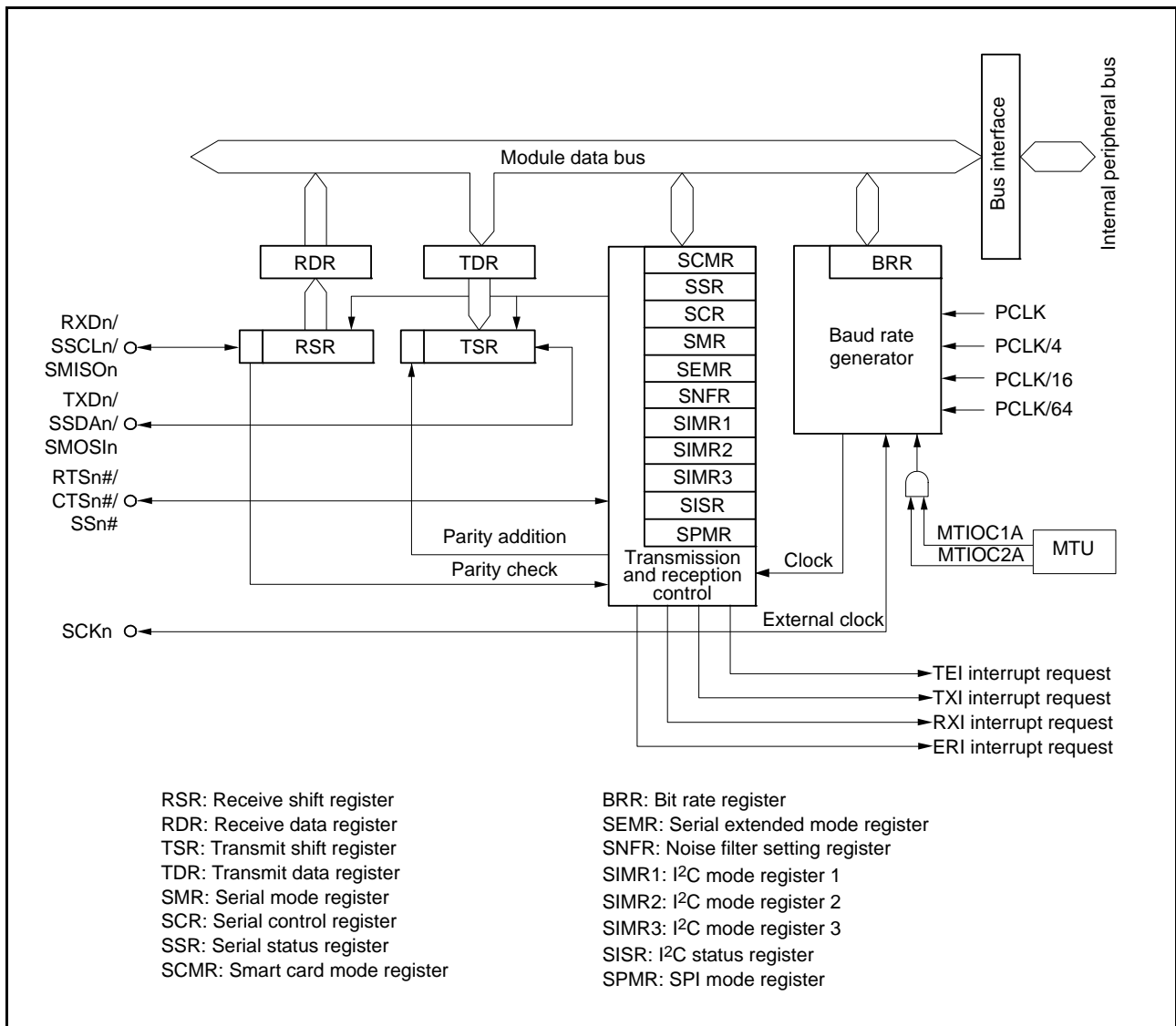


Figure 27.2 Block Diagram of SCIE (SCI1)

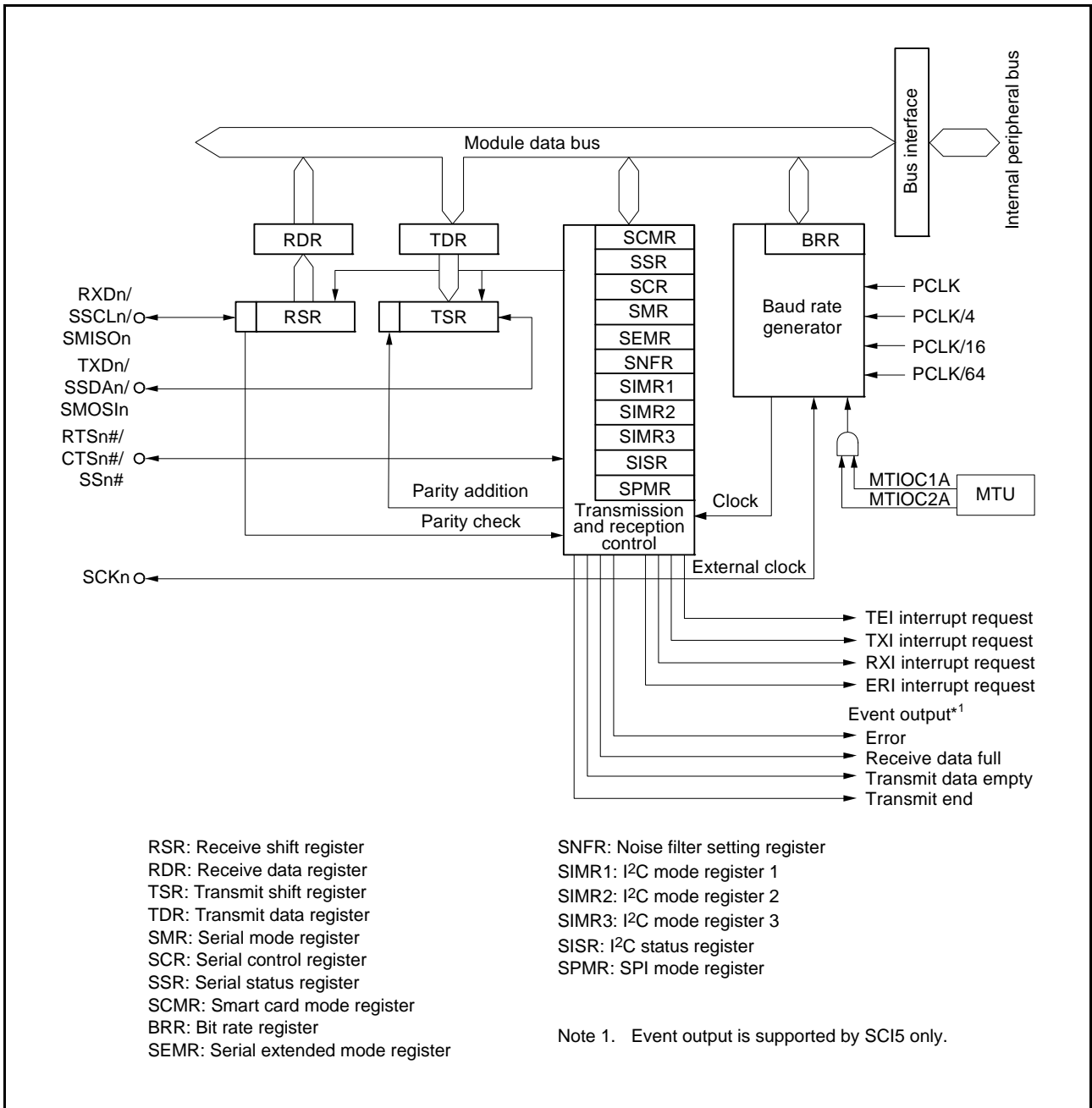


Figure 27.3 Block Diagram of SCIE (SCI5)

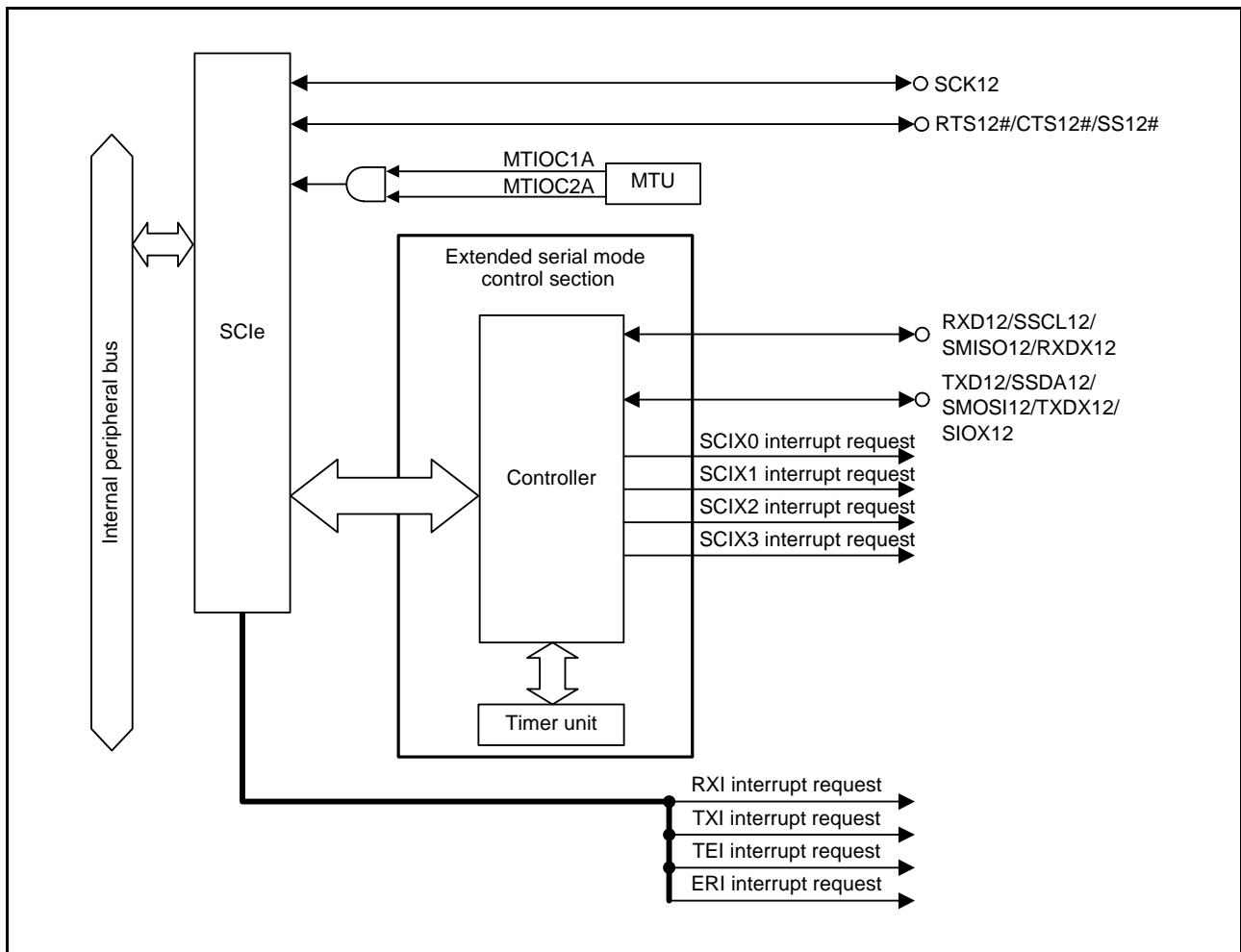


Figure 27.4 Block Diagram of SCIf (SCI12)

Table 27.4 to Table 27.7 list the pin configuration of the SCIs for the individual modes.

**Table 27.4 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode**

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
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
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
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 27.5 SCI Pin Configuration in Simple I<sup>2</sup>C Mode**

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
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
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
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 27.6 SCI Pin Configuration in 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
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
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

**Table 27.6 SCI Pin Configuration in Simple SPI Mode (2/2)**

Channel	Pin Name	I/O	Function
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
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 27.7 SCI Pin Configuration in Extended Serial Mode**

Channel	Pin Name	I/O	Function
SCI12	RDX12	Input	SCI12 receive data input
	TXDX12	Output	SCI12 transmit data output
	SIOX12	I/O	SCI12 transfer data input/output

## 27.2 Register Descriptions

### 27.2.1 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 automatically transferred to the RDR register. The RSR register cannot be directly accessed by the CPU.

### 27.2.2 Receive Data Register (RDR)

Address(es): SCI0.RDR 0008 A005h, SCI1.RDR 0008 A025h, SCI2.RDR 0008 A045h, SCI5.RDR 0008 A0A5h, SCI6.RDR 0008 A0C5h, SCI8.RDR 0008 A105h, SCI9.RDR 0008 A125h, SCI12.RDR 0008 B305h



RDR is an 8-bit register that stores receive data.

When one frame of serial data has been received, the received serial data is transferred from RSR to RDR. Then the RSR register can 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.

### 27.2.3 Transmit Data Register (TDR)

Address(es): SCI0.TDR 0008 A003h, SCI1.TDR 0008 A023h, SCI2.TDR 0008 A043h, SCI5.TDR 0008 A0A3h, SCI6.TDR 0008 A0C3h, SCI8.TDR 0008 A103h, SCI9.TDR 0008 A123h, SCI12.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 transmit data to TDR once after each instance of the transmit data empty interrupt (TXI).

### 27.2.4 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.

## 27.2.5 Serial Mode Register (SMR)

Note: Some bits in SMR have different functions in smart card interface mode and non-smart card interface mode.

### (1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SC10.SMR 0008 A000h, SC11.SMR 0008 A020h, SC12.SMR 0008 A040h, SC15.SMR 0008 A0A0h, SC16.SMR 0008 A0C0h, SC18.SMR 0008 A100h, SC19.SMR 0008 A120h, SC112.SMR 0008 B300h

b7	b6	b5	b4	b3	b2	b1	b0
CM	CHR	PE	PM	STOP	MP	CKS[1:0]	

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK 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) 0: Selects even parity 1: Selects odd parity	R/W*4
b5	PE	Parity Enable	(Valid only in asynchronous mode) • When transmitting 0: Parity bit addition is not performed 1: The parity bit is added • When receiving 0: Parity bit checking is not performed 1: The parity bit is checked	R/W*4
b6	CHR	Character Length	(Valid only in asynchronous mode) 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 or simple SPI mode	R/W*4

Note 1. n is the decimal notation of the value of n in BRR (refer to section 27.2.9, Bit Rate Register (BRR)).

Note 2. In other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.

Note 3. LSB first is fixed and the MSB (bit 7) in 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, refer to section 27.2.9, 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)**

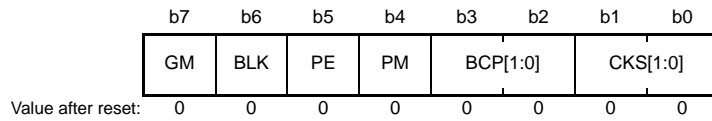
When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.  
Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

**CHR Bit (Character Length)**

Selects the data length for transmission and reception.  
In other than asynchronous mode, a fixed data length of 8 bits is used.

## (2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SC10.SMR 0008 A000h, SC11.SMR 0008 A020h, SC12.SMR 0008 A040h, SC15.SMR 0008 A0A0h,  
SC16.SMR 0008 A0C0h, SC18.SMR 0008 A100h, SC19.SMR 0008 A120h, SC112.SMR 0008 B300h



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 SCMR.BCP2 bit. Table 27.8 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*3
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*3
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*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 (refer to section 27.2.9, Bit Rate Register (BRR)).

Note 2. S is the value of S in BRR (refer to section 27.2.9, 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, refer to section 27.2.9, 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 SCMR.BCP2 bit.

For details, refer to section 27.6.4, Receive Data Sampling Timing and Reception Margin.

**Table 27.8 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits**

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in BRR (refer to section 27.2.9, Bit Rate Register (BRR)).

**PM Bit (Parity Mode)**

Selects the parity mode for transmission and reception (even or odd).

For details on the usage of this bit in smart card interface mode, refer to section 27.6.2, Data Format (Except in Block Transfer Mode).

**PE Bit (Parity Enable)**

Set the PE bit to 1.

The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

**BLK Bit (Block Transfer Mode)**

Setting this bit to 1 allows block transfer mode operation.

For details, refer to section 27.6.3, Block Transfer Mode.

**GM Bit (GSM Mode)**

Setting this bit to 1 allows GSM mode operation.

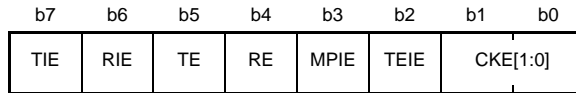
In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, refer to section 27.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 27.6.8, Clock Output Control.

## 27.2.6 Serial Control Register (SCR)

Note: Some bits in SCR have different functions in smart card interface mode and non-smart card interface mode.

### (1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI0.SCR 0008 A002h, SCI1.SCR 0008 A022h, SCI2.SCR 0008 A042h, SCI5.SCR 0008 A0A2h,  
SCI6.SCR 0008 A0C2h, SCI8.SCR 0008 A102h, SCI9.SCR 0008 A122h, 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, SCI2, SCI6, SCI8, and SCI9 (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 SCI1, SCI5, 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 is available for use as an I/O port according to the I/O port settings.</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 MTU clock                   <ul style="list-style-type: none"> <li>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 MTU 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
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



Bit	Symbol	Bit Name	Description	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0, while the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. While the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

### CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.

The combination of the settings of these bits and of the SEMR.ACS0 bit sets the internal MTU clock.

### TEIE Bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.

A TEI interrupt request is disabled by setting the TEIE bit to 0.

In simple I<sup>2</sup>C mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI).

In this case, the TEIE bit can be used to enable or disable the STI.

### MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags 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, refer to section 27.4, 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 setting 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)**

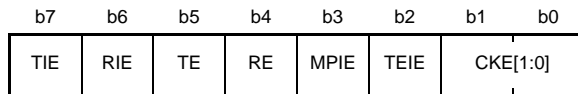
Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in SSR and then setting the flag to 0, or setting the RIE bit to 0.

## (2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SCI0.SCR 0008 A002h, SCI1.SCR 0008 A022h, SCI2.SCR 0008 A042h, SCI5.SCR 0008 A0A2h,  
SCI6.SCR 0008 A0C2h, SCI8.SCR 0008 A102h, SCI9.SCR 0008 A122h, 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 SMR.GM = 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 according to the I/O port settings.)</li> <li>0 1: Clock output</li> <li>1 x: (Setting prohibited)</li> </ul> </li> <li>When SMR.GM = 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. 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE.

For details on interrupt requests, refer to section 27.11, 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, refer to section 27.6.8, 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 setting 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 setting the RIE bit to 0.

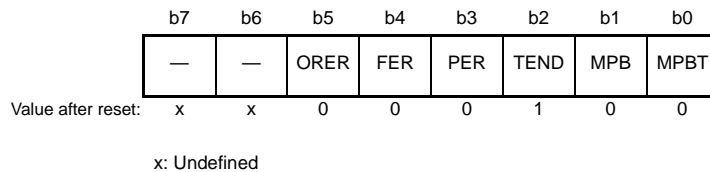
An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in SSR and then setting the flag to 0, or setting the RIE bit to 0.

## 27.2.7 Serial Status Register (SSR)

Note: Some bits in SSR have different functions in smart card interface mode and non-smart card interface mode.

### (1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI0.SSR 0008 A004h, SCI1.SSR 0008 A024h, SCI2.SSR 0008 A044h, SCI5.SSR 0008 A0A4h,  
SCI6.SSR 0008 A0C4h, SCI8.SSR 0008 A104h, SCI9.SSR 0008 A124h, SCI12.SSR 0008 B304h



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 SCR.RE bit is 0.

#### TEND Flag (Transmit End Flag)

Indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 (serial transmission is disabled)  
When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1  
When setting the TEND flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

**PER Flag (Parity Error Flag)**

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception

Although receive data when the parity error occurs is transferred to 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

When setting the PER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

**FER Flag (Framing Error Flag)**

Indicates that a framing error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When the stop bit is 0

In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. Note that although receive data when the framing error occurs is transferred to 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

When setting the FER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

Even when the SCR.RE bit is set to 0, the FER flag is not affected and retains its previous value.

**ORER Flag (Overrun Error Flag)**

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from 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 0 is written to ORER after reading ORER = 1

When setting the ORER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

## (2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SC10.SSR 0008 A004h, SC11.SSR 0008 A024h, SC12.SSR 0008 A044h, SC15.SSR 0008 A0A4h,  
SC16.SSR 0008 A0C4h, SC18.SSR 0008 A104h, SC19.SSR 0008 A124h, 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.

**TEND Flag (Transmit End Flag)**

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR.TE bit = 0 (serial transmission is disabled)  
When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated  
The set timing is determined by register settings as listed below.  
When SMR.GM = 0 and SMR.BLK = 0, 12.5 etu after the start of transmission  
When SMR.GM = 0 and SMR.BLK = 1, 11.5 etu after the start of transmission  
When SMR.GM = 1 and SMR.BLK = 0, 11.0 etu after the start of transmission  
When SMR.GM = 1 and SMR.BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1  
When setting the TEND flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

**PER Flag (Parity Error Flag)**

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception  
Although receive data when the parity error occurs is transferred to 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

When setting the PER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

### **ERS Flag (Error Signal Status Flag)**

[Setting condition]

- When a low error signal is sampled

[Clearing condition]

- When 0 is written to ERS after reading ERS = 1

When setting the ERS flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

Even when the SCR.RE bit is set to 0, the ERS flag is not affected and retains its previous value.

### **ORER Flag (Overrun Error Flag)**

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from 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

When setting the ORER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.



## 27.2.8 Smart Card Mode Register (SCMR)

Address(es): SCI0.SCMR 0008 A006h, SCI1.SCMR 0008 A026h, SCI2.SCMR 0008 A046h, SCI5.SCMR 0008 A0A6h, SCI6.SCMR 0008 A0C6h, SCI8.SCMR 0008 A106h, SCI9.SCMR 0008 A126h, SCI12.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: Non-smart card interface mode (Asynchronous mode, clock synchronous mode, simple SPI mode, or simple I <sup>2</sup> C 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	This bit can be used in the following modes. <ul style="list-style-type: none"> <li>• Smart card interface mode</li> <li>• Asynchronous mode (multi-processor mode)</li> <li>• Clock synchronous mode</li> <li>• Simple SPI mode</li> </ul> Set this bit to 1 if operation is to be in simple I <sup>2</sup> C mode. 0: Transfer with LSB first 1: Transfer with MSB first	R/W*1
b6 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	BCP2	Base Clock Pulse 2	Selects the number of base clock cycles in combination with the SMR.BCP[1:0] bits. Table 27.9 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*1

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

### SMIF Bit (Smart Card Interface Mode Select)

When this bit is set to 1, smart card interface mode is selected.

When this bit is set to 0, non-smart card interface mode, i.e., asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode, or simple I<sup>2</sup>C 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.

### BCP2 Bit (Base Clock Pulse 2)

Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR.BCP[1:0] bits.

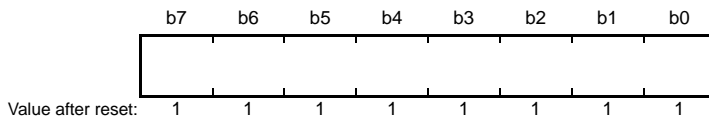
**Table 27.9 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits**

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)* <sup>1</sup>
0	0	1	128 clock cycles (S = 128)* <sup>1</sup>
0	1	0	186 clock cycles (S = 186)* <sup>1</sup>
0	1	1	512 clock cycles (S = 512)* <sup>1</sup>
1	0	0	32 clock cycles (S = 32)* <sup>1</sup> (Initial Value)
1	0	1	64 clock cycles (S = 64)* <sup>1</sup>
1	1	0	372 clock cycles (S = 372)* <sup>1</sup>
1	1	1	256 clock cycles (S = 256)* <sup>1</sup>

Note 1. S is the value of S in BRR (refer to section 27.2.9, Bit Rate Register (BRR)).

### 27.2.9 Bit Rate Register (BRR)

Address(es): SCI0.BRR 0008 A001h, SCI1.BRR 0008 A021h, SCI2.BRR 0008 A041h, SCI5.BRR 0008 A0A1h, SCI6.BRR 0008 A0C1h, SCI8.BRR 0008 A101h, SCI9.BRR 0008 A121h, 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 27.10 shows the relationship 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, but it can be written to only when the TE and RE bits in SCR are 0.

**Table 27.10 Relationship between N Setting in BRR and Bit Rate B**

Mode	SEMR.ABCS Bit	BRR Setting	Error
Asynchronous, multi-processor transfer	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1 \right\} \times 100$
	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N + 1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI		$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface		$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N + 1)} - 1 \right\} \times 100$
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 on-chip baud rate generator (0 ≤ N ≤ 255)

PCLK: Operating frequency (MHz)

n and S: Determined by the settings of the SMR and SCMR registers as listed in the table below.

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 27.11 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)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$
	Width at low level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$

**Table 27.12 Clock Source Settings**

SMR.CKS[1:0] Bit Setting	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 27.13 Base Clock Settings in Smart Card Interface Mode**

SCMR.BCP2 Bit Setting	SMR.BCP[1:0] Bit Setting	Base Clock Cycles for 1-bit Period	S
0	0 0	93 clock cycles	93
0	0 1	128 clock cycles	128
0	1 0	186 clock cycles	186
0	1 1	512 clock cycles	512
1	0 0	32 clock cycles	32
1	0 1	64 clock cycles	64
1	1 0	372 clock cycles	372
1	1 1	256 clock cycles	256

Table 27.14 lists examples of N settings in BRR in normal asynchronous mode. Table 27.15 lists the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in clock synchronous mode and simple SPI mode are listed in Table 27.18. Examples of BRR (N) settings in smart card interface mode are listed in Table 27.20. Examples of BRR (N) settings in simple I<sup>2</sup>C mode are listed in Table 27.23. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 27.6.4, Receive Data Sampling Timing and Reception Margin. Table 27.16 and Table 27.19 list the maximum bit rates with external clock input.

When either 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 becomes twice that listed in Table 27.14.

Table 27.14 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	-0.26	2	177	-0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	25	0.16	0	31	0.00	0	32	-1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	-2.34	0	19	0.00
31250	0	7	0.00	0	9	-1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	-2.34	0	9	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	-0.17	3	70	0.03	3	75	0.48	3	79	-0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00
9600	0	45	-0.93	0	51	0.16	0	55	0.00	0	58	-0.69	0	63	0.00
19200	0	22	-0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	-1.70
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	-2.34	0	15	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)								
	20			25			30		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	-0.25	3	110	-0.02	3	132	0.13
150	3	64	0.16	3	80	0.47	3	97	-0.35
300	2	129	0.16	2	162	-0.15	2	194	0.16
600	2	64	0.16	2	80	0.47	2	97	-0.35
1200	1	129	0.16	1	162	-0.15	1	194	0.16
2400	1	64	0.16	1	80	0.47	1	97	-0.35
4800	0	129	0.16	0	162	-0.15	0	194	0.16
9600	0	64	0.16	0	80	0.47	0	97	-0.35
19200	0	32	-1.36	0	40	-0.76	0	48	-0.35
31250	0	19	0.00	0	24	0.00	0	29	0
38400	0	15	1.73	0	19	1.73	0	23	1.73

Note: This is an example when the ABCS bit is 0.  
When the ABCS bit is set to 1, the bit rate doubles.

**Table 27.15 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

Note: When the ABCS bit in SEMR is set to 1, the bit rate is two times.

**Table 27.16 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

**Table 27.17 Maximum Bit Rate with MTU Clock Input (Asynchronous Mode)**

PCLK (MHz)	MTU Clock (MHz)	Maximum Bit Rate (bps)	
		SEMR.ABCS Bit = 0	SEMR.ABCS Bit = 1
8	4	250000	500000
9.8304	4.9152	307200	614400
10	5	312500	625000
12	6	375000	750000
12.288	6.144	384000	768000
14	7	437500	875000
16	8	500000	1000000
17.2032	8.6016	537600	1075200
18	9	562500	1125000
19.6608	9.8304	614400	1228800
20	10	625000	1250000
25	12.5	781250	1562500
30	15	937500	1875000

**Table 27.18 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	
	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
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187
5 k	1	99	1	124	1	199	1	249	2	77	2	93
10 k	0	199	0	249	1	99	1	124	1	155	1	187
25 k	0	79	0	99	0	159	0	199	0	249	1	74
50 k	0	39	0	49	0	79	0	99	0	124	0	149
100 k	0	19	0	24	0	39	0	49	0	62	0	74
250 k	0	7	0	9	0	15	0	19	0	24	0	29
500 k	0	3	0	4	0	7	0	9	—	—	0	14
1 M	0	1			0	3	0	4	—	—	—	—
2 M	0	0 <sup>*1</sup>	—	—	0	1	—	—	—	—	—	—
2.5 M			0	0 <sup>*1</sup>			0	1	—	—	0	2
4 M					0	0 <sup>*1</sup>	—	—	—	—	—	—
5 M							0	0 <sup>*1</sup>	—	—	—	—
6.25 M									0	0 <sup>*1</sup>	—	—
7.5 M											0	0 <sup>*1</sup>

Space: Setting prohibited.

—: Can be set, but an error will occur.

Note 1. Continuous transmission or reception is impossible. After transmitting/receiving one frame of data, there is an interval of a 1-bit period before starting transmitting/receiving the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. For this reason, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is  $\frac{8}{9}$  times the bit rate.

**Table 27.19 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)**

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)
8	1.3333	1.3333
10	1.6667	1.6667
12	2.0000	2.0000
14	2.3333	2.3333
16	2.6667	2.6667
18	3.0000	3.0000
20	3.3333	3.3333
25	4.1667	4.1667
30	5.0000	5.0000

**Table 27.20 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372)**

Bit Rate (bps)	PCLK (MHz)	n	N	Error (%)
9600	7.1424	0	0	0.00
	10.00	0	1	30
	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

**Table 27.21 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)**

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0

**Table 27.22 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



Table 27.23 BRR Settings for Various Bit Rates (Simple I<sup>2</sup>C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	31	-2.3	1	12	-3.8	1	15	-2.3	1	19	-2.3
25 k	0	9	0.0	0	12	-3.8	1	4	0.0	1	6	-10.7	1	7	-2.3
50 k	0	4	0.0	0	6	-10.7	1	2	-16.7	1	3	-21.9	1	3	-2.3
100 k	0	2	-16.7	0	3	-21.9	0	4	0.0	0	6	-10.7	1	1	-2.3
250 k	0	0	0.0	0	1	-37.5	0	1	0.0	0	2	-16.7	0	3	-21.9
350 k										0	1	-10.7	0	2	-25.6

Bit Rate (bps)	Operating Frequency PCLK (MHz)		
	30		
	n	N	Error (%)
10 k	1	23	-2.3
25 k	1	9	-6.3
50 k	1	4	-6.3
100 k	1	2	-21.9
250 k	0	3	-6.3
350 k	0	2	-10.7

Table 27.24 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I<sup>2</sup>C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	8			10			16			20		
	n	N	Min. Widths at High/Low Level for SCL (μs)	n	N	Min. Widths at High/Low Level for SCL (μs)	n	N	Min. Widths at High/Low Level for SCL (μs)	n	N	Min. Widths at High/Low Level for SCL (μs)
10 k	0	24	43.75/50.00	0	31	44.80/51.20	1	12	45.5/52.00	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

Bit Rate (bps)	Operating Frequency PCLK (MHz)					
	25			30		
	n	N	Min. Widths at High/Low Level for SCL (μs)	n	N	Min. Widths at High/Low Level for SCL (μs)
10 k	1	19	44.80/51.20	1	23	44.80/51.20
25 k	1	7	17.92/20.48	1	9	18.66/21.33
50 k	1	3	8.96/10.24	1	4	9.33/10.66
100 k	1	1	4.48/5.12	1	2	5.60/6.40
250 k	0	3	2.24/2.56	0	3	1.86/2.13
350 k	0	2	1.68/1.92	0	2	1.40/1.60

## 27.2.10 Serial Extended Mode Register (SEMR)

Address(es): SCI0.SEMR 0008 A007h, SCI1.SEMR 0008 A027h, SCI2.SEMR 0008 A047h, SCI5.SEMR 0008 A0A7h, SCI6.SEMR 0008 A0C7h, SCI8.SEMR 0008 A107h, SCI9.SEMR 0008 A127h, SCI12.SEMR 0008 B307h

b7	b6	b5	b4	b3	b2	b1	b0
RXDESEL	—	NFEN	ABCS	—	—	—	ACS0

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

Bit	Symbol	Bit Name	Description	R/W												
b0	ACS0	Asynchronous Mode Clock Source Select	(Valid only in asynchronous mode) 0: External clock input 1: Logical AND of two clock cycles output from the MTU	R/W*1												
			<table border="1"> <thead> <tr> <th>SCI</th> <th>MTU</th> <th>Compare Match Output</th> </tr> </thead> <tbody> <tr> <td>SCI1</td> <td>MTU1, MTU2</td> <td>MTIOC1A, MTIOC2A</td> </tr> <tr> <td>SCI5</td> <td>MTU1, MTU2</td> <td>MTIOC1A, MTIOC2A</td> </tr> <tr> <td>SCI12</td> <td>MTU1, MTU2</td> <td>MTIOC1A, MTIOC2A</td> </tr> </tbody> </table>	SCI	MTU	Compare Match Output	SCI1	MTU1, MTU2	MTIOC1A, MTIOC2A	SCI5	MTU1, MTU2	MTIOC1A, MTIOC2A	SCI12	MTU1, MTU2	MTIOC1A, MTIOC2A	
SCI	MTU	Compare Match Output														
SCI1	MTU1, MTU2	MTIOC1A, MTIOC2A														
SCI5	MTU1, MTU2	MTIOC1A, MTIOC2A														
SCI12	MTU1, MTU2	MTIOC1A, MTIOC2A														
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												
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W												
b7	RXDESEL	Asynchronous Start Bit Edge Detection Select	(Valid only in asynchronous mode) 0: The low level on the RXDn pin is detected as the start bit. 1: A falling edge on the RXDn pin is detected as the start bit.	R/W*1												

Note 1. Writable only when 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.

The MTIOC1A and MTIOC2A output of the MTU can be set as the serial transfer base clock.

Figure 27.5 shows a setting example when the MTU clock input 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 MTU clock input can be selected.

Set the ACS0 bit to 0 in other than asynchronous mode.

### 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.

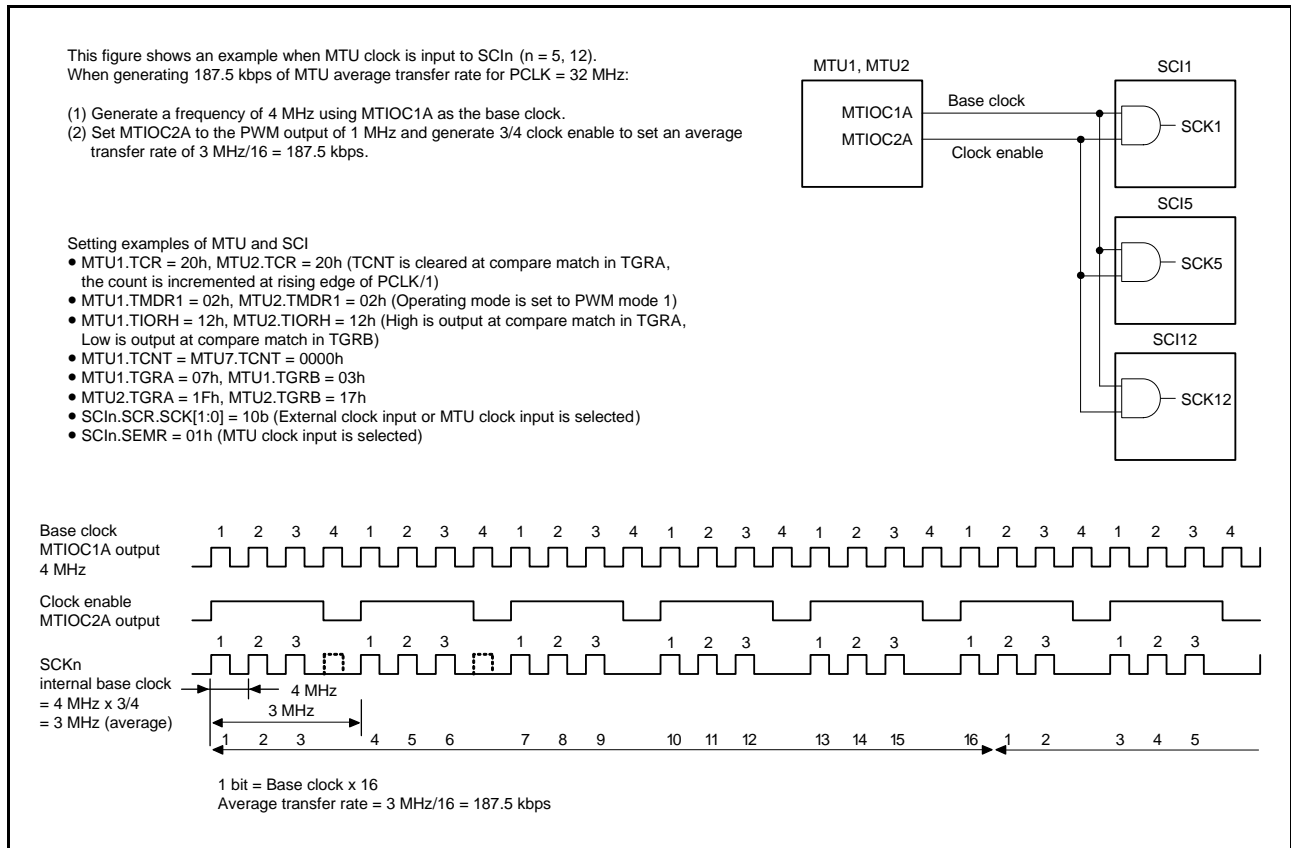
In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function.

When the function is disabled, input signals are transferred as is, as internal signals.

**RXDESEL Bit (Asynchronous Start Bit Edge Detection Select)**

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.

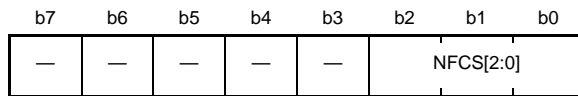
Set this bit to 0 in modes other than asynchronous mode.



**Figure 27.5 Example of Average Transfer Rate Setting When MTU Clock is Input**

### 27.2.11 Noise Filter Setting Register (SNFR)

Address(es): SC10.SNFR 0008 A008h, SC11.SNFR 0008 A028h, SC12.SNFR 0008 A048h, SC15.SNFR 0008 A0A8h, SC16.SNFR 0008 A0C8h, SC18.SNFR 0008 A108h, SC19.SNFR 0008 A128h, SC112.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	<p>In asynchronous mode, the standard setting for the base clock is as follows.</p> <p>b2 b0 0 0 0: The clock signal divided by 1 is used with the noise filter.</p> <p>In simple I<sup>2</sup>C mode, the standard settings for the clock source of the on-chip baud rate generator selected by the SMR.CKS[1:0] bits are given below.</p> <p>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.</p> <p>Settings other than above are prohibited.</p>	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

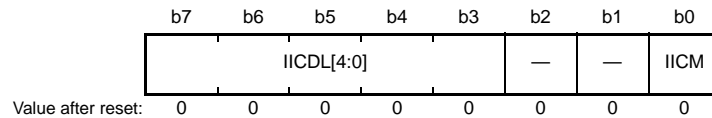
Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (serial reception and transmission disabled).

#### NFCS[2:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter. To use the noise filter in asynchronous mode, set these bits to 000b. In simple I<sup>2</sup>C mode, set the bits to a value in the range from 001b to 100b.

## 27.2.12 I<sup>2</sup>C Mode Register 1 (SIMR1)

Address(es): SCI0.SIMR1 0008 A009h, SCI1.SIMR1 0008 A029h, SCI2.SIMR1 0008 A049h, SCI5.SIMR1 0008 A0A9h, SCI6.SIMR1 0008 A0C9h, SCI8.SIMR1 0008 A109h, SCI9.SIMR1 0008 A129h, 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: Asynchronous mode, Multi-processor mode, Clock synchronous mode (in asynchronous mode, 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 Output Delay Select	(Cycles below are of the clock signal from the on-chip baud rate generator.) b7      b3 0 0 0 0 0: No output delay 0 0 0 0 1: 0 to 1 cycle 0 0 0 1 0: 1 to 2 cycles 0 0 0 1 1: 2 to 3 cycles 0 0 1 0 0: 3 to 4 cycles 0 0 1 0 1: 4 to 5 cycles :            : 1 1 1 1 0: 29 to 30 cycles 1 1 1 1 1: 30 to 31 cycles	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (both serial transmission and reception are disabled).

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.

### 27.2.13 I<sup>2</sup>C Mode Register 2 (SIMR2)

Address(es): SC10.SIMR2 0008 A00Ah, SC11.SIMR2 0008 A02Ah, SC12.SIMR2 0008 A04Ah, SC15.SIMR2 0008 A0AAh, SC16.SIMR2 0008 A0CAh, SC18.SIMR2 0008 A10Ah, SC19.SIMR2 0008 A12Ah, SC112.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 (serial reception and transmission disabled).

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 pin has been placed at the low level in the case of a wait inserted by the other device, etc.

The SSCLn clock signal is not synchronized if the IICCSC bit is 0. The SSCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SSCLn pin.

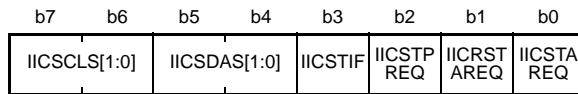
Set the IICCSC bit to 1 except during debugging.

#### IICACKT Bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

### 27.2.14 I<sup>2</sup>C Mode Register 3 (SIMR3)

Address(es): SCI0.SIMR3 0008 A00Bh, SCI1.SIMR3 0008 A02Bh, SCI2.SIMR3 0008 A04Bh, SCI5.SIMR3 0008 A0ABh, SCI6.SIMR3 0008 A0CBh, SCI8.SIMR3 0008 A10Bh, SCI9.SIMR3 0008 A12Bh, 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, *5	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*2, *3, *4, *5	R/W
b2	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*2, *3, *4, *5	R/W
b3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag	0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R/W
b5, b4	IICSDAS[1:0]	SSDA Output Select	b5 b4 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSDAn pin. 1 1: Place the SSDAn pin in the high-impedance state.	R/W
b7, b6	IICSCLS[1:0]	SSCL Output Select	b7 b6 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSCLn pin. 1 1: Place the SSCLn pin in the high-impedance state.	R/W

Note 1. 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.

Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

SIMR3 is used to control the simple 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 generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR.TEIE bit, an STI request is output.

[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the other condition takes precedence)

[Clearing conditions]

- Writing 0 to the bit (confirm that the IICSTIF flag is 0 before doing so)
- Writing 0 to the SIMR1.IICM bit (when operation is not in simple I<sup>2</sup>C mode)
- Writing 0 to the SCR.TE bit

**IICSDAS[1:0] Bits (SSDA Output Select)**

These bits control output from the SSDAn pin.

Set the IICSDAS[1:0] and IICSCLS[1:0] bits to the same value during normal operations.

**IICSCLS[1:0] Bits (SSCL Output Select)**

These bits control output from the SSCLn pin.

Set the IICSCLS[1:0] and IICSDAS[1:0] bits to the same value during normal operations.



### 27.2.15 I<sup>2</sup>C Status Register (SISR)

Address(es): SCI0.SISR 0008 A00Ch, SCI1.SISR 0008 A02Ch, SCI2.SISR 0008 A04Ch, SCI5.SISR 0008 A0ACh, SCI6.SISR 0008 A0CCh, SCI8.SISR 0008 A10Ch, SCI9.SISR 0008 A12Ch, SCI12.SISR 0008 B30Ch

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

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received	R/W*1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	—	Reserved	The read value is undefined.	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	—	Reserved	The read value is undefined.	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag.

SISR is used to monitor state in relation to simple I<sup>2</sup>C mode.

#### IICACKR Flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.

The IICACKR flag is updated at the rising of SSCLn clock for the ACK/NACK receiving bit.

## 27.2.16 SPI Mode Register (SPMR)

Address(es): SC10.SPMR 0008 A00Dh, SC11.SPMR 0008 A02Dh, SC12.SPMR 0008 A04Dh, SC15.SPMR 0008 A0ADh, SC16.SPMR 0008 A0CDh, SC18.SPMR 0008 A10Dh, SC19.SPMR 0008 A12Dh, SC12.SPMR 0008 B30Dh

b7	b6	b5	b4	b3	b2	b1	b0
CKPH	CKPOL	—	MFF	—	MSS	CTSE	SSE

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

Bit	Symbol	Bit Name	Description	R/W
b0	SSE	SSn# Pin Function Enable	0: SSn# pin function is disabled. 1: SSn# pin function is enabled.	R/W*1
b1	CTSE	CTS Enable	0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W*1
b2	MSS	Master Slave Select	0: Transmission is through the 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 (both serial transmission and reception are disabled).

Note 2. Only 0 can be written to these bits, which clears the flag.

SPMR is used to select the extension settings in asynchronous and clock synchronous modes.

### SSE Bit (SSn# Pin Function Enable)

Set this bit to 1 if the SSn# pin is to be used in control of transmission and reception (in simple SPI mode). Set this bit to 0 in any other mode. Furthermore, even for usage in simple SPI mode, the SSn# pin on the master side is not required to control reception and transmission when master mode (SCR.CKE[1:0] = 00b and MSS = 0) is selected and there is a single master, so the setting for the SSE bit is 0. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

### CTSE Bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for inputting of the CTS control signal to control of transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple I<sup>2</sup>C mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

### MSS Bit (Master Slave Select)

This bit selects between master and slave operation in simple SPI mode. The functions of the TXDn and RXDn pins are reversed when the MSS bit is set to 1, so that data is 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 SSn# pin being at the low level during master operation in simple SPI mode (SSE bit = 1 and MSS bit = 0)

[Clearing condition]

- Writing 0 to the bit after it was read as 1

**CKPOL Bit (Clock Polarity Select)**

This bit selects the polarity of the clock signal output through the SCKn pin. See Figure 27.57 for details.

Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

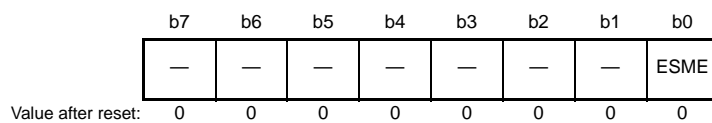
**CKPH Bit (Clock Phase Select)**

This bit selects the phase of the clock signal output through the SCKn pin. See Figure 27.57 for details.

Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

**27.2.17 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 27.25 Settings of the ESME Bit and Guaranteed Operation by Timer Operation Mode**

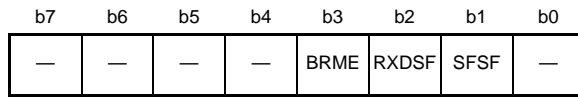
ESME Bit	Timer Mode	Break Field Low Width Determination Mode	Break Field Low Width Output Mode
0	○*1	×	×
1	○	○	○

○: Guarantee of operation is necessary. ×: Guarantee of operation is not necessary.

Note 1. Operation is only possible with PCLK selected.

### 27.2.18 Control Register 0 (CR0)

Address(es): SC112.CR0 0008 B321h

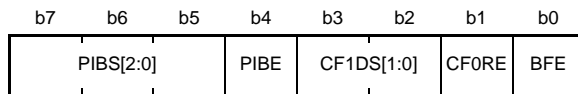


Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	SFSF	Start Frame Status Flag	0: Start Frame detection function is disabled. 1: Start Frame detection function is enabled.	R
b2	RXDSF	RXDX12 Input Status Flag	0: RXDX12 input is enabled. 1: RXDX12 input is disabled.	R
b3	BRME	Bit Rate Measurement Enable	0: Measurement of bit rate is disabled. 1: Measurement of bit rate is enabled.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

### 27.2.19 Control Register 1 (CR1)

Address(es): SC112.CR1 0008 B322h

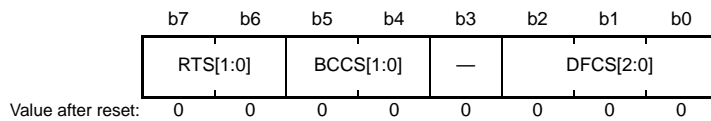


Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	BFE	Break Field Enable	0: Break Field detection is disabled. 1: Break Field detection is enabled.	R/W
b1	CF0RE	Control Field 0 Reception Enable	0: Reception of Control Field 0 is disabled. 1: Reception of Control Field 0 is enabled.	R/W
b3, b2	CF1DS[1:0]	Control Field 1 Data Register Select	b3 b2 0 0: Selects comparison with the value in 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

## 27.2.20 Control Register 2 (CR2)

Address(es): SCI12.CR2 0008 B323h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DFCS[2:0]	RXDX12 Signal Digital Filter Clock Select	b2 b0 0 0 0: Filter is disabled. 0 0 1: Filter clock is SCI base clock*1 0 1 0: Filter clock is PCLK/8 0 1 1: Filter clock is PCLK/16 1 0 0: Filter clock is PCLK/32 1 0 1: Filter clock is PCLK/64 1 1 0: Filter clock is PCLK/128 1 1 1: Setting prohibited	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	BCCS[1:0]	Bus Collision Detection Clock Select	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: The period of the SCI base clock is 1/16 of a single bit period when the SCI12.SEMR.ABCS is 0, and 1/8 of a single bit period when the SCI12.SEMR.ABCS is 1.

Note 1. To use the SCI base clock, set the SCI12.SCR.TE bit to 1.

### 27.2.21 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 Bit (Start Frame Detection Start)

Detection of a Start Frame begins when this bit is set to 1. The bit is read as 0.

### 27.2.22 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	TXDX12 Signal Polarity Select	0: The polarity of TXDX12 signal is not inverted for output. 1: The polarity of TXDX12 signal is inverted for output.	R/W
b1	RXDXPS	RXDX12 Signal Polarity Select	0: The polarity of RXDX12 signal is not inverted for input. 1: The polarity of RXDX12 signal is inverted for input.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SHARPS	TXDX12/RXDX12 Pin Multiplexing Select	0: The TXDX12 and RXDX12 pins are independent. 1: The TXDX12 and RXDX12 signals are multiplexed on the same pin.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### SHARPS Bit (TXDX12/RXDX12 Pin Multiplexing Select)

When this bit is set to 1, the TXDX12 and RXDX12 signals are multiplexed on the same pin so that half-duplex communications become possible.

### 27.2.23 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

## 27.2.24 Status Register (STR)

Address(es): SCI12.STR 0008 B327h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	AEDF	BCDF	PIBDF	CF1MF	CF0MF	BDFD

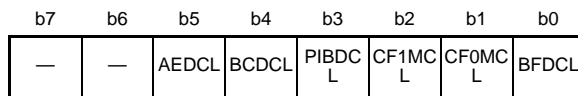
Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	BDFD	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. The write value should be 0.	R



### 27.2.25 Status Clear Register (STCR)

Address(es): SCI12.STCR 0008 B328h

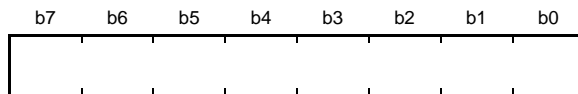


Value after reset: 0 0 0 0 0 0 0 0

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	PIBDCL	PIBDF Clear	Setting this bit to 1 clears the STR.PIBDF flag. This bit is read as 0.	R/W
b4	BCDCL	BCDF Clear	Setting this bit to 1 clears the STR.BCDF flag. This bit is read as 0.	R/W
b5	AEDCL	AEDF Clear	Setting this bit to 1 clears the STR.AEDF flag. This bit is read as 0.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

### 27.2.26 Control Field 0 Data Register (CF0DR)

Address(es): SCI12.CF0DR 0008 B329h

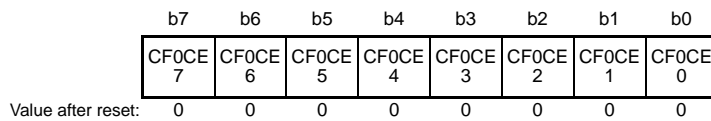


Value after reset: 0 0 0 0 0 0 0 0

CF0DR is an 8-bit readable and writable register that holds a value for comparison with Control Field 0.

### 27.2.27 Control Field 0 Compare Enable Register (CF0CR)

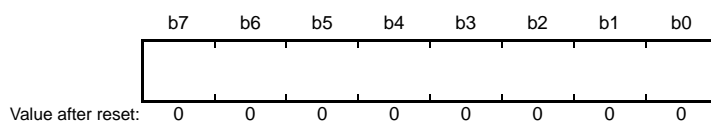
Address(es): SCI12.CF0CR 0008 B32Ah



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

### 27.2.28 Control Field 0 Receive Data Register (CF0RR)

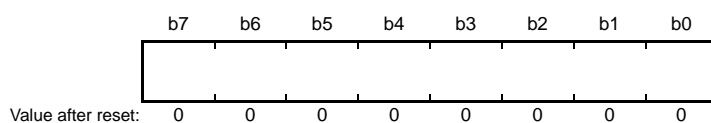
Address(es): SCI12.CF0RR 0008 B32Bh



CF0RR is a readable register that holds the value received in Control Field 0. Writing to this register from the CPU or DTC is not possible.

### 27.2.29 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.

### 27.2.30 Secondary Control Field 1 Data Register (SCF1DR)

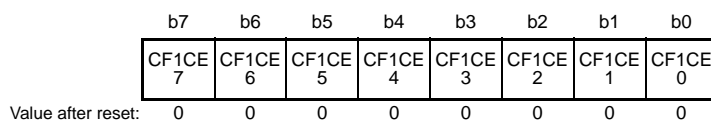
Address(es): SCI12.SCF1DR 0008 B32Dh



PCF1DR is an 8-bit readable and writable register that holds the 8-bit secondary value for comparison with Control Field 1.

### 27.2.31 Control Field 1 Compare Enable Register (CF1CR)

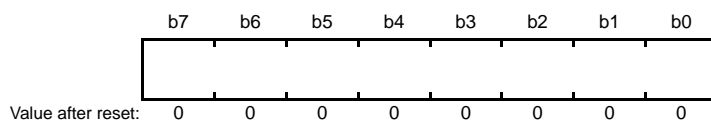
Address(es): SCI12.CF1CR 0008 B32Eh



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

### 27.2.32 Control Field 1 Receive Data Register (CF1RR)

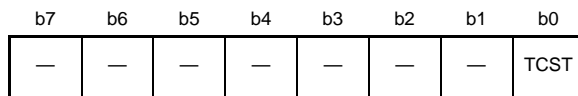
Address(es): SCI12.CF1RR 0008 B32Fh



CF1RR is a readable register that holds the value received in Control Field 1. Writing to this register from the CPU or DTC is not possible.

### 27.2.33 Timer Control Register (TCR)

Address(es): SC112.TCR 0008 B330h

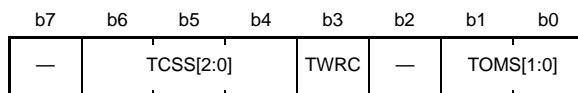


Value after reset: 0 0 0 0 0 0 0 0

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

### 27.2.34 Timer Mode Register (TMR)

Address(es): SC112.TMR 0008 B331h



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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOMS[1:0]	Timer Operating Mode Select*1	b1 b0 0 0: Timer mode 0 1: Break Field low width determination mode 1 0: Break Field low width output mode 1 1: Setting prohibited	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	TWRC	Counter Write Control	0: Data is written to the reload register and counter 1: Data is written to the reload register only	R/W
b6 to b4	TCSS[2:0]	Timer Count Clock Source Select*1	b6 b4 0 0 0: PCLK 0 0 1: PCLK/2 0 1 0: PCLK/4 0 1 1: PCLK/8 1 0 0: PCLK/16 1 0 1: PCLK/32 1 1 0: PCLK/64 1 1 1: PCLK/128	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Rewrite the TOMS[1:0] and TCSS[2:0] bits only when the timer is stopped (TCST = 0).

#### TWRC Bit (Counter Write Control)

This bit determines whether a value written to TPRES or TCNT is written to the reload register only or is written to both the reload register and the counter.

### 27.2.35 Timer Prescaler Register (TPRE)

Address(es): SCI12.TPRE 0008 B332h



TPRE consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. The counter counts down in synchronization with the counter clock selected by the TMR.TCSS[2:0] bits, and is reloaded with the value in the reload register when it underflows. Underflows of this register provide the clock source to drive counting by the TCNT register. The reload register and read buffer are allocated to the same address, 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.

### 27.2.36 Timer Count Register (TCNT)

Address(es): SCI12.TCNT 0008 B333h



TCNT consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. This down-counter counts underflows of the TPRE register until the TCNT register underflows, and is then reloaded with the value from the reload register. The reload register and read buffer are allocated to the same address, 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.

### 27.3 Operation in Asynchronous Mode

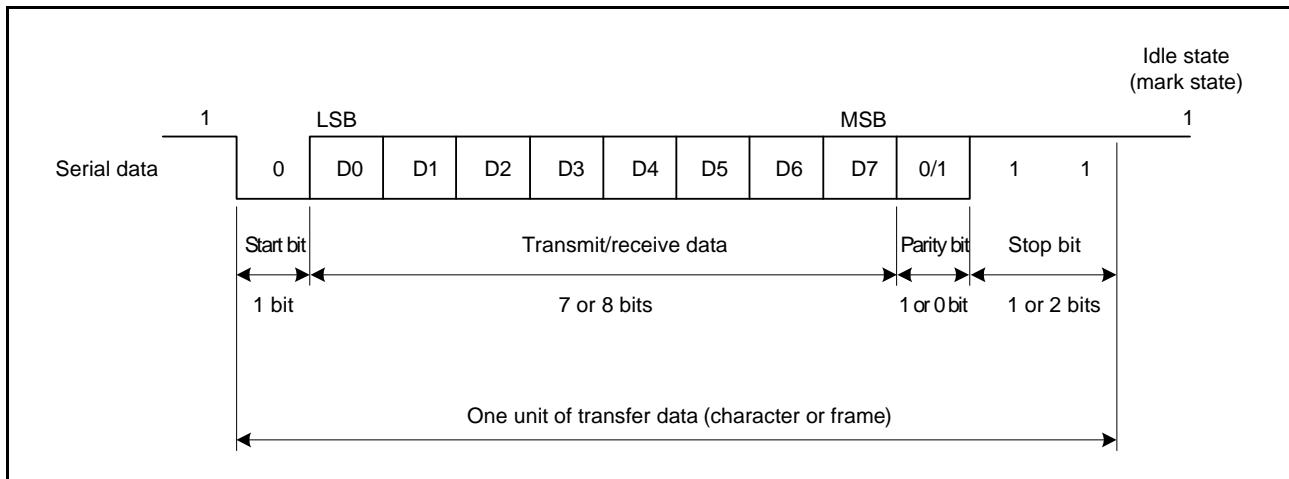
Figure 27.6 shows the general format for asynchronous serial communications.

One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).

In asynchronous serial communications, the communications line is usually held in the mark state (high level).

The SCI monitors the communications line. When the SCI detects a low, it regards that as a start bit and starts serial communication.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.



**Figure 27.6 Data Format in Asynchronous Serial Communications  
(Example with 8-Bit Data, Parity, 2 Stop Bits)**

#### 27.3.1 Serial Data Transfer Format

Table 27.26 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, refer to section 27.4, Multi-Processor Communications Function.

**Table 27.26 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

### 27.3.2 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 27.7. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left( 0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100 [\%] \cdots \text{Formula (1)}$$

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 13)

F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100 (\%) = 46.875\%$$

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.

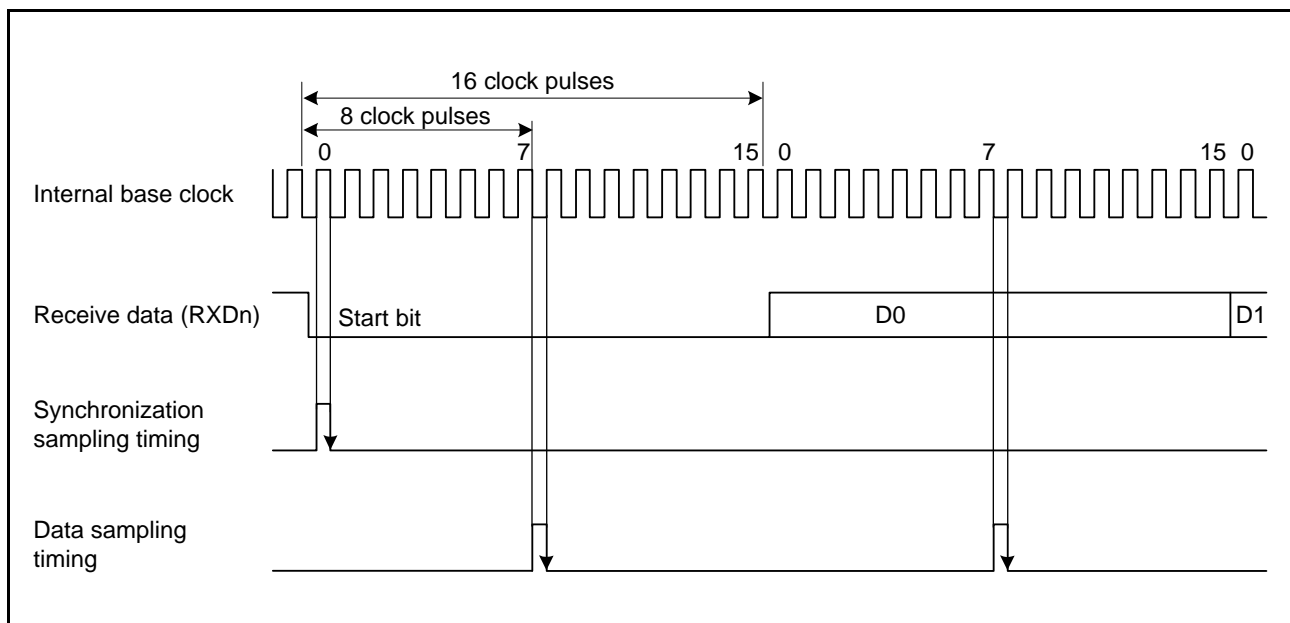


Figure 27.7 Receive Data Sampling Timing in Asynchronous Mode

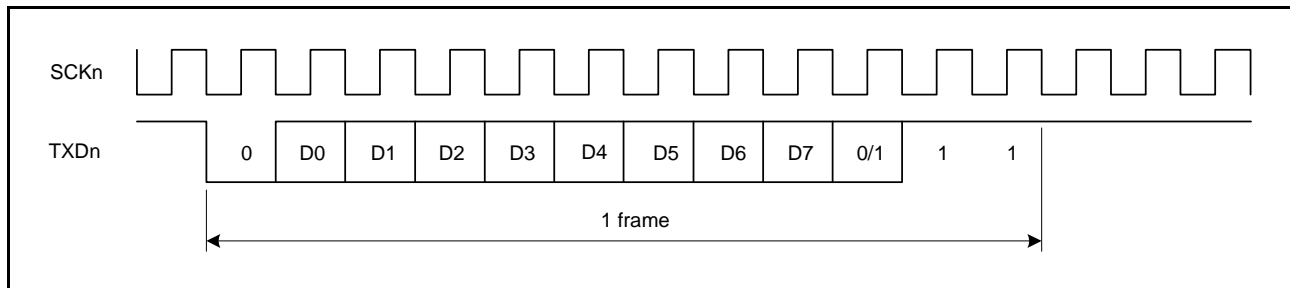


### 27.3.3 Clock

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 MTIOC1A and MTIOC2A can be selected by the SCIn.SEMR.ACS0 bit (n = 1, 5, 12).

When the SCI is operated on an internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 27.8.



**Figure 27.8** Phase Relationship between Output Clock and Transmit Data  
(Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

### 27.3.4 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 high level to the CTS# pin while transmission is in progress does not affect transmission of the current frame, which continues.

In the RTS function, by using the function of output on the RTSn# pin, a low level is output when reception becomes possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]

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]

- The conditions for low-level output have not been satisfied.

### 27.3.5 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 Figure 27.9. 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 setting 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.

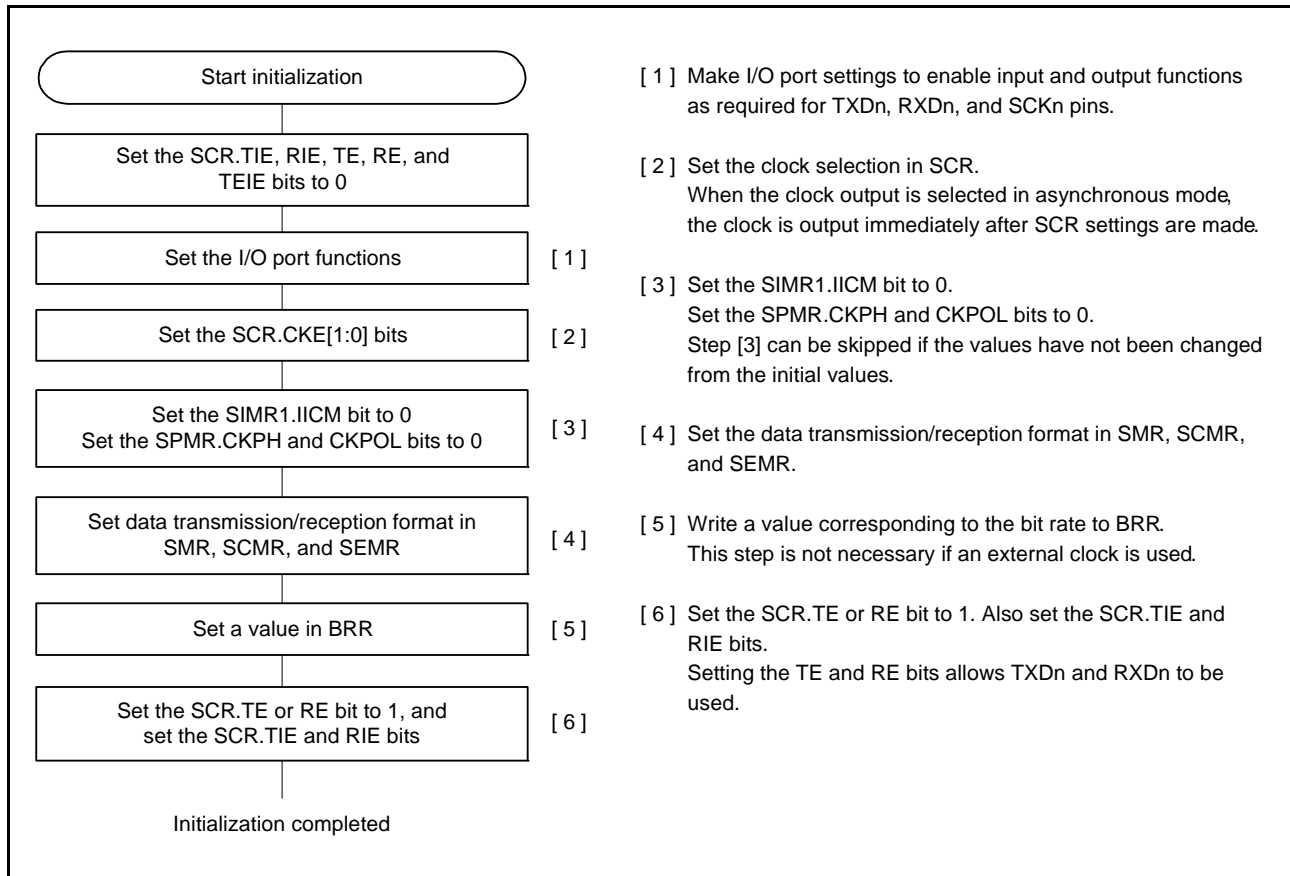


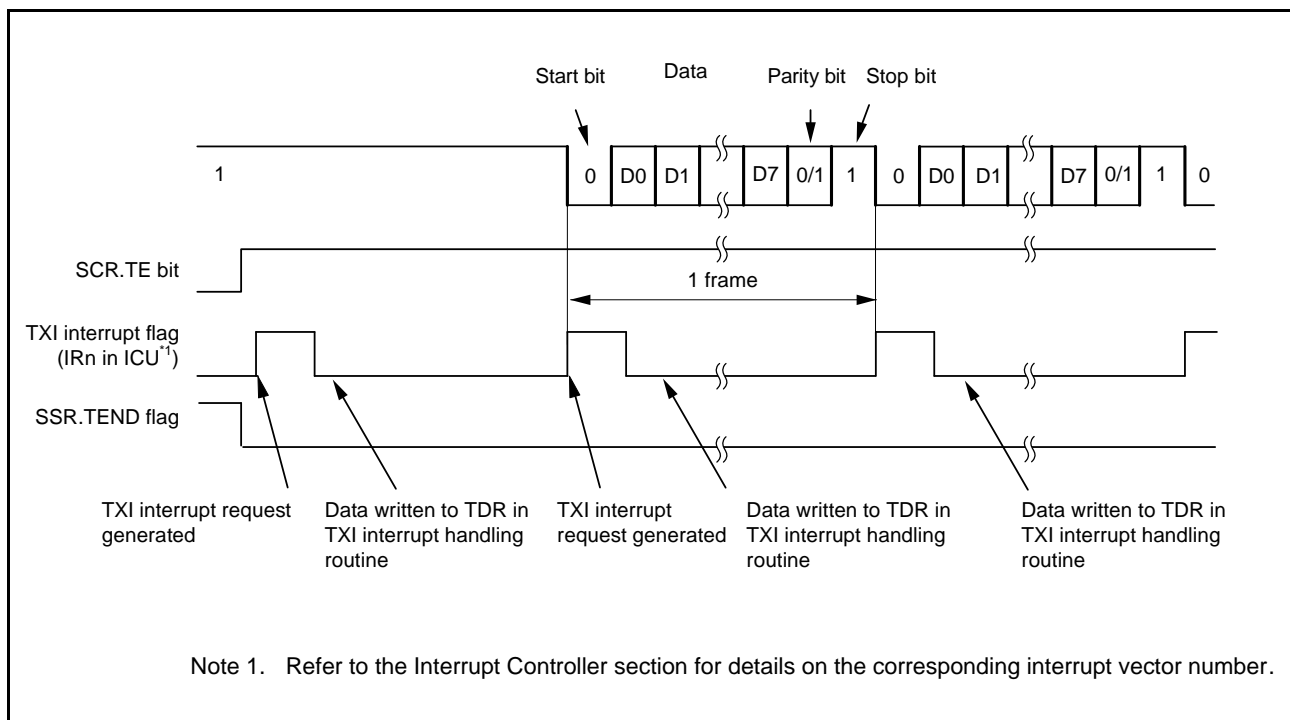
Figure 27.9 Sample SCI Initialization Flowchart (Asynchronous Mode)

### 27.3.6 Serial Data Transmission (Asynchronous Mode)

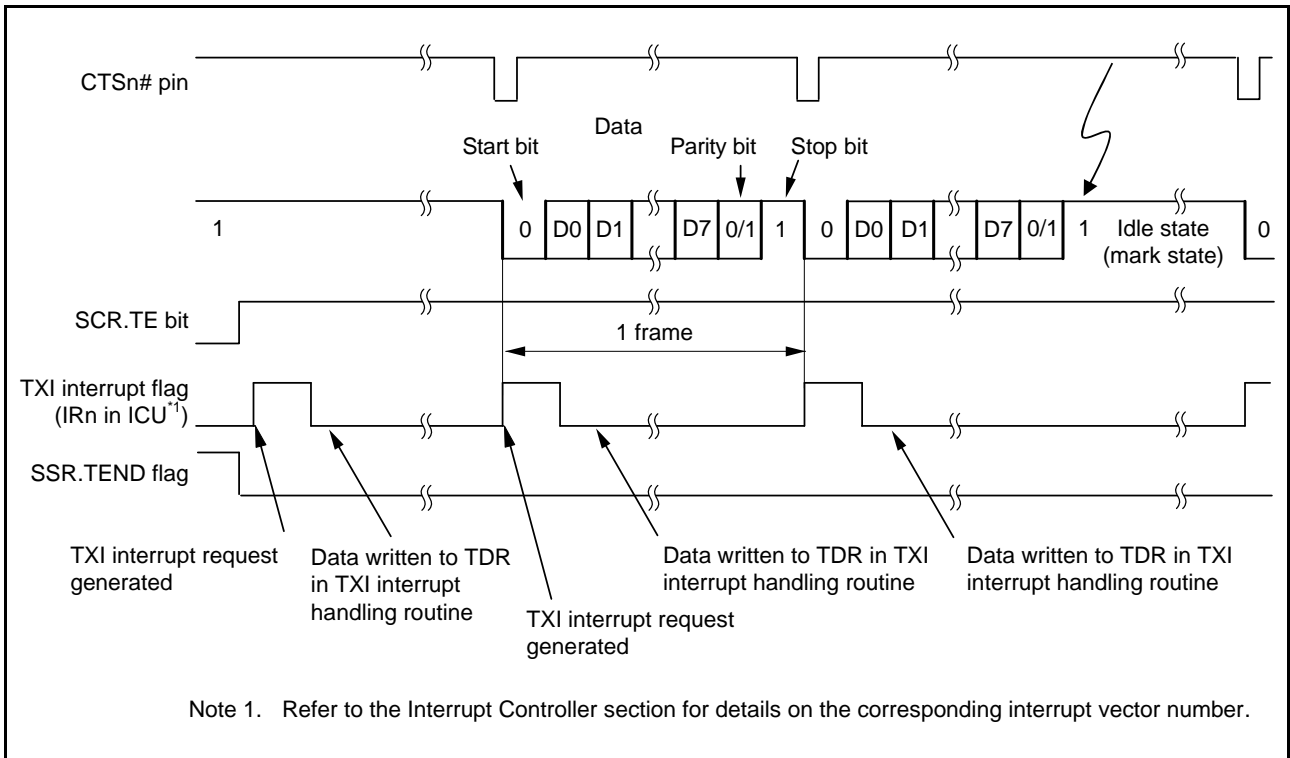
Figure 27.10 to Figure 27.12 show an example of the operation for serial transmission in asynchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt handling routine. 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 2 bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CTSE bit in SPMR is set to 0 (CTS function is disabled) and a low level on the CTSn# 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 transmit data to TDR in the TXI interrupt handling routine before transmission of the current transmit data is completed. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR from the handling routine for TXI requests.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks for updating of (writing to) TDR at the time of stop bit output.
5. When TDR is updated, setting of the CTSE bit in SPMR to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from TDR to TSR and sending of the stop bit, after which serial transmission of the next frame starts.
6. If TDR is not updated, 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 bit in SCR is 1 at this time, the TEND flag in SSR is set to 1 and a TEI interrupt request is generated.

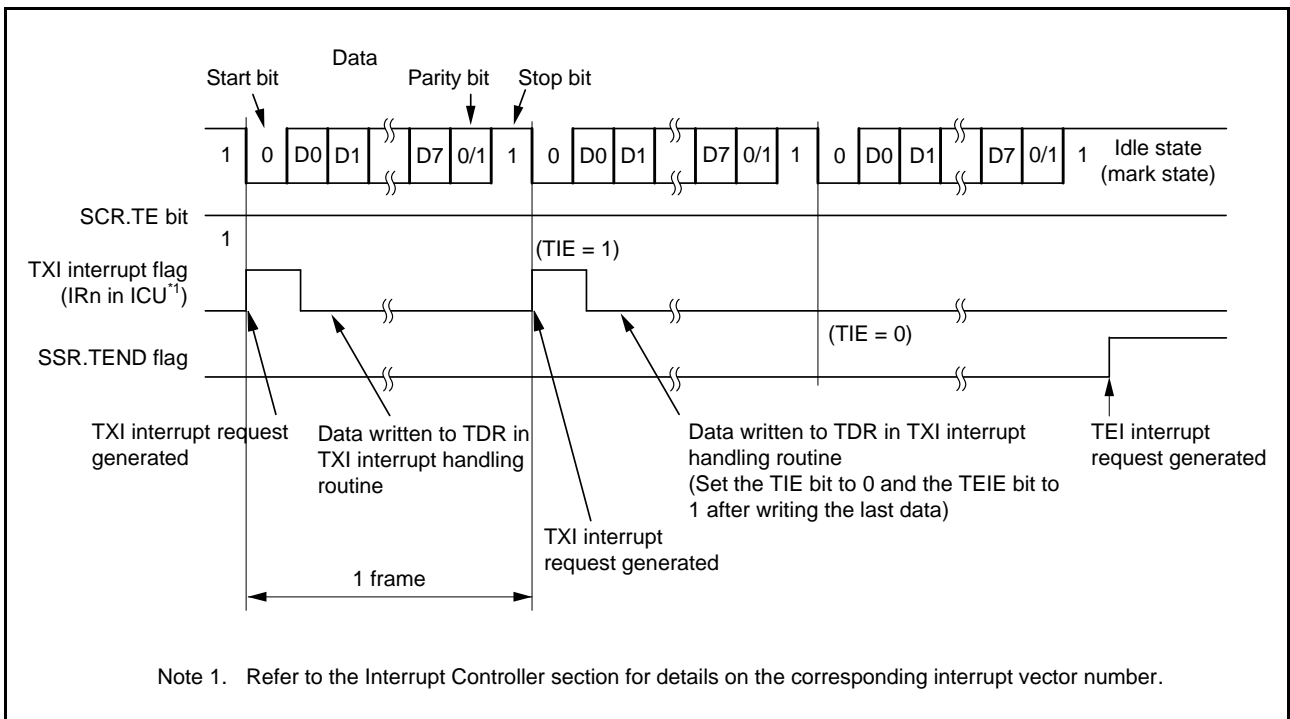
Figure 27.13 shows a sample flowchart for serial transmission in asynchronous mode.



**Figure 27.10 Example of Operation for Serial Transmission in Asynchronous Mode (1)**  
**(with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, at the Beginning of Transmission)**



**Figure 27.11 Example of Operation for Serial Transmission in Asynchronous Mode (2)**  
 (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Used, at the Beginning of Transmission)



**Figure 27.12 Example of Operation for Serial Transmission in Asynchronous Mode (3)**  
 (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)

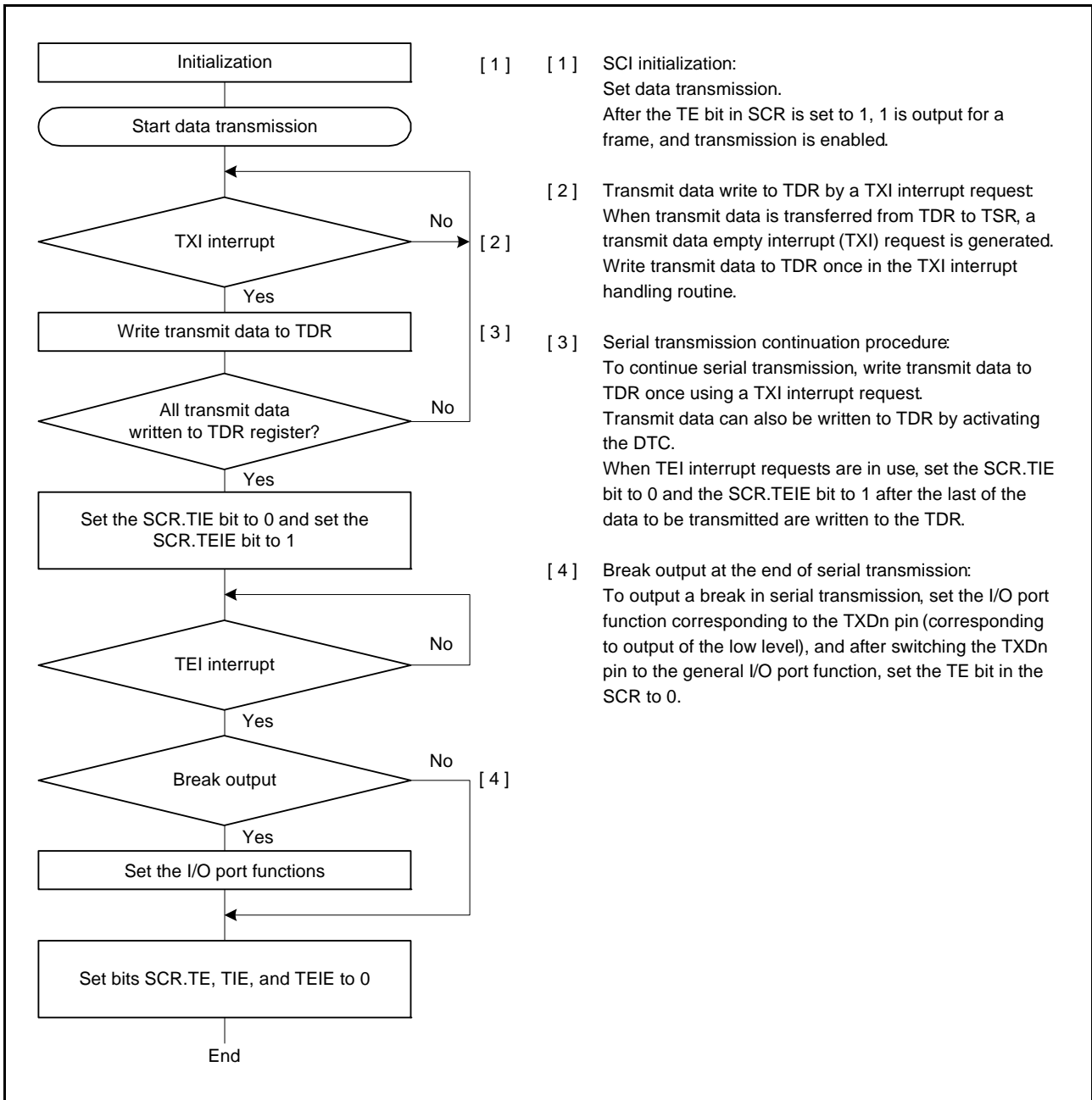
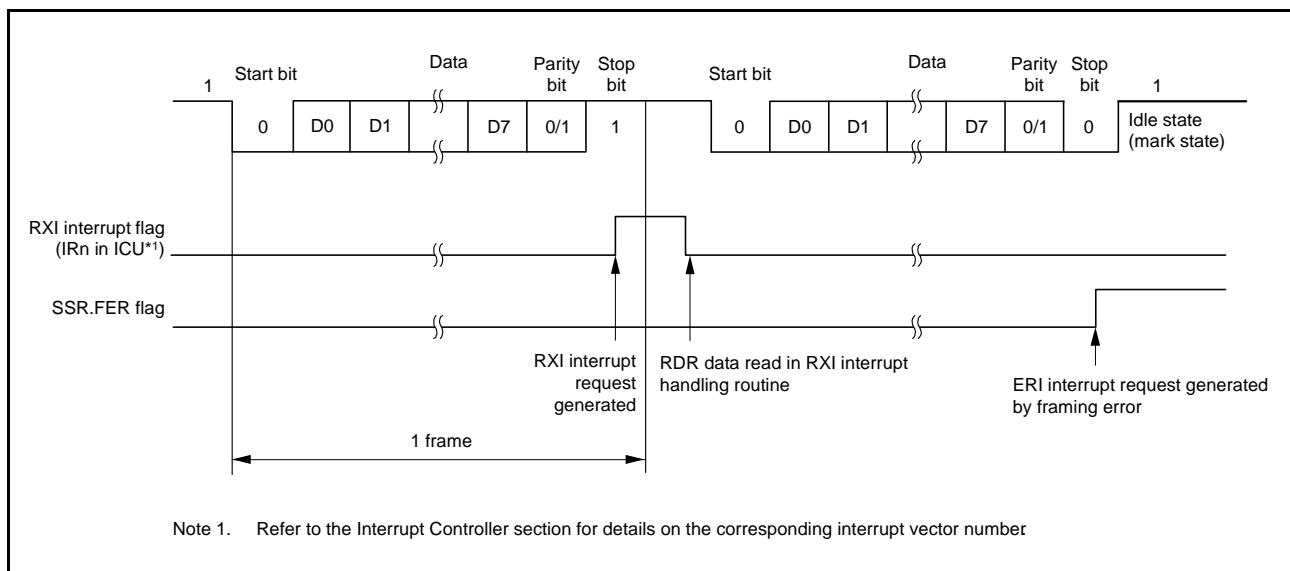


Figure 27.13 Example of Serial Transmission Flowchart in Asynchronous Mode

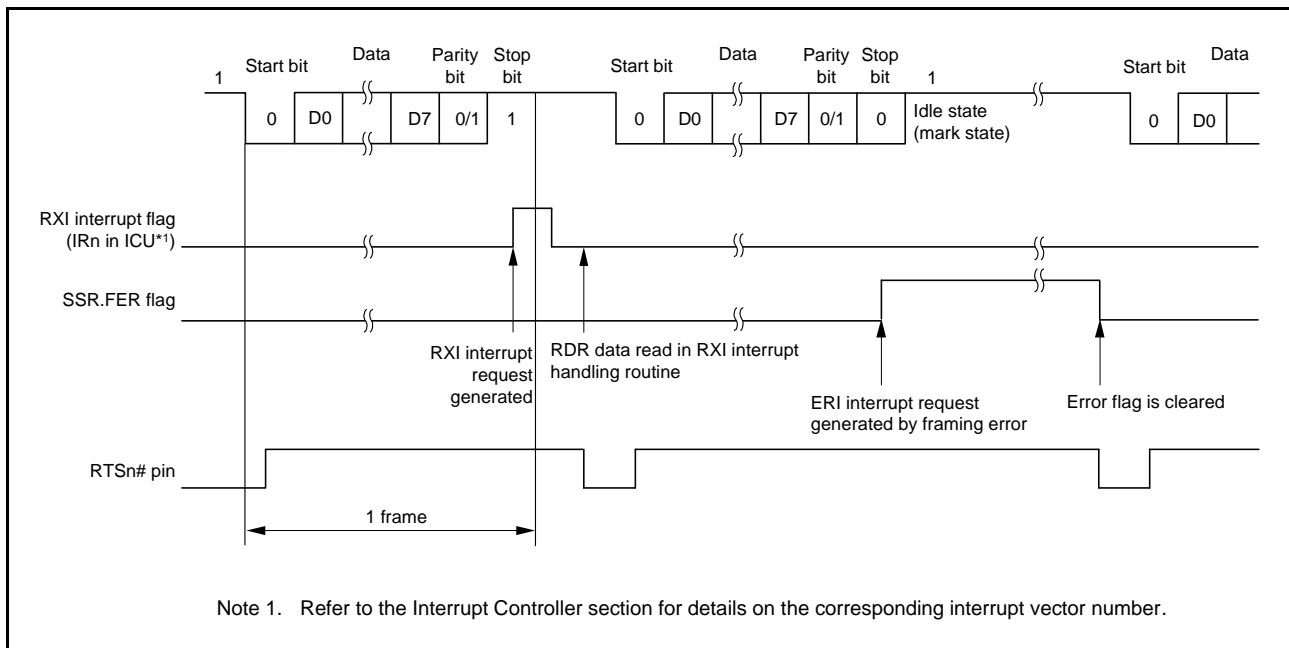
### 27.3.7 Serial Data Reception (Asynchronous Mode)

Figure 27.14 and Figure 27.15 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

1. When the value of the 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 handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to RDR causes the RTSn# pin to output the low level.



**Figure 27.14 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (When RTS Function is Not Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)**



**Figure 27.15 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (When RTS Function is Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)**

Table 27.27 lists the states of the flags in the SSR status register and receive data handling when a receive error is detected.

If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER bits to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in RDR.

Figure 27.16 and Figure 27.17 show samples of flowcharts for serial data reception.

**Table 27.27 Flags in the SSR Status Register and Receive Data Handling**

Flags in the SSR Status Register			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

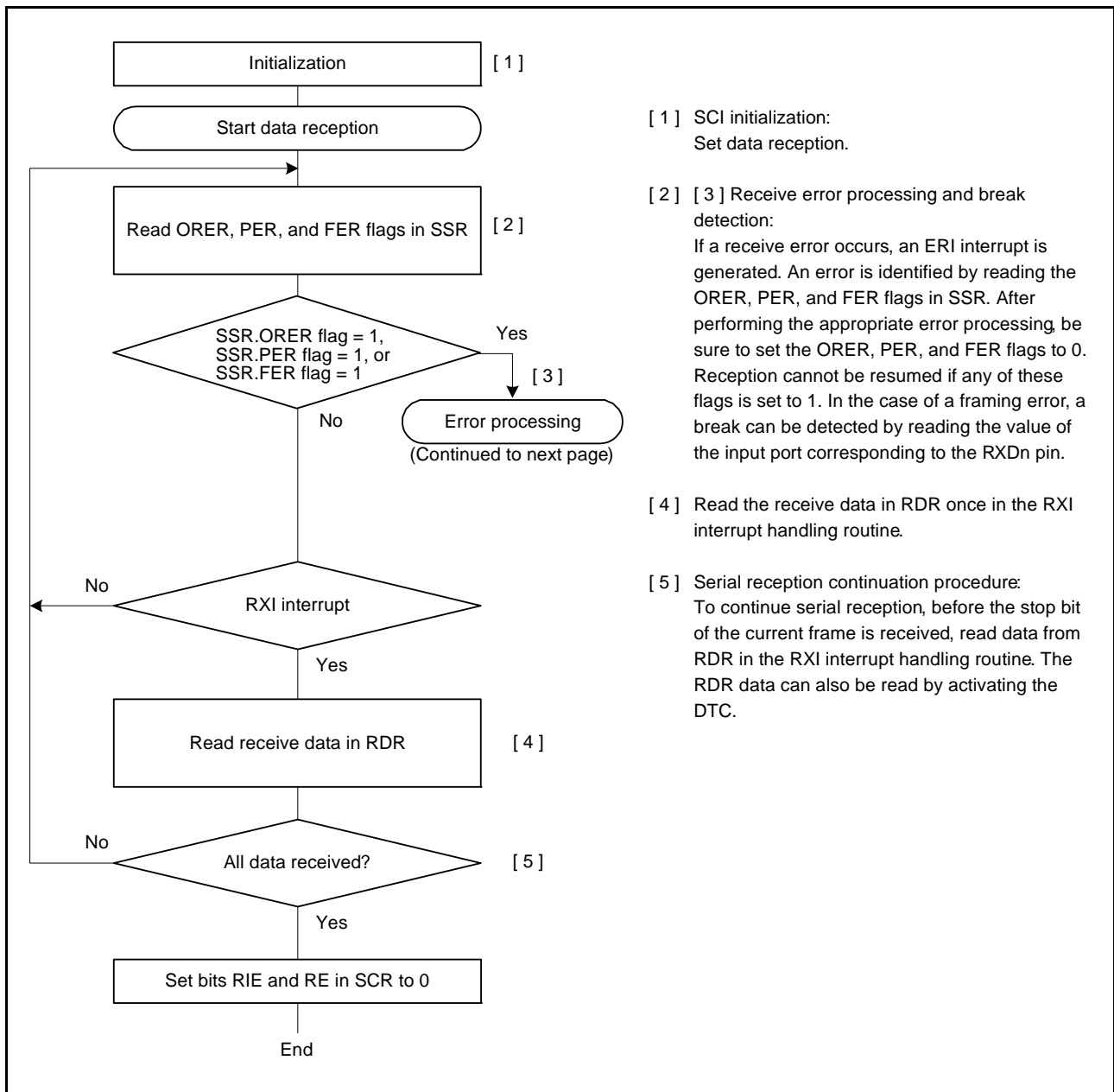


Figure 27.16 Example Flowchart of Serial Reception in Asynchronous Mode (1)



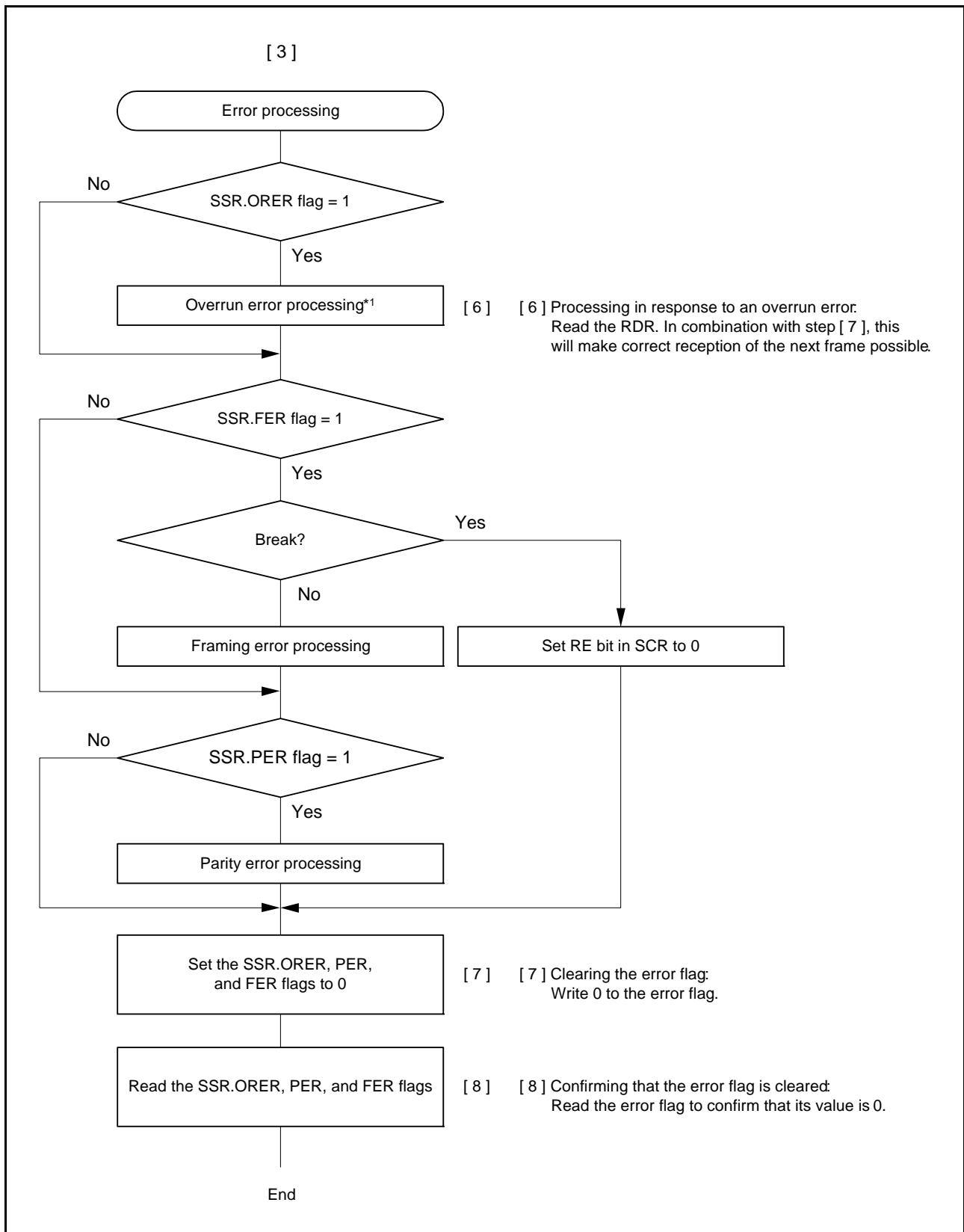
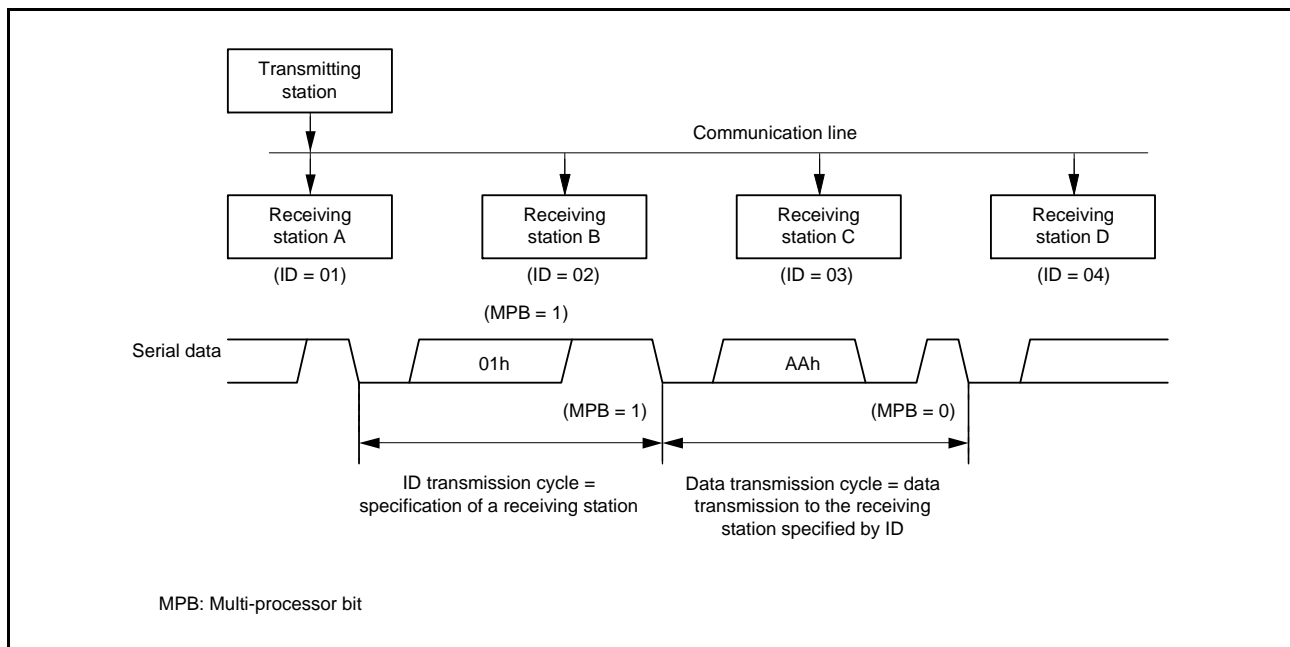


Figure 27.17 Example Flowchart of Serial Reception in Asynchronous Mode (2)

## 27.4 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 27.18 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two match, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1. For supporting this function, the SCI provides the 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 receive 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 27.18 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)**

### 27.4.1 Multi-Processor Serial Data Transmission

Figure 27.19 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.

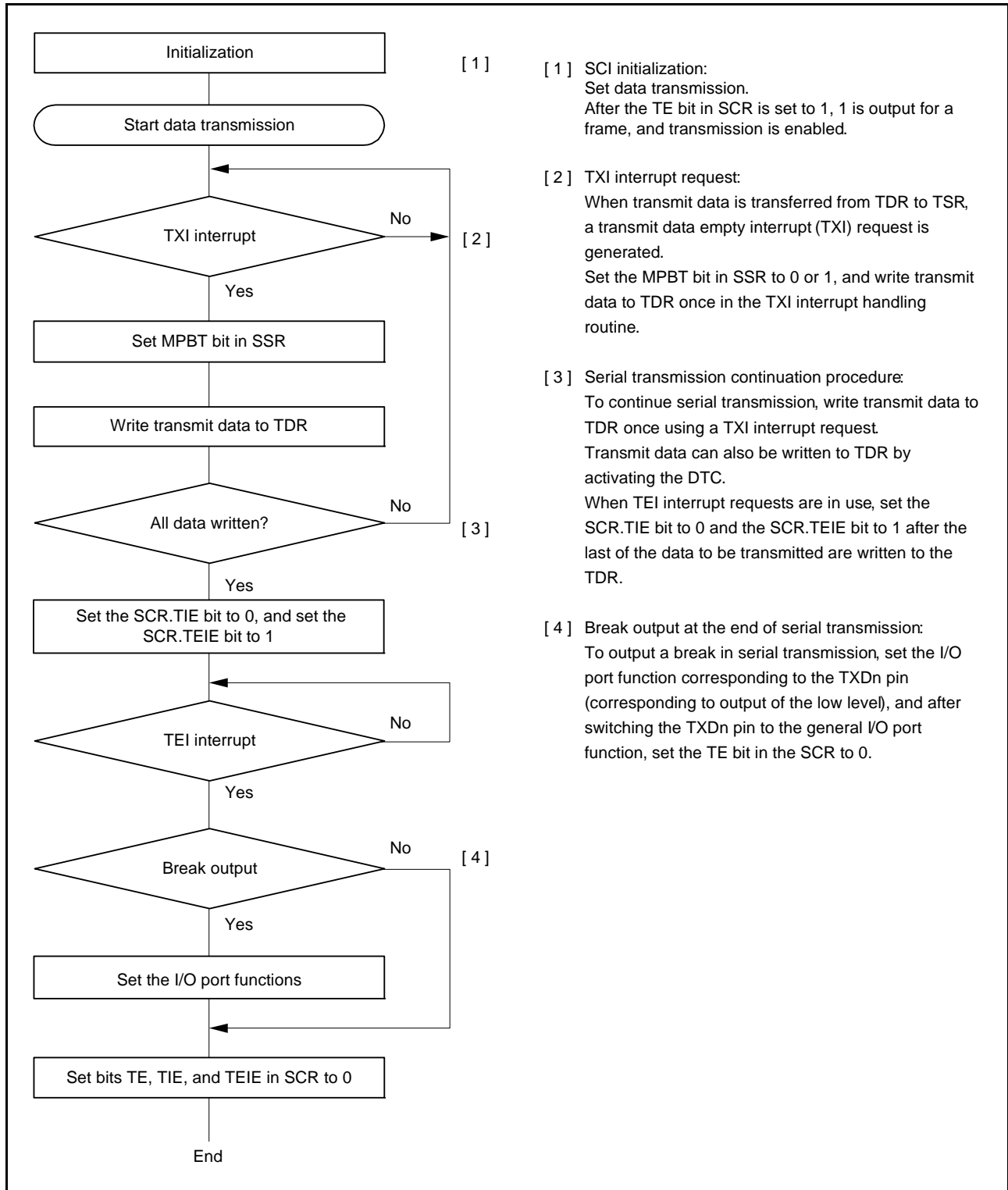


Figure 27.19 Example of Multi-Processor Serial Transmission Flowchart

### 27.4.2 Multi-Processor Serial Data Reception

Figure 27.21 and Figure 27.22 are sample flowcharts of multi-processor data reception. When the MPIE 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 27.20 is the example of operation for reception.

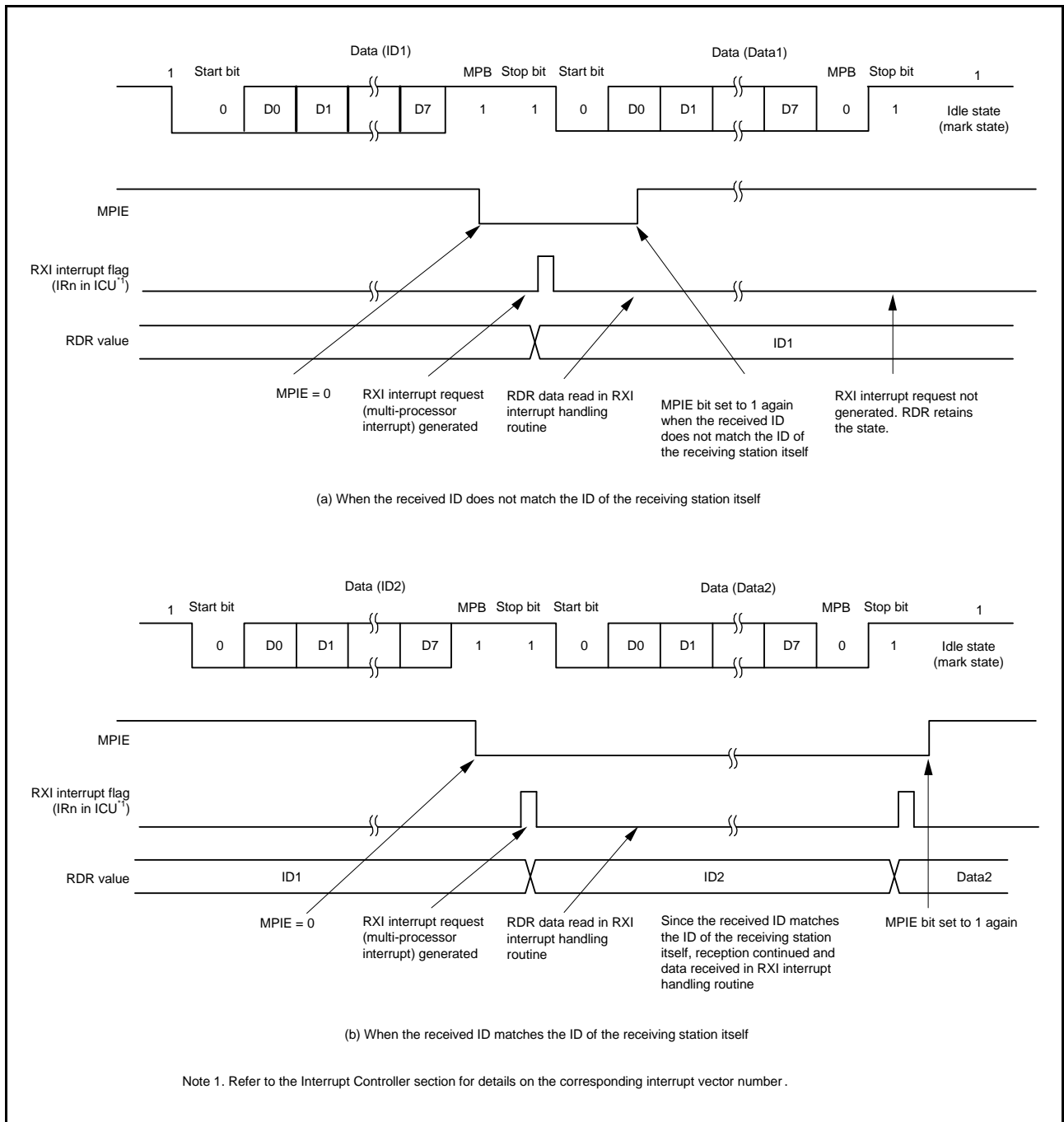


Figure 27.20 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)

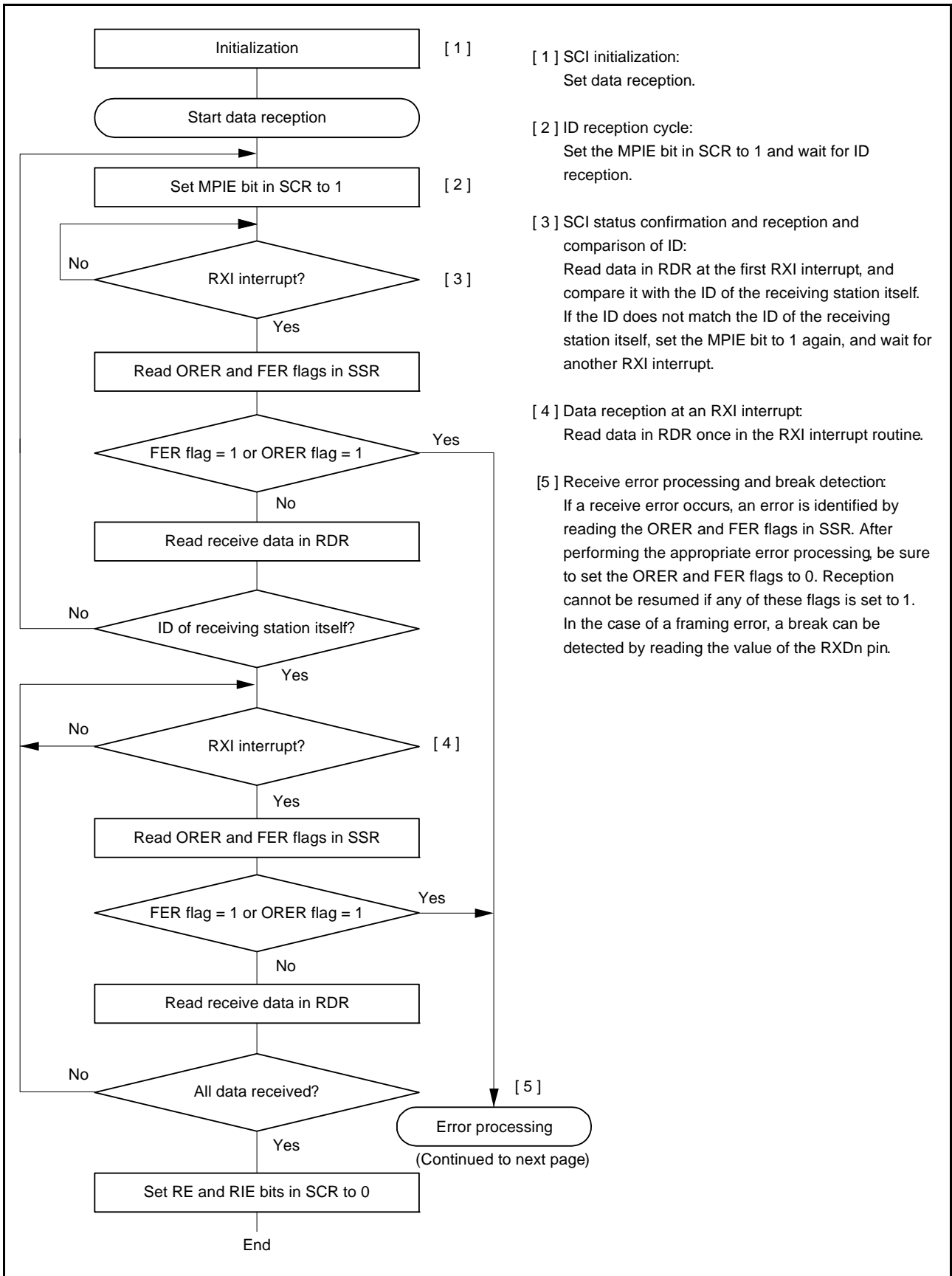


Figure 27.21 Example of Multi-Processor Serial Reception Flowchart (1)

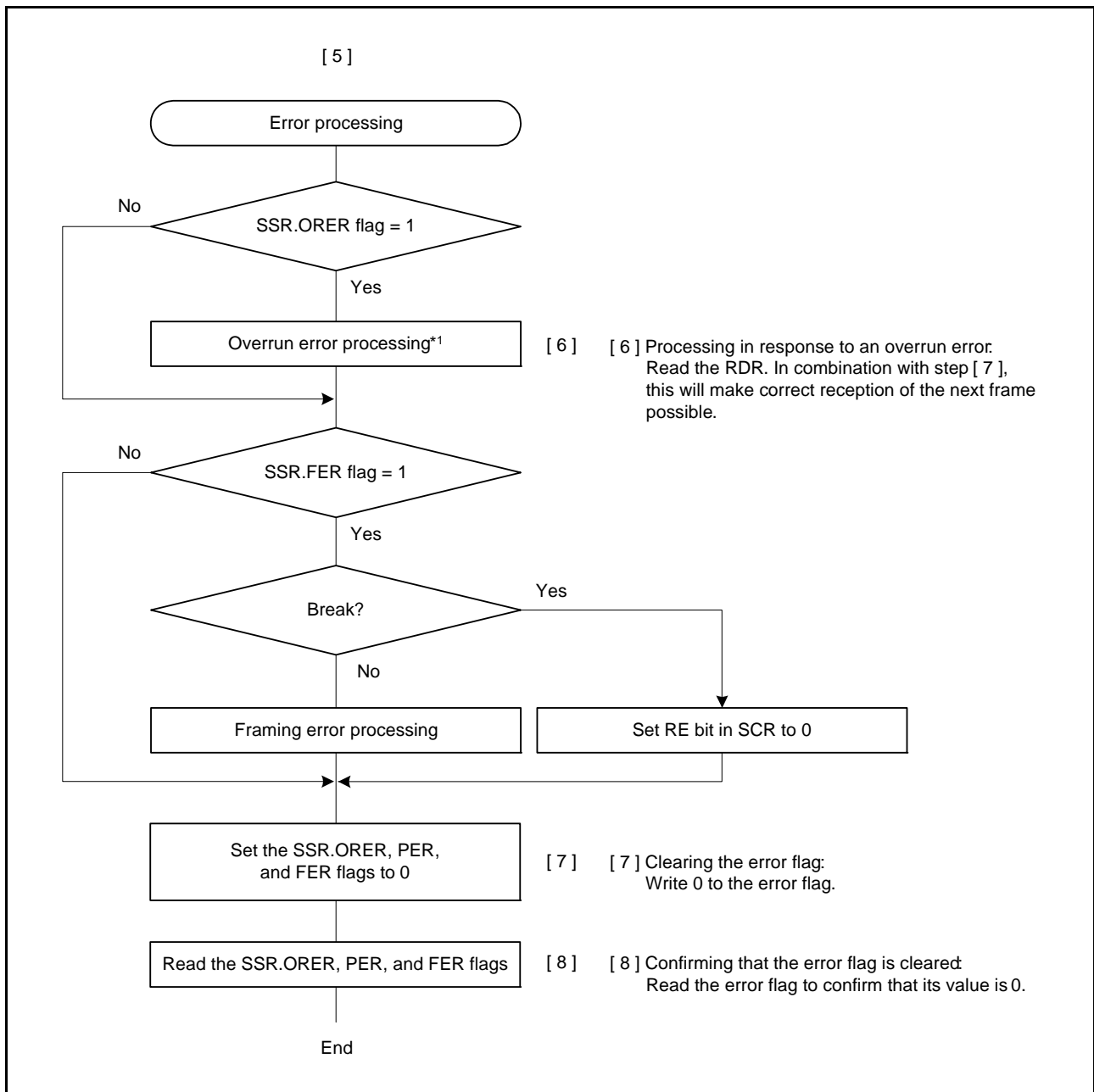


Figure 27.22 Example of Multi-Processor Serial Reception Flowchart (2)

## 27.5 Operation in Clock Synchronous Mode

Figure 27.23 shows the data format for clock synchronous serial data communications.

In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.

In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the communication line holds the last bit output state.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

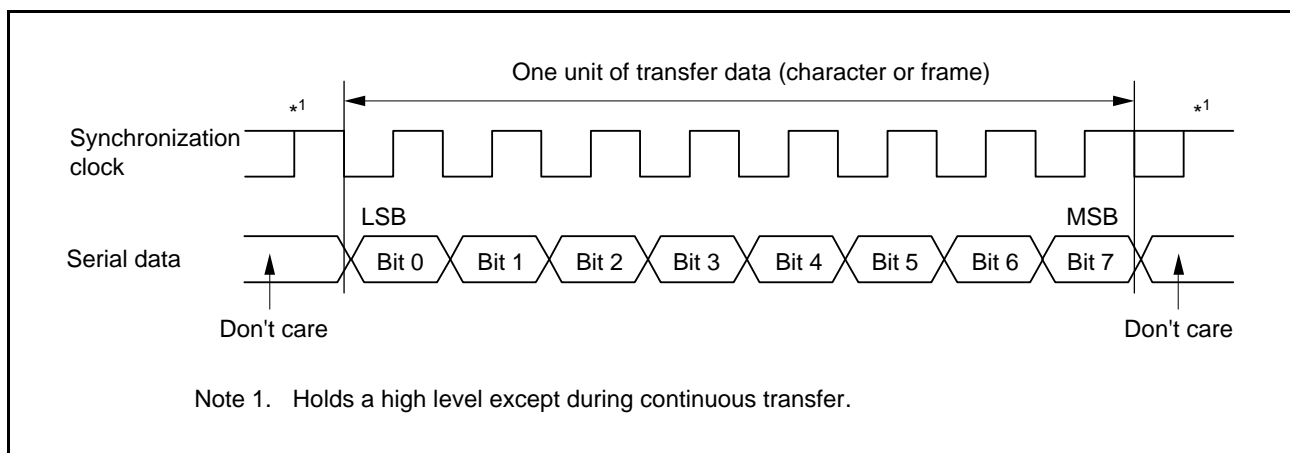


Figure 27.23 Data Format in Clock Synchronous Serial Communications (LSB First)

### 27.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected, according to the setting of the SCR.CKE[1:0] bits.

When the SCI is operated on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output is started at the same time when the SCR.RE bit set to 1. The synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.

When only data reception is performed and the CTS function is enabled, the clock output is not started even when the SCR.RE bit set to 1 if the CTSn# pin input is high when the SCR.RE bit is 0. The synchronization clock output is started when the SCR.RE bit is set to 1 and the CTSn# pin input is low. After that, if the CTSn# pin input is high on completion of the frame reception, the synchronization clock output is stopped at the high level. If the CTSn# pin input continues to be low, the synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.

### 27.5.2 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]

The conditions for low-level output have not been satisfied.



### 27.5.3 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 Figure 27.24. Whenever the operating mode or transfer format is changed, the SCR must be initialized before the change is made.

Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in 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.

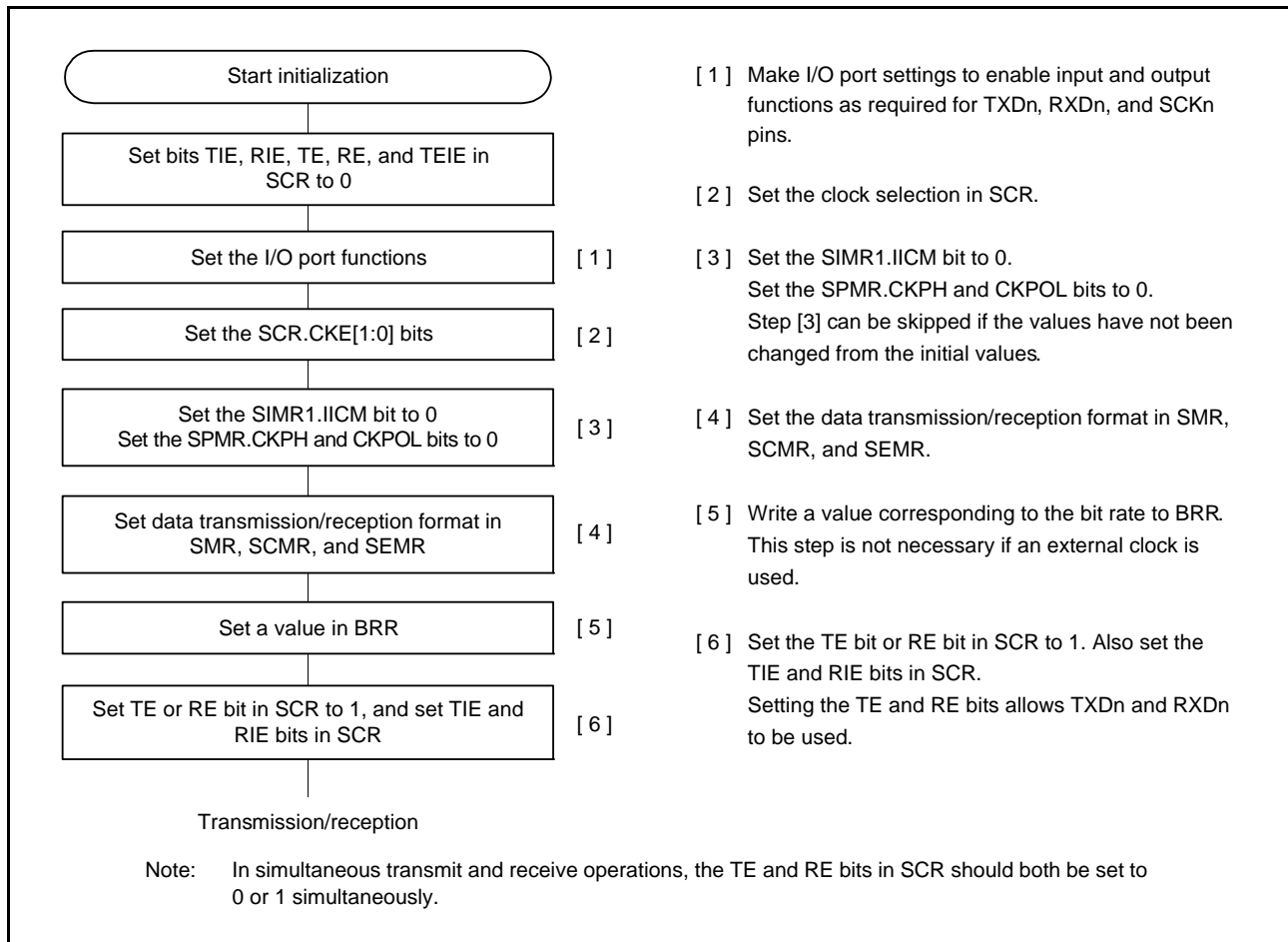


Figure 27.24 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

### 27.5.4 Serial Data Transmission (Clock Synchronous Mode)

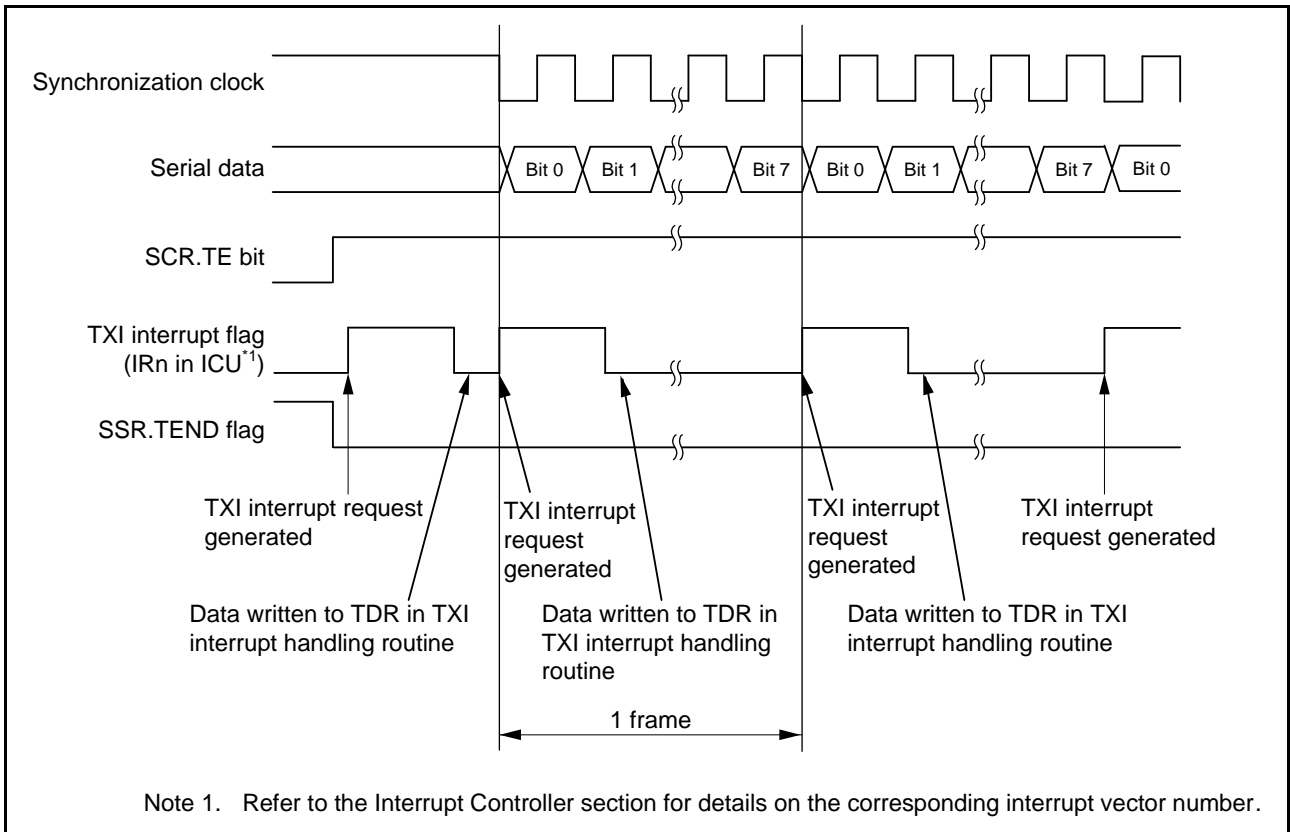
Figure 27.24, Figure 27.25, and Figure 27.26 show an example of the operation for serial transmission in clock synchronous mode.

In serial data transmission, the SCI operates as described below.

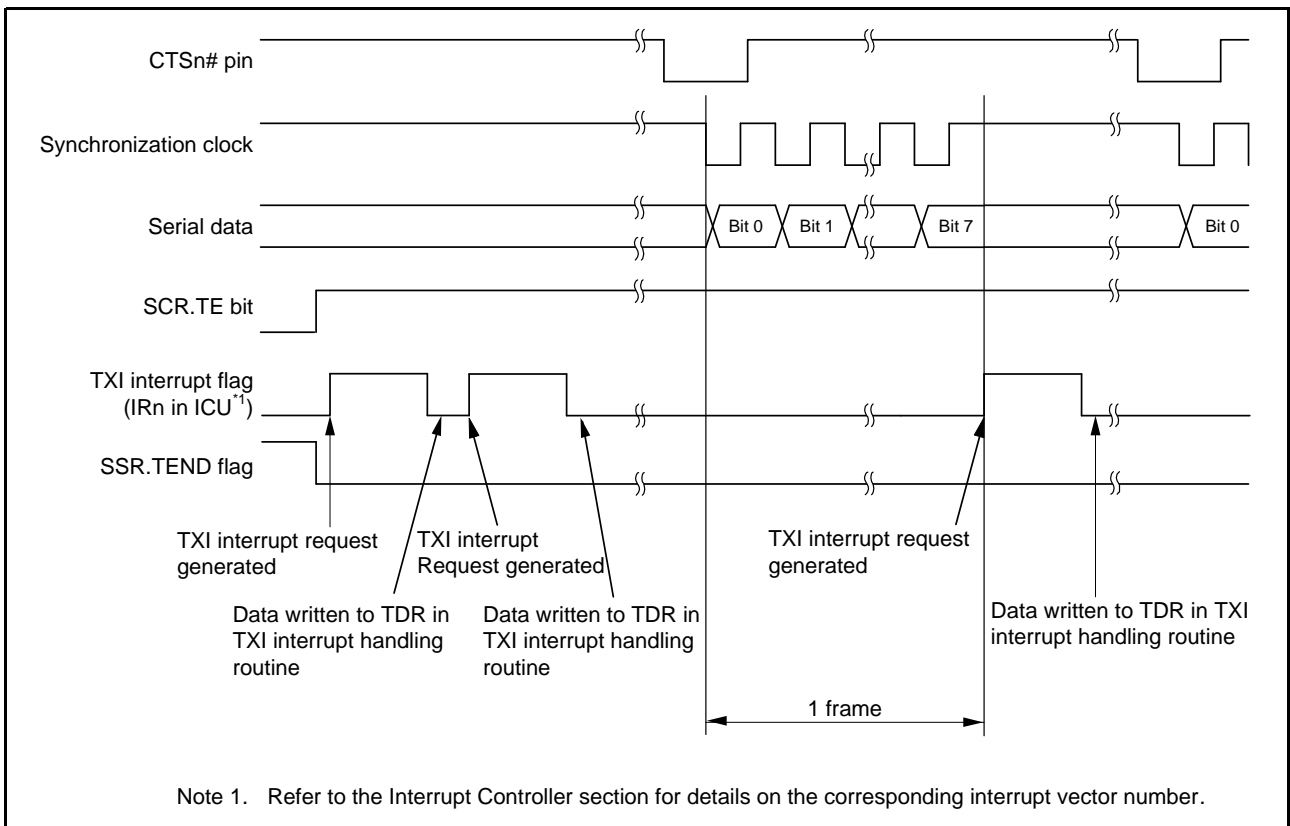
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these 2 bits are set to 1 simultaneously 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 handling routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR from the handling routine for TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when clock output mode has been specified and in synchronization with the input clock when use of an external clock has been specified. Output of the clock signal is suspended until the input CTS signal is at the low level while the CTSE bit in SPMR is 1 (CTS function is enabled).
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.TEND flag 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 27.28 shows a sample flowchart of serial data transmission.

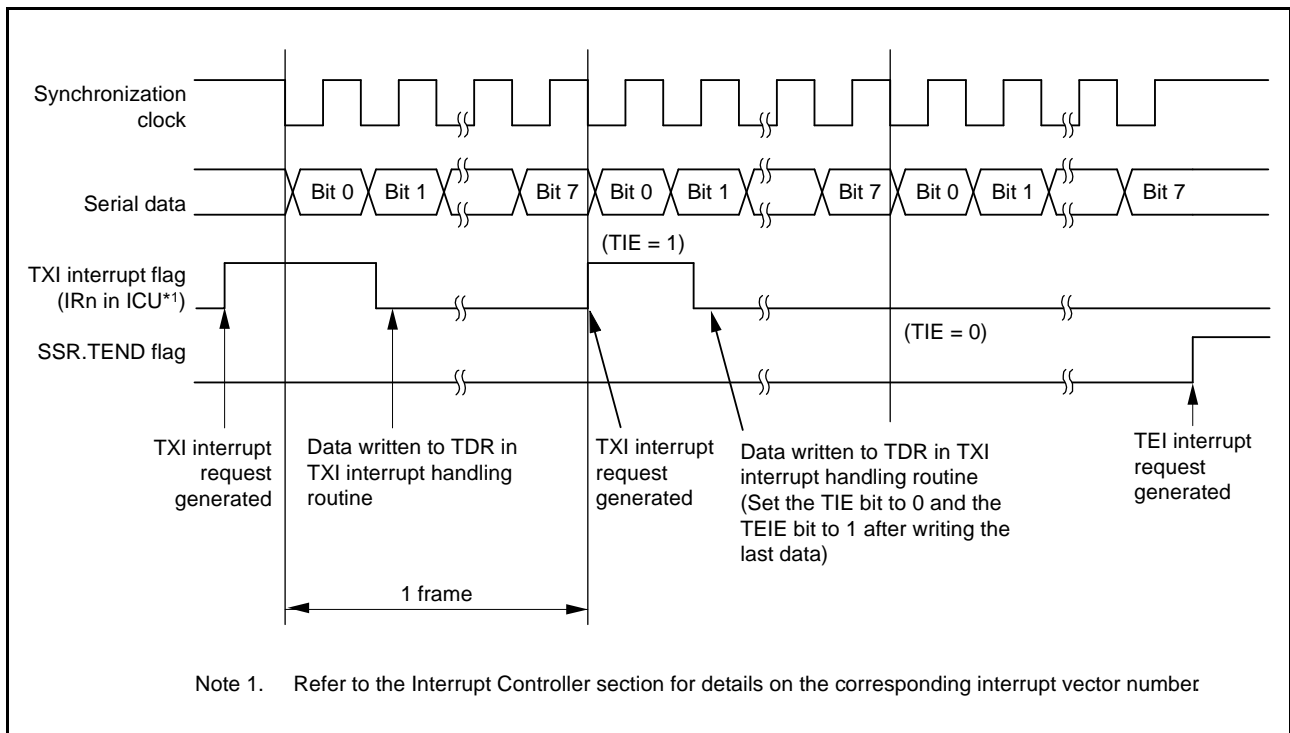
Transmission will not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Be sure to set the receive error flags to 0 before starting transmission. Note that setting the RE bit in SCR to 0 does not clear the receive error flags.



**Figure 27.25 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Not Used at the Beginning of Transmission**



**Figure 27.26 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Used at the Beginning of Transmission**



**Figure 27.27 Example of Serial Data Transmission in Clock Synchronous Mode from the Middle of Transmission until Transmission Completion**

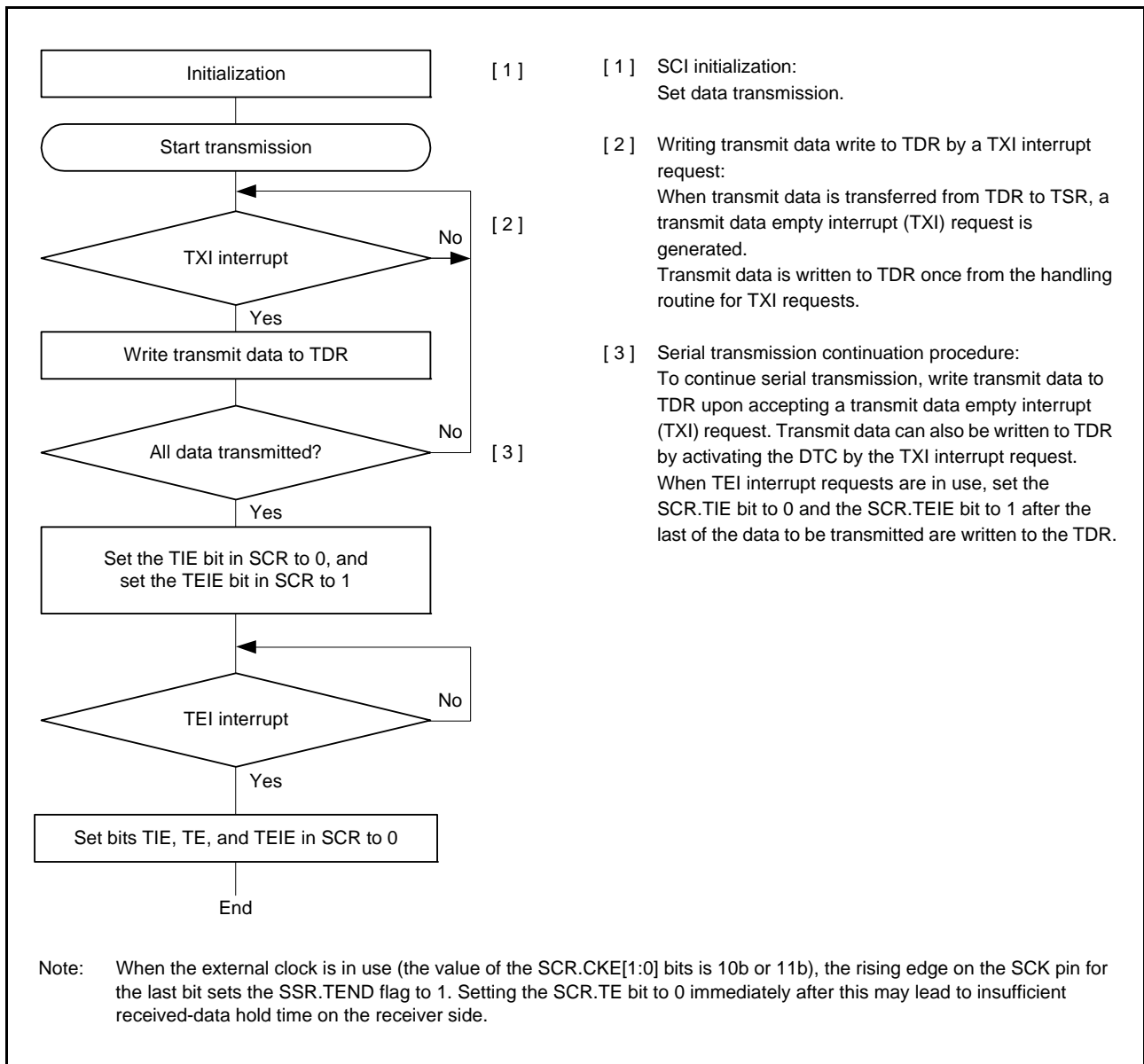
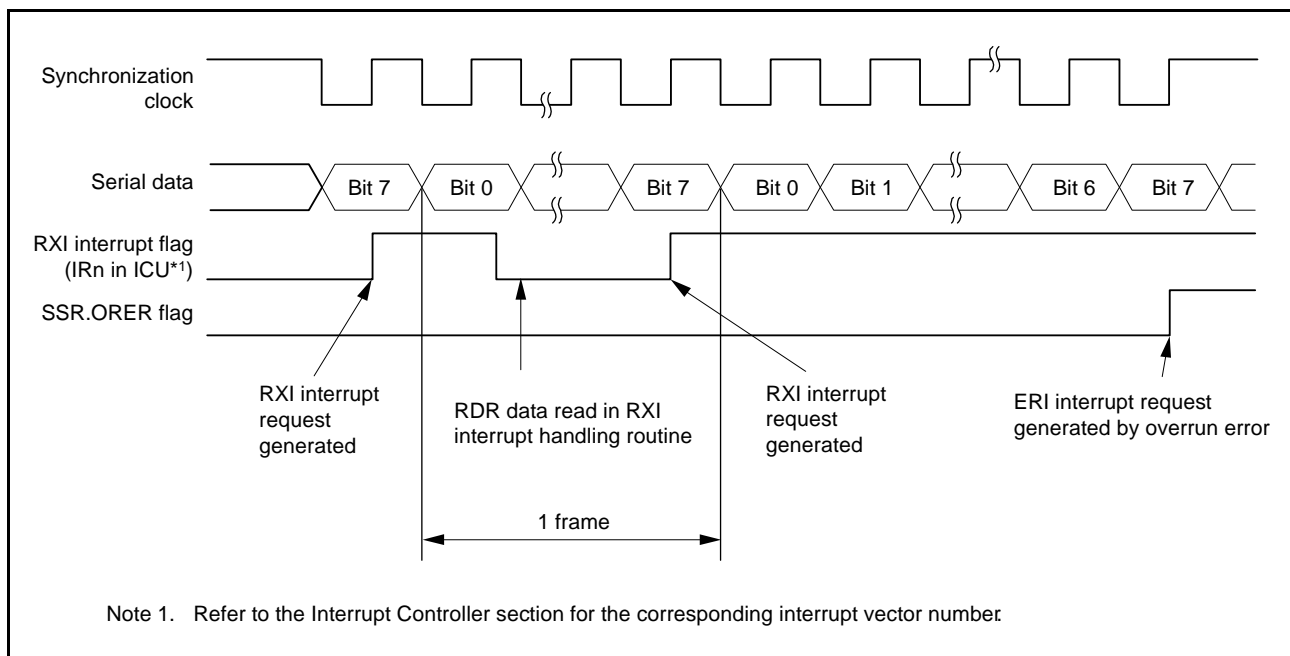


Figure 27.28 Example Flowchart of Serial Transmission in Clock Synchronous Mode

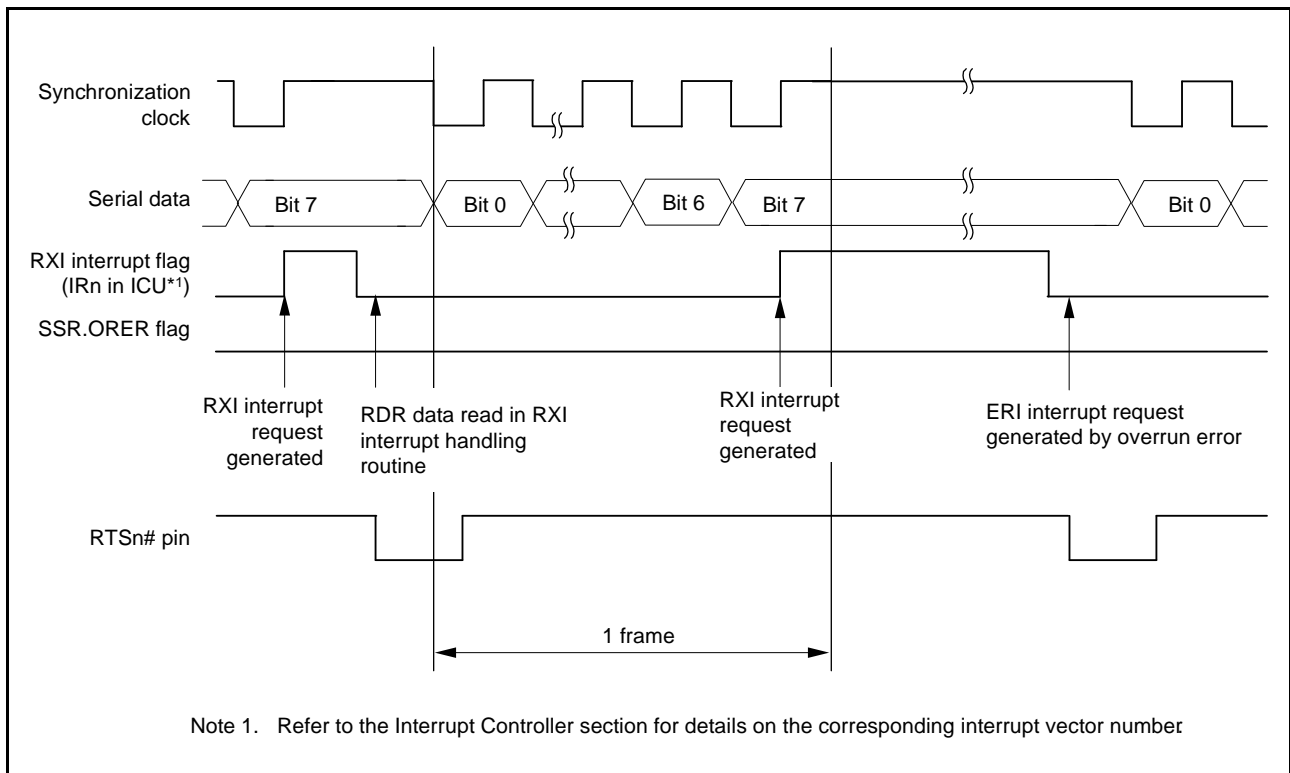
### 27.5.5 Serial Data Reception (Clock Synchronous Mode)

Figure 27.29 and Figure 27.30 show an example of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

1. The value of the RE bit in SCR becoming 1 places the signal output on the RTSn# pin at the low level (when the RTS function is in use).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in RSR.
3. If an overrun error occurs, the ORER 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 handling 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 27.29 Example of Operation for Serial Reception in Clock Synchronous Mode (1)  
(When RTS Function is Not Used)**



**Figure 27.30 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. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read RDR because received data which has not yet been read may be left in RDR.

Figure 27.31 shows a sample flowchart for serial data reception.

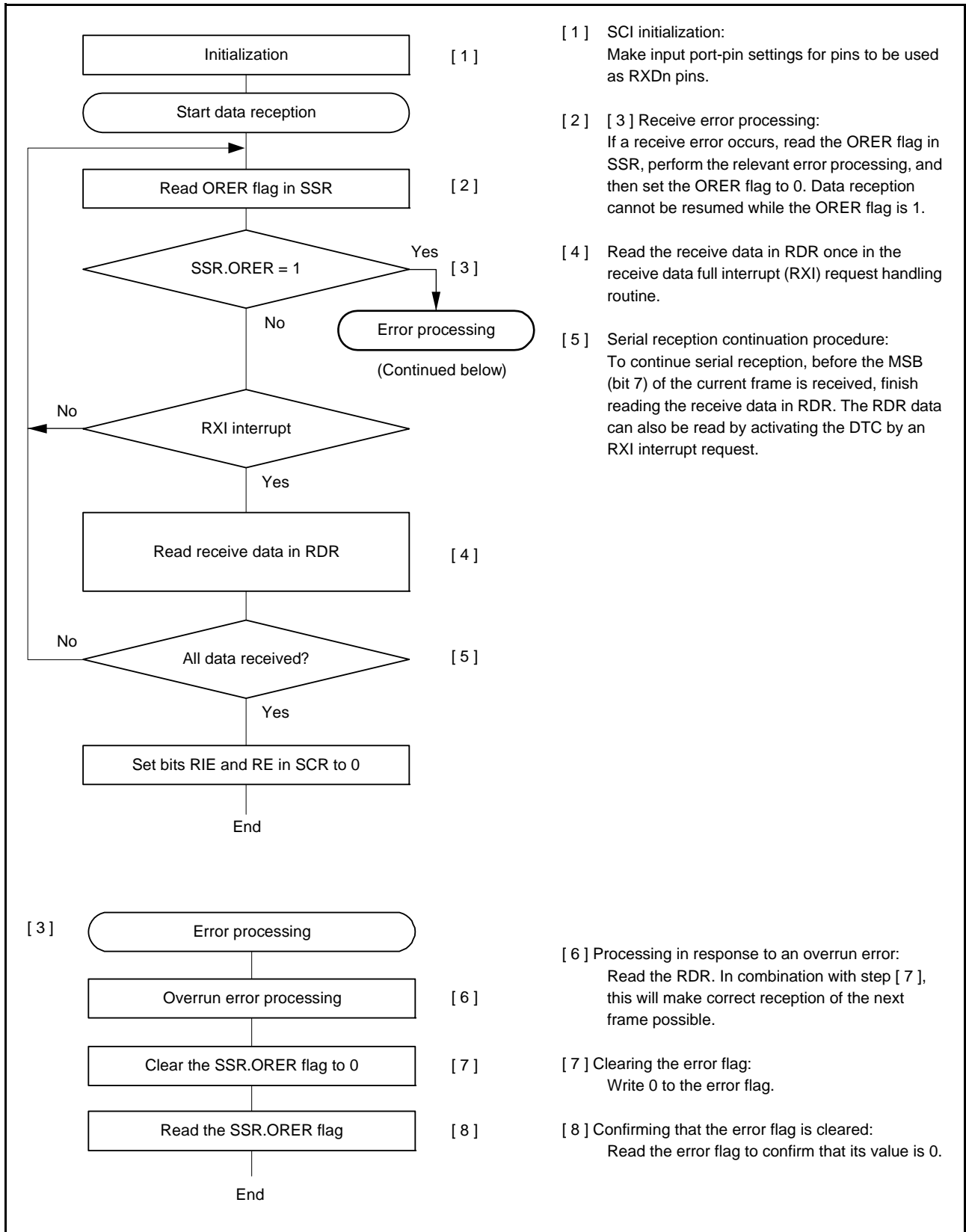


Figure 27.31 Example Flowchart of Serial Reception in Clock Synchronous Mode



### 27.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 27.32 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 set 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.

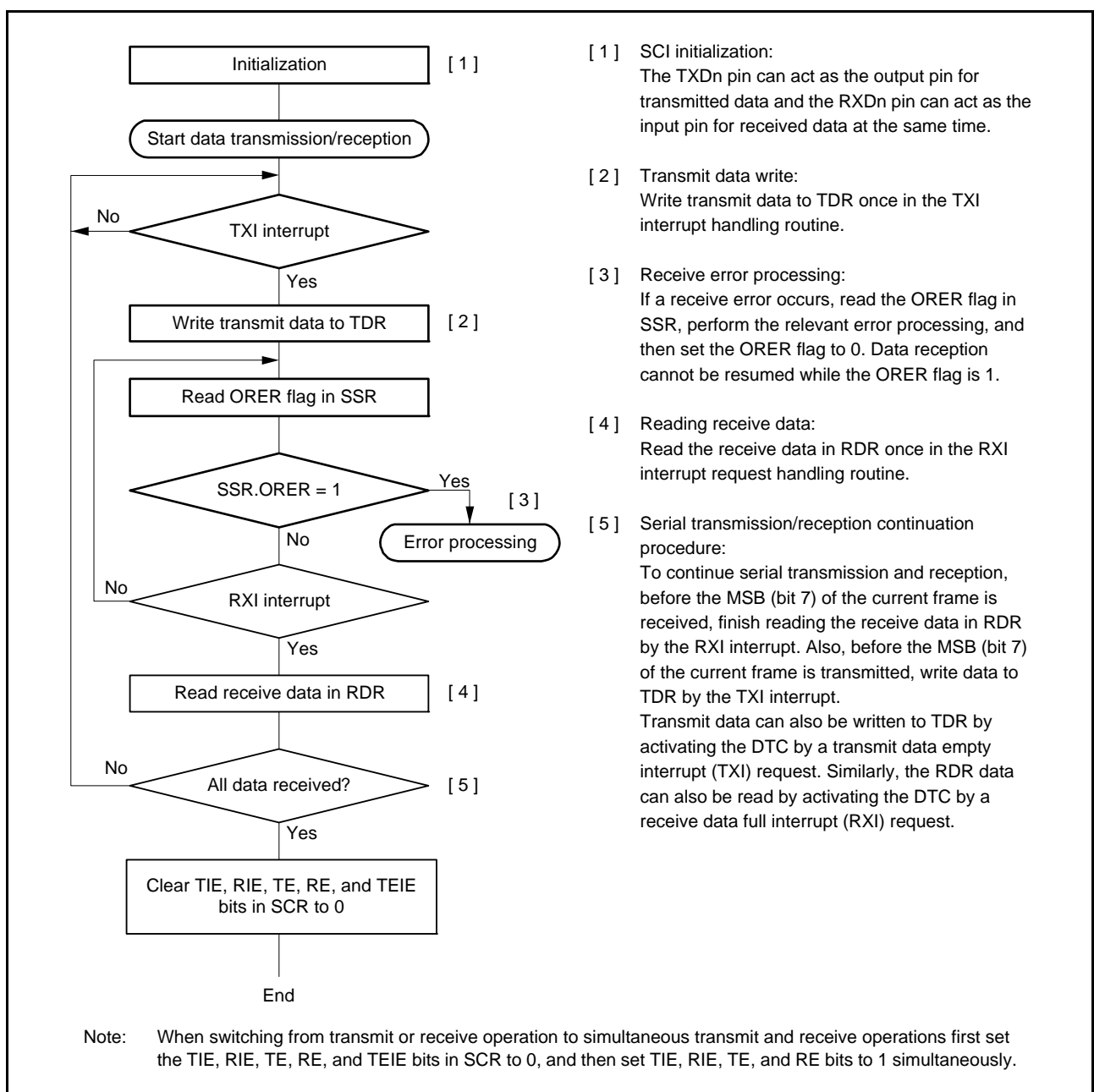


Figure 27.32 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode

## 27.6 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.

Smart card interface mode can be selected using the appropriate register.

### 27.6.1 Sample Connection

Figure 27.33 shows a sample connection between a smart card (IC card) and this MCU.

As in the figure, since this MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.

Setting the TE and RE bits in 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 this MCU can be used to output a reset signal.

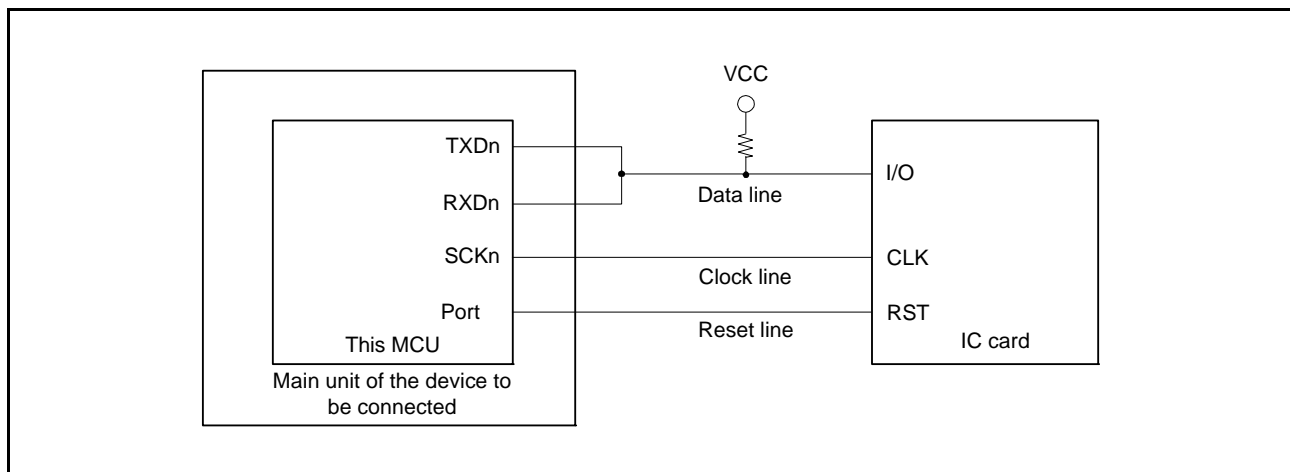


Figure 27.33 Sample Connection with a Smart Card (IC Card)

### 27.6.2 Data Format (Except in Block Transfer Mode)

Figure 27.34 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

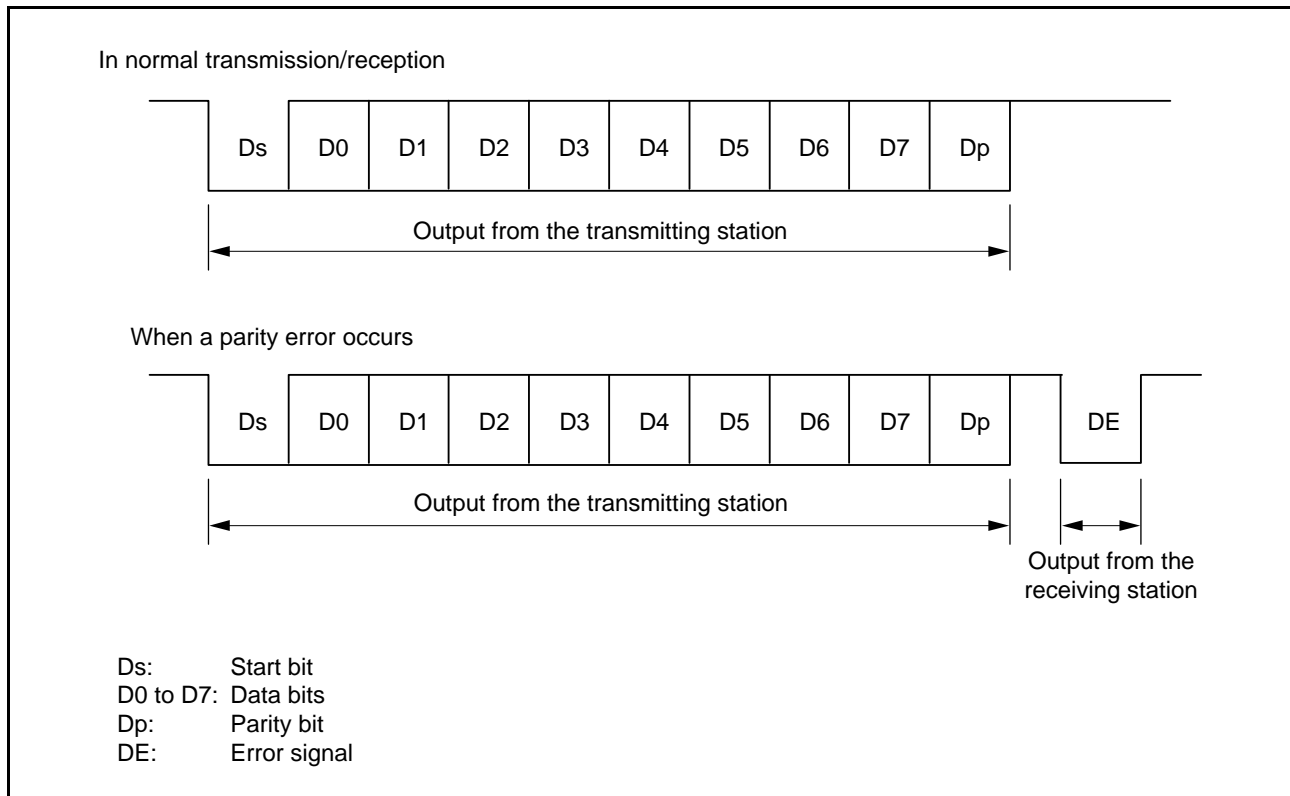


Figure 27.34 Data Formats in Smart Card Interface Mode

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 27.35. 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.

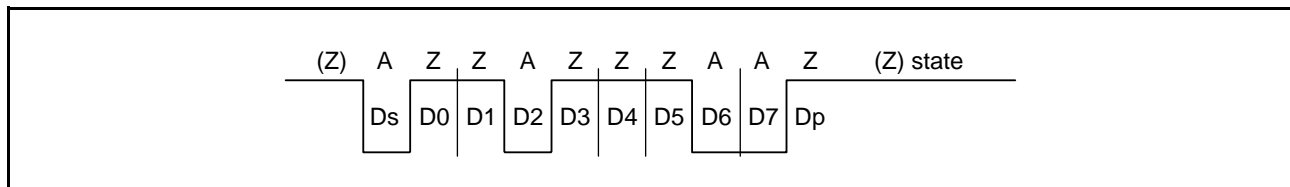


Figure 27.35 Direct Convention (SDIR in SCMR = 0, SINV in SCMR = 0, PM in SMR = 0)

### (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 27.36. 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 this MCU 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.

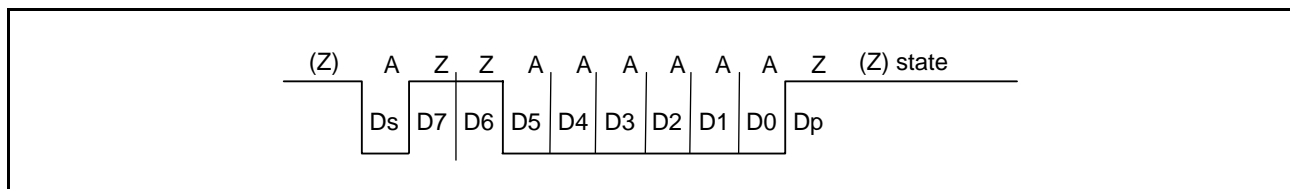


Figure 27.36 Inverse Convention (SDIR in SCMR = 1, SINV in SCMR = 1, PM in SMR = 1)

## 27.6.3 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 retransmitted 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 read as 0 because no error signal is transferred.

### 27.6.4 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).

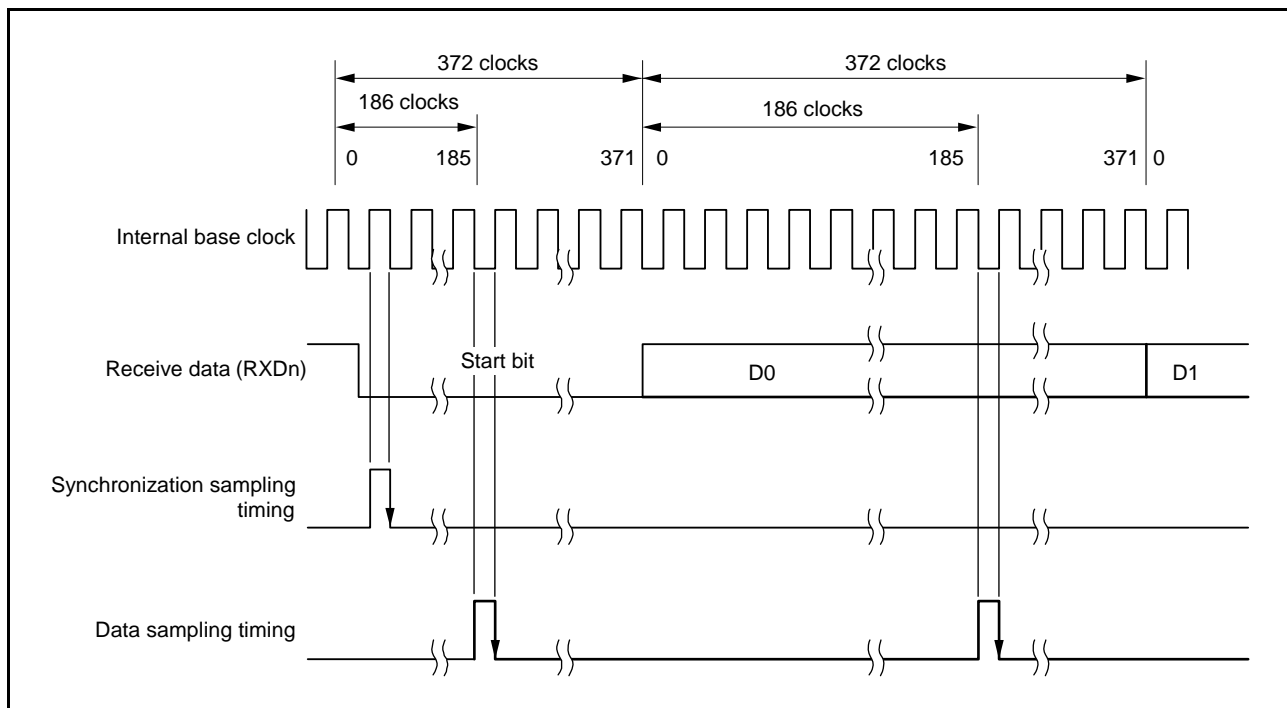
For data reception, the falling edge of the start bit is sampled with the base clock to perform internal synchronization. Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 27.37. 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 27.37 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)**

### 27.6.5 SCI Initialization (Smart Card Interface Mode)

Initialize the SCI following the example of flowchart shown in Figure 27.38.

Be sure to initialize the SCI before switching from transmission mode to reception mode and vice versa. Even if the RE bit is set to 0, the RDR register is not initialized.

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.

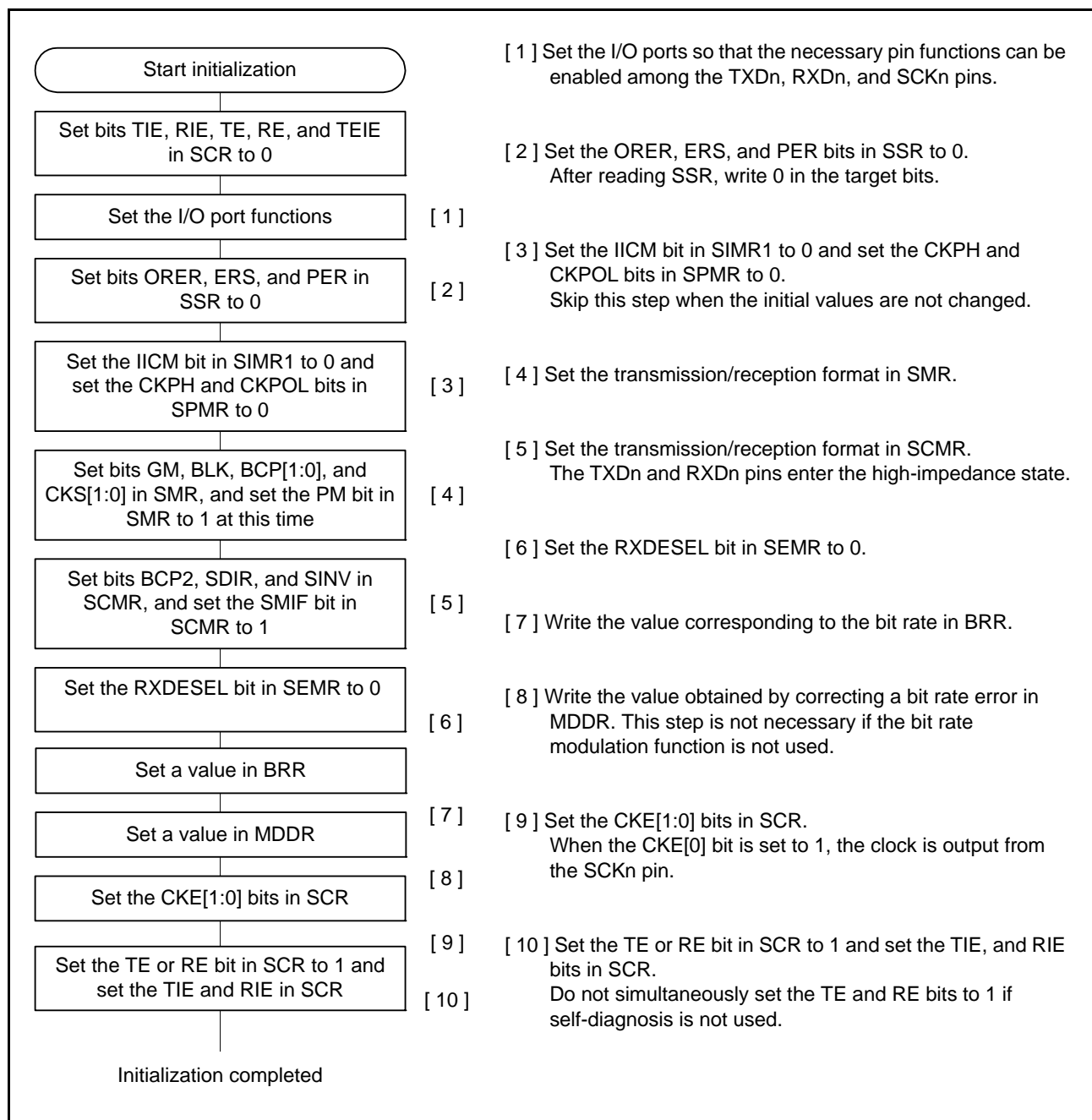


Figure 27.38 Example of SCI Initialization Flowchart (Smart Card Interface Mode)

### 27.6.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. Figure 27.39 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 27.41 shows a sample flowchart of serial transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC.

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 is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DTC transfers the data. If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DTC is not activated. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared, set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag to 0.

When transmitting/receiving data using the DTC, be sure to make settings to enable the DTC before making SCI settings. For DTC settings, refer to section 16, Data Transfer Controller (DTCa).

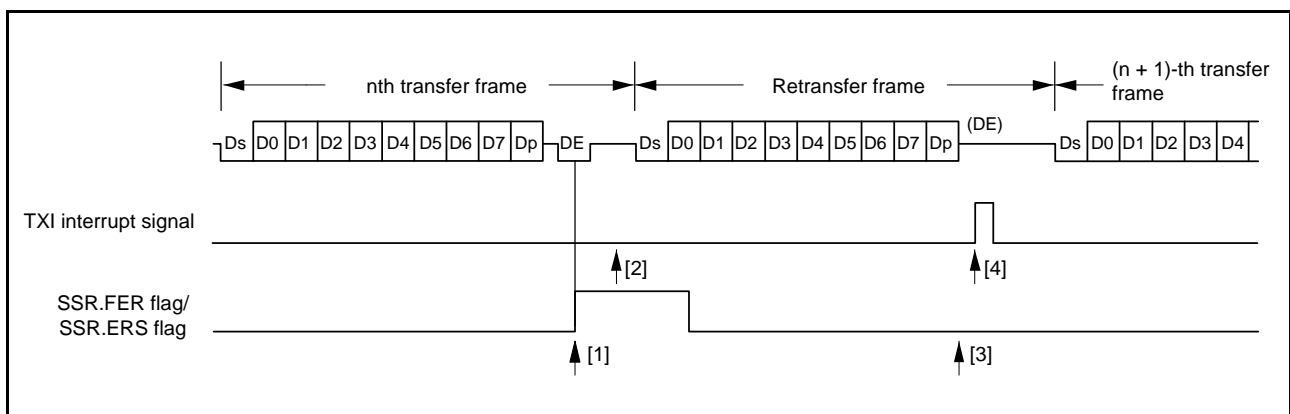


Figure 27.39 Data Retransfer Operation in SCI Transmission Mode

Note that the SSR.TEND flag is set in different timings depending on the GM bit setting in SMR. Figure 27.40 shows the TEND flag generation timing.

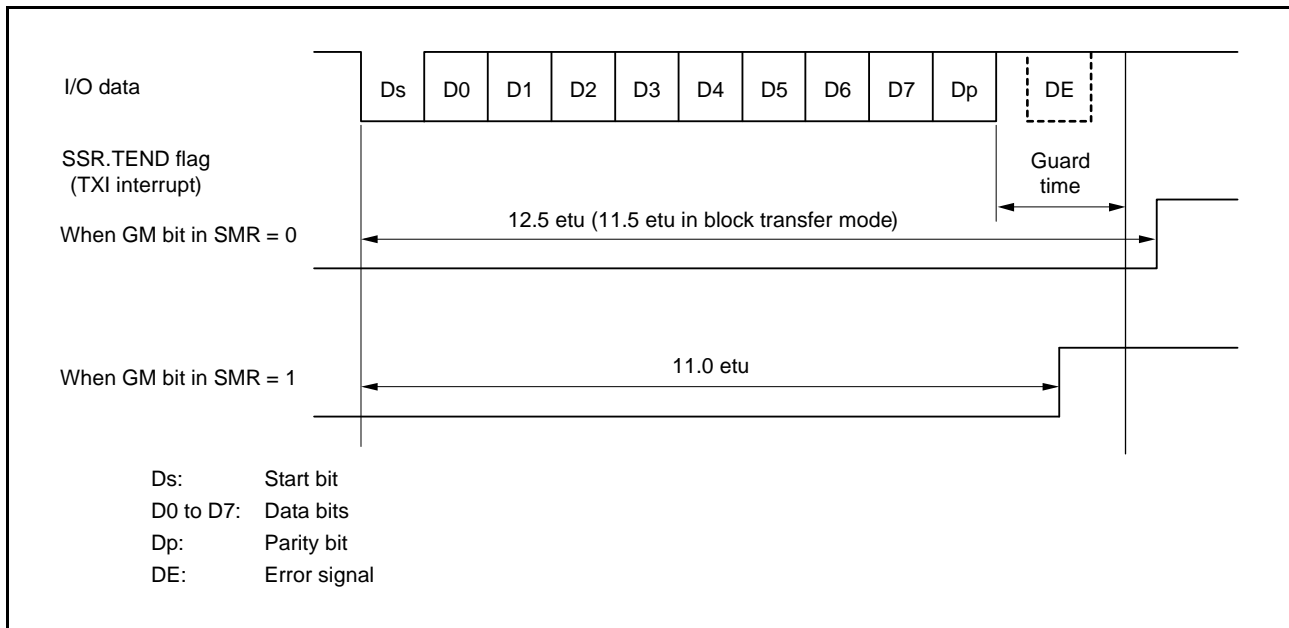


Figure 27.40 SSR.TEND Flag Generation Timing during Transmission



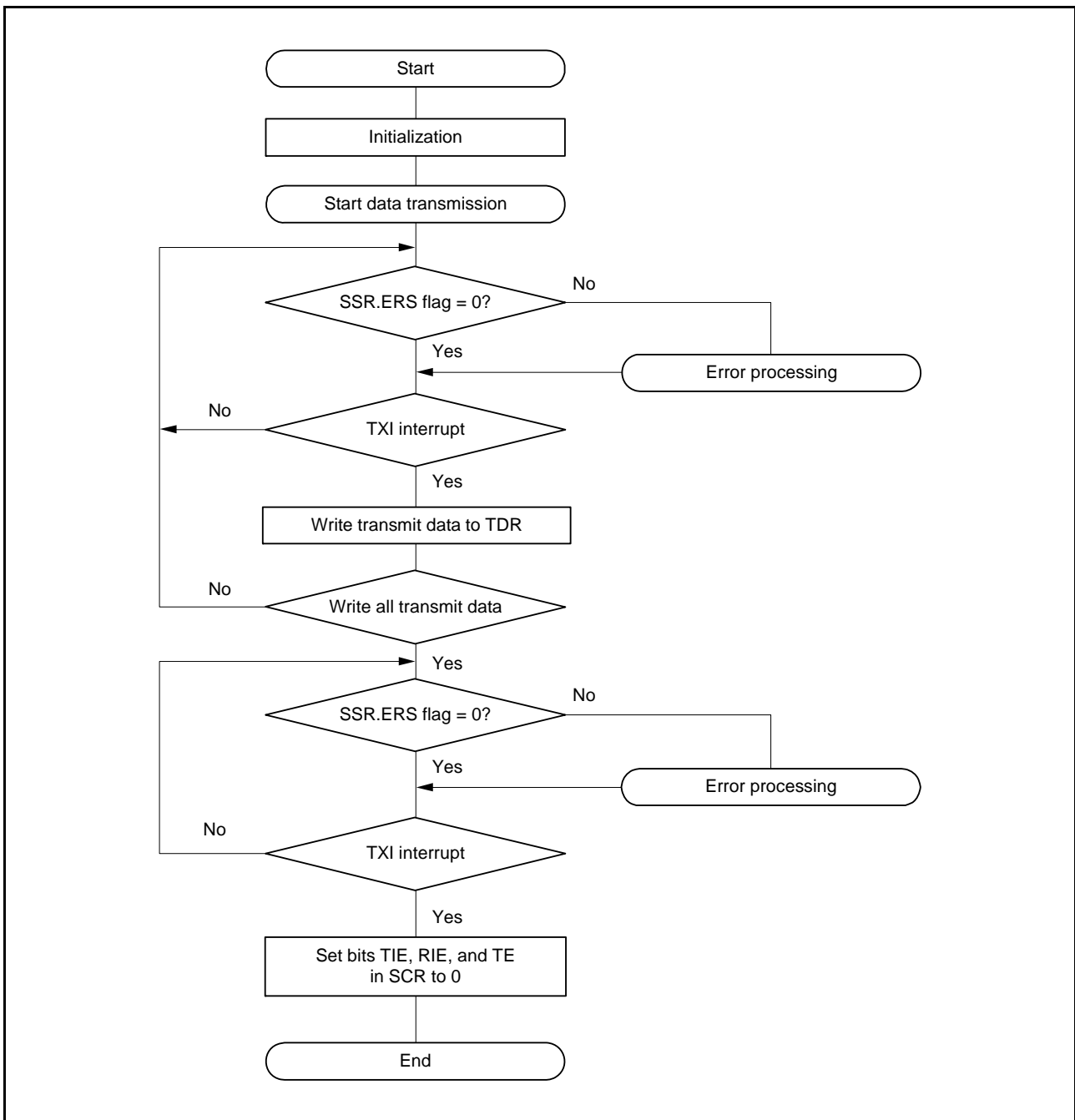


Figure 27.41 Sample Smart Card Interface Transmission Flowchart

### 27.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 27.42 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 27.43 shows a sample flowchart for serial data reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DTC.

In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DTC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DTC activation beforehand, allowing transfer of receive data.

If an error occurs during reception and either the ORER or PER flag in 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 is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC is transferred.

Even if a parity error occurs and the PER flag is set to 1 during reception, receive data is transferred to RDR, thus allowing the data to be read.

Note 1. For operations in block transfer mode, refer to section 27.3, Operation in Asynchronous Mode.

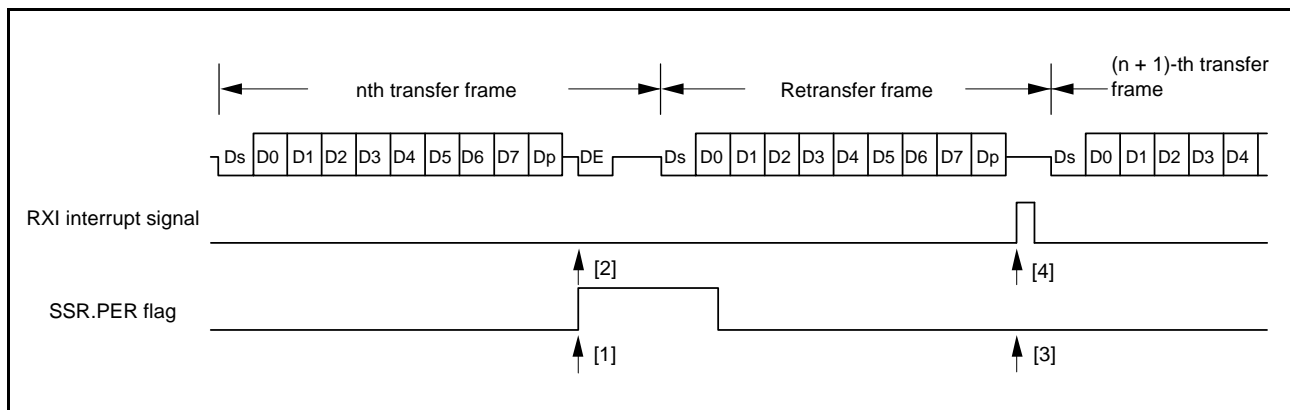


Figure 27.42 Data Retransfer Operation in SCI Reception Mode (Data Retransfer Operation during Reception)

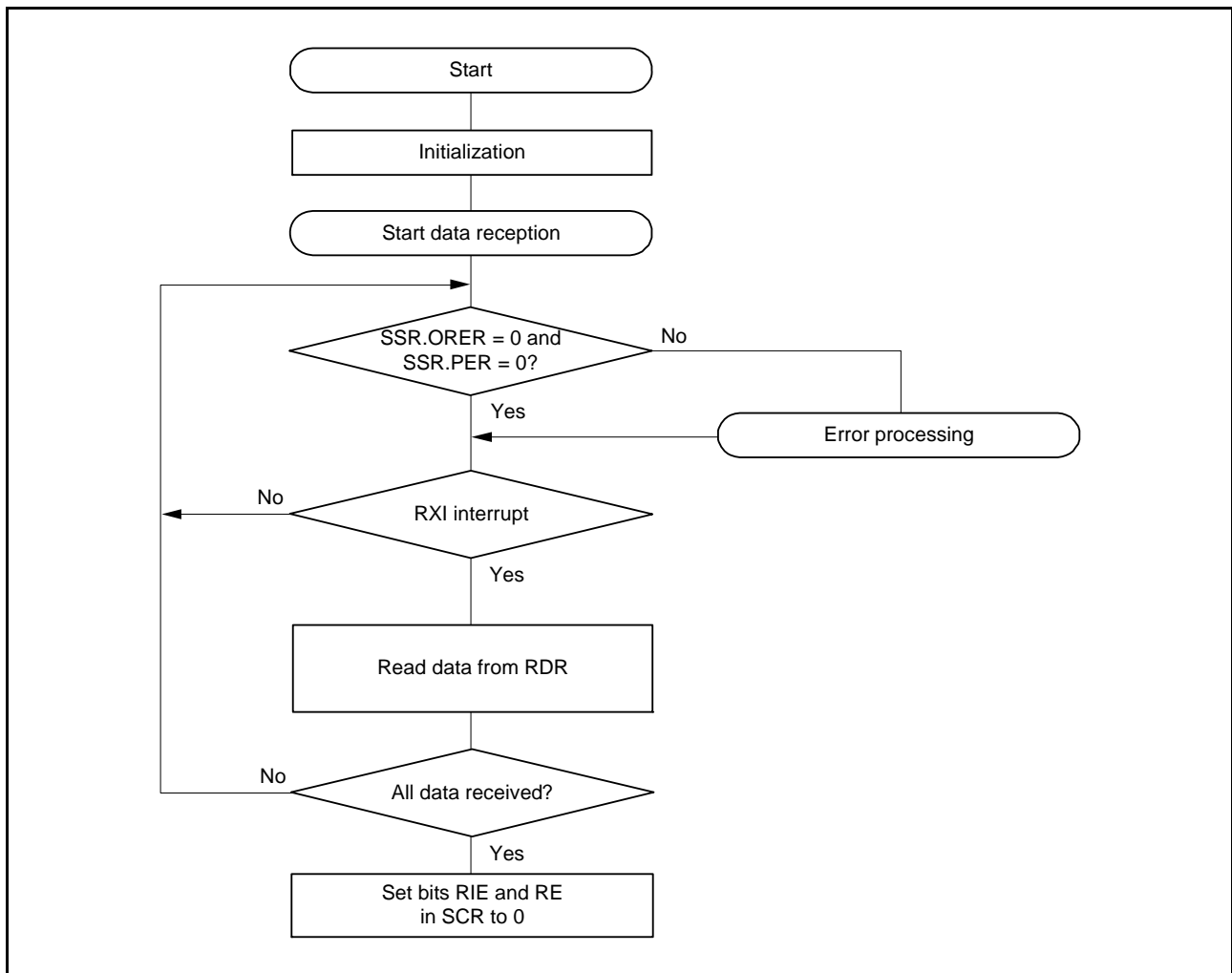
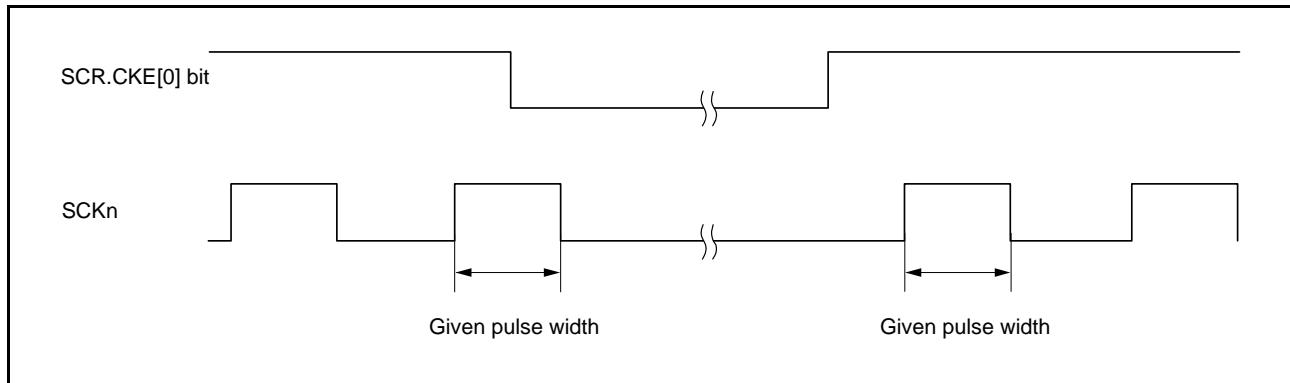


Figure 27.43 Sample Smart Card Interface Reception Flowchart

### 27.6.8 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 27.44 shows an example of clock output fixing timing when the CKE[0] bit is controlled with GM = 1 and CKE[1] = 0.



**Figure 27.44** Clock Output Fixing Timing

At power-on, 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.

### 27.7 Operation in Simple I<sup>2</sup>C Mode

Simple I<sup>2</sup>C bus format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device is able to specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied.

The 8 data bits in all frames are transmitted in order from the MSB.

The I<sup>2</sup>C format and timing of the I<sup>2</sup>C bus are shown in Figure 27.45 and Figure 27.46.

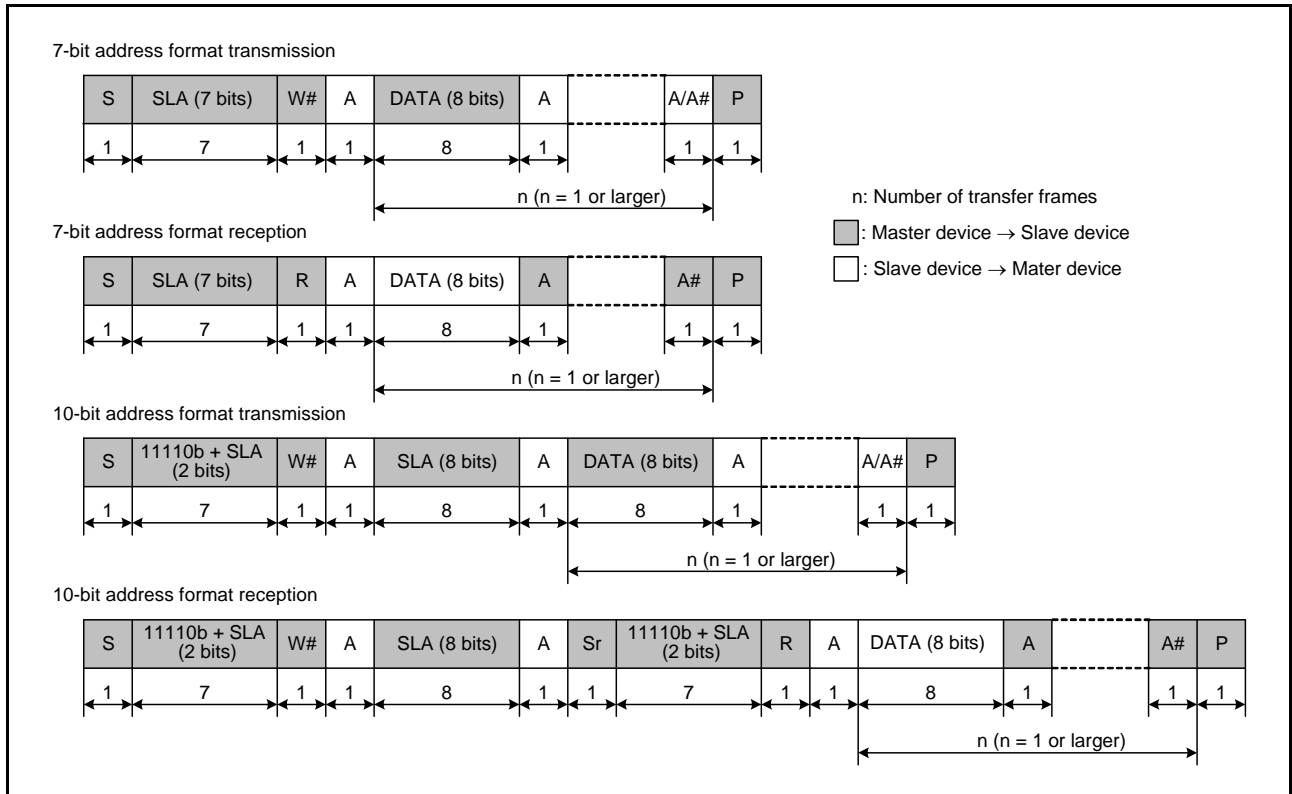


Figure 27.45 I<sup>2</sup>C Bus Format

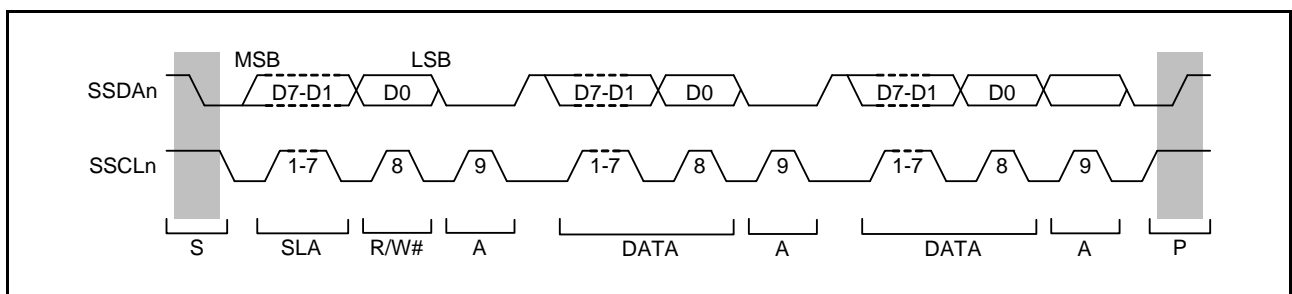


Figure 27.46 I<sup>2</sup>C Bus Timing (When SLA is 7 Bits)

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.

### 27.7.1 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 set (to 0), and a start-condition generated interrupt is output.

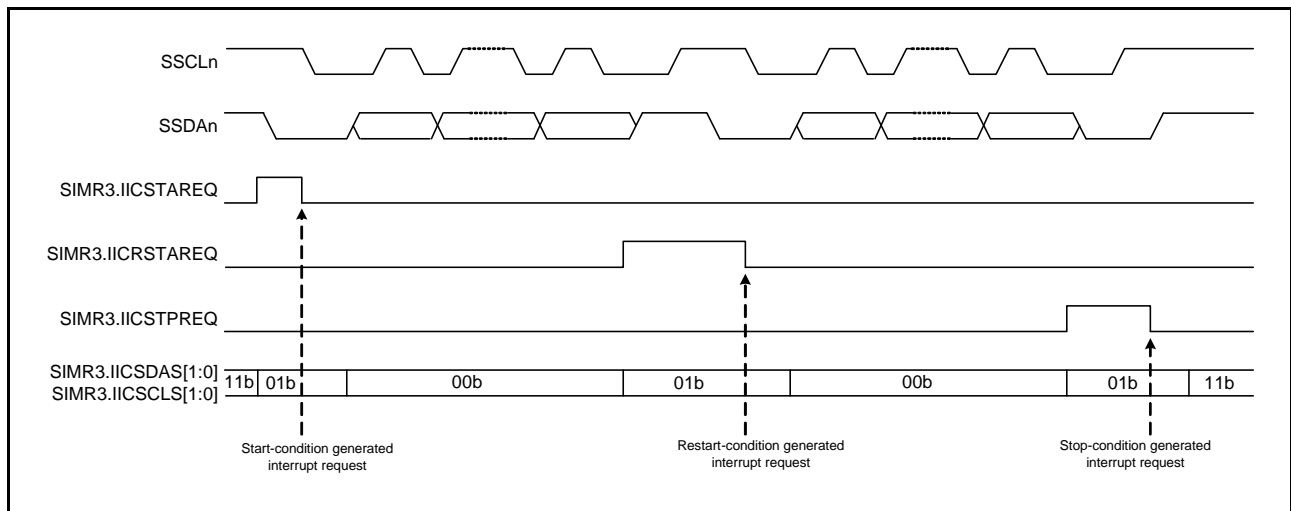
Writing 1 to the IICRSTAREQ bit in SIMR3 causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- 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 set (to 0), and a restart-condition generated interrupt is output.

Writing 1 to the IICSTPREQ 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 IICSTPREQ bit in SIMR3 is set (to 0), and a stop-condition generated interrupt is output.

Figure 27.47 shows the timing of operations in the generation of start, restart, and stop conditions.



**Figure 27.47** Timing of Operations in the Generation of Start, Restart, and Stop Conditions

### 27.7.2 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 output, delay for noise filtering of the input on the SSCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SSCLn clock is extended even if other devices are not placing the low level on the SSCLn line. If the 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 27.48 shows an example of operations to synchronize the clocks.

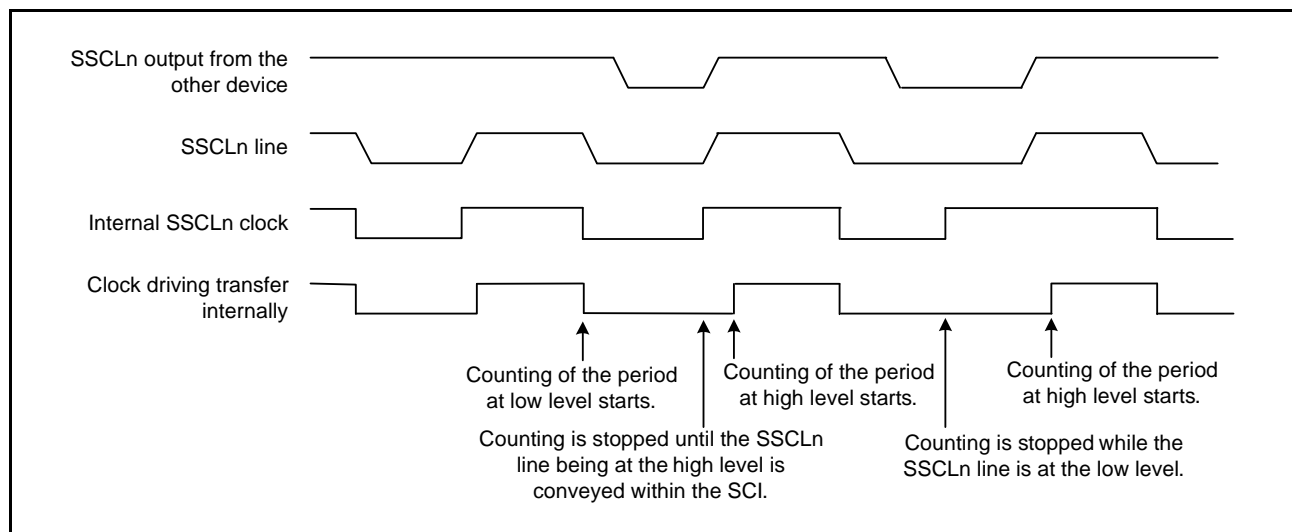


Figure 27.48 Example of Operations for Clock Synchronization



### 27.7.3 SSDA Output Delay

The IICDL[4:0] bits in SIMR1 can be used to set a delay for output on the SSDAn pin relative to falling edges of output on the SSCLn pin. Delay-time settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, PCLK, by the divisor selected by the 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 SSDAn 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 27.49 shows the timing of delays in SSDA output.

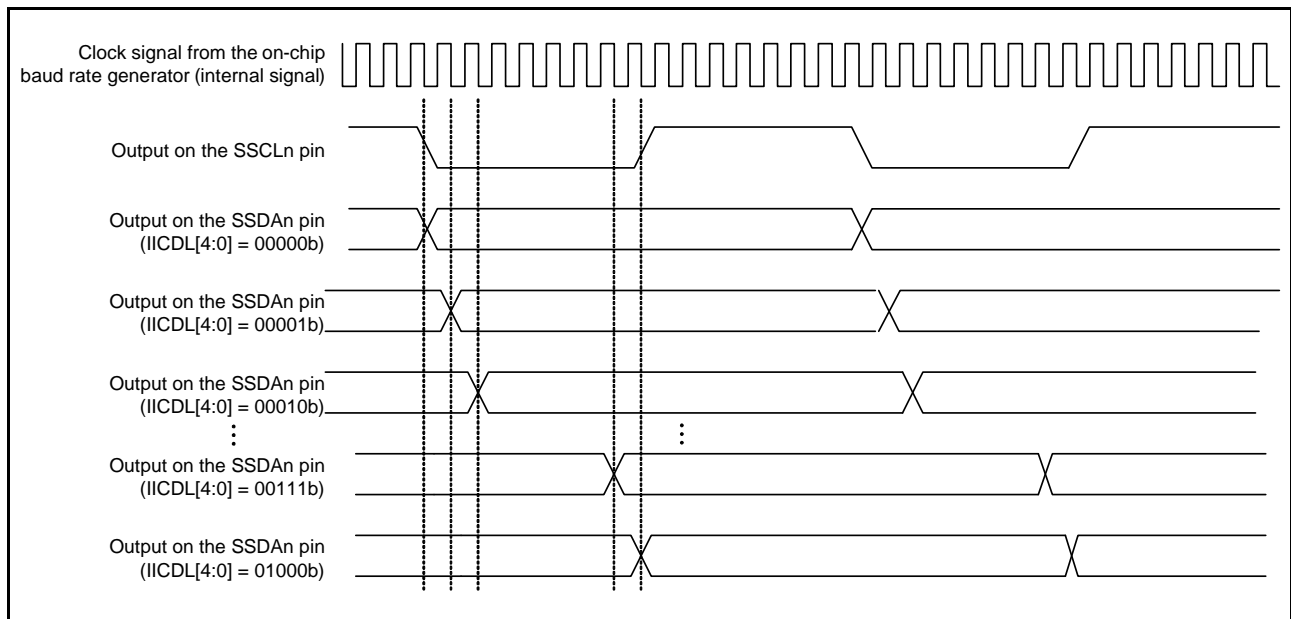


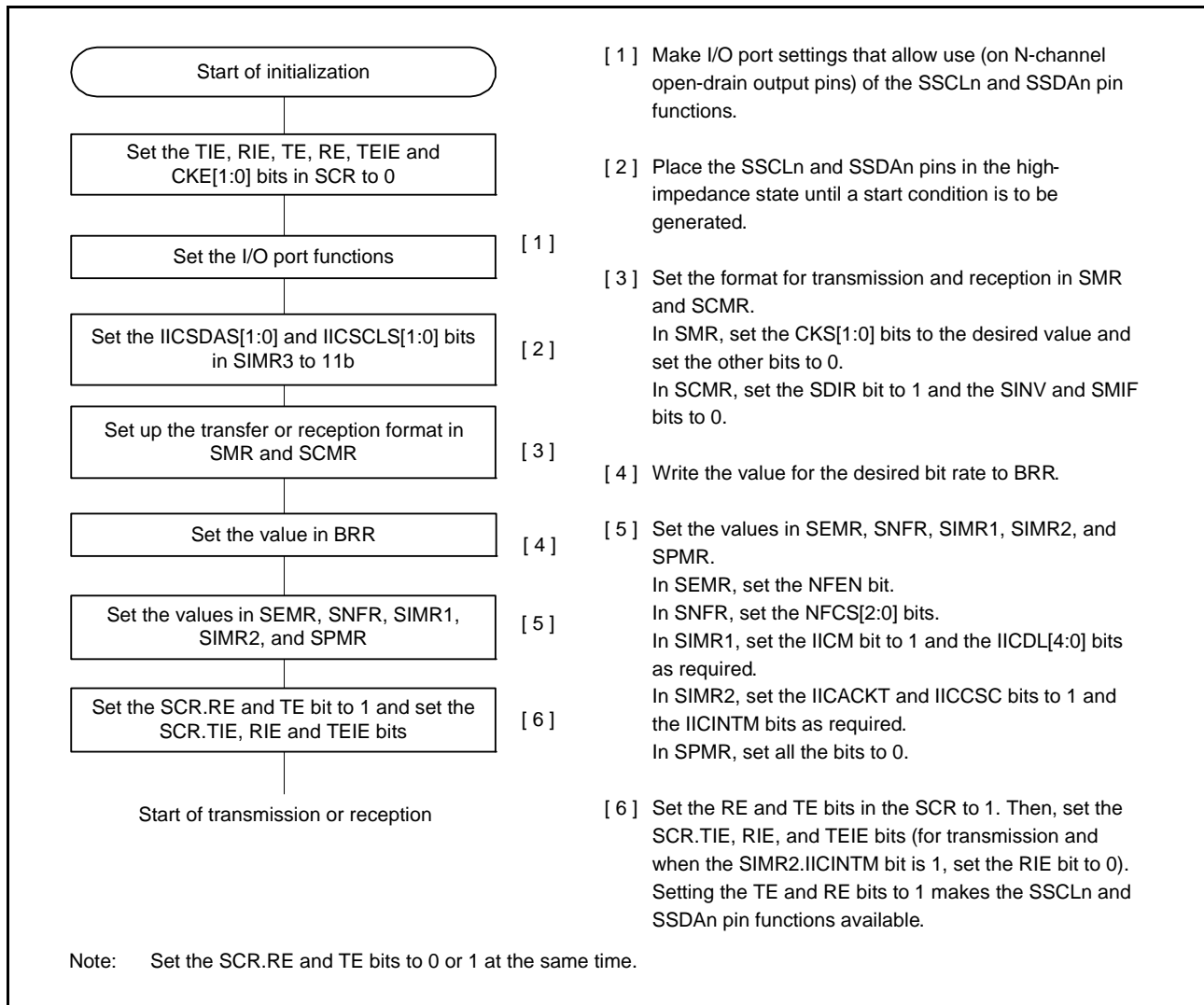
Figure 27.49 Timing of Delays in SSDA Output

### 27.7.4 SCI Initialization (Simple I<sup>2</sup>C Mode)

Before transferring data, write the initial value (00h) to SCR and initialize the interface following the example shown in Figure 27.50.

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.



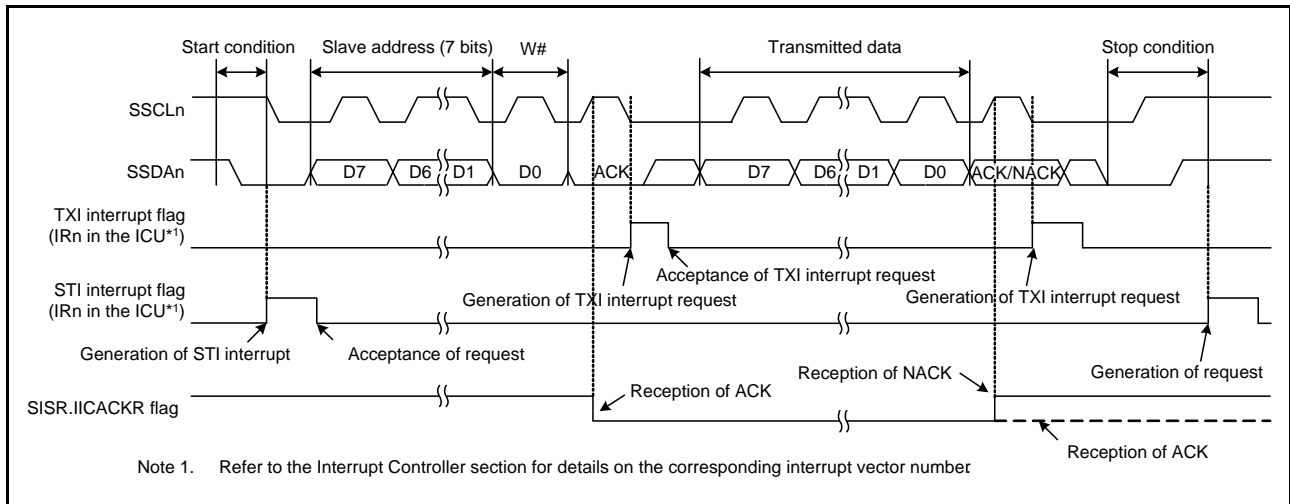
**Figure 27.50 Example of the Flowchart of SCI Initialization (for Simple I<sup>2</sup>C Mode)**

### 27.7.5 Operation in Master Transmission (Simple I<sup>2</sup>C Mode)

Figure 27.51 and Figure 27.52 show examples of operations in master transmission and Figure 27.53 is a flowchart showing the procedure for data transmission. The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and the value of the SCR.RIE bit is assumed to be 0 (RXI and ERI interrupt requests are disabled). See Table 27.32 for more information on the STI interrupt.

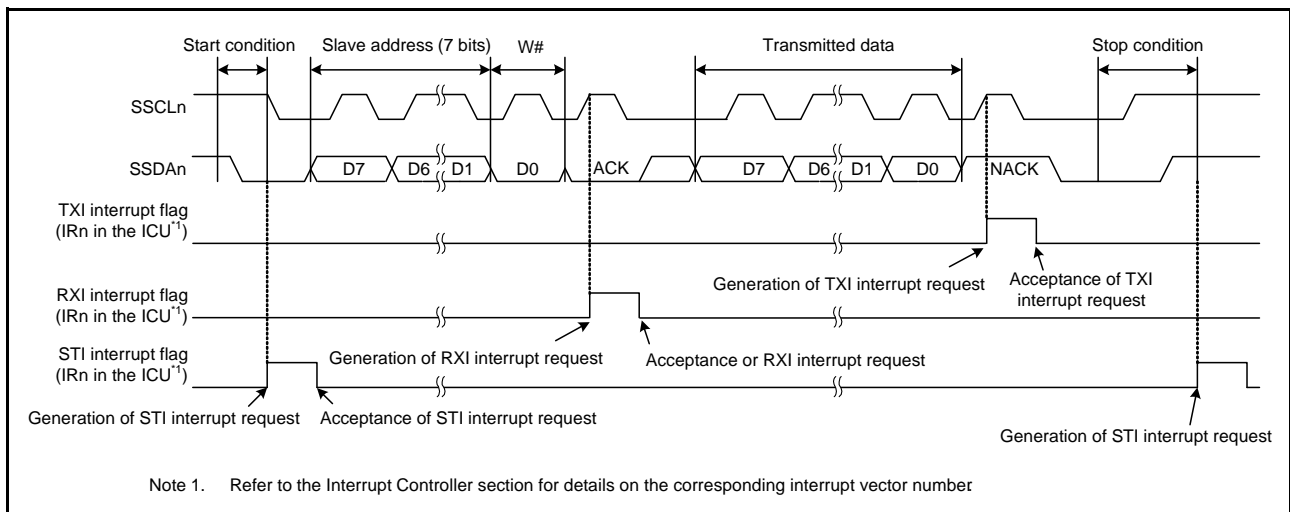
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 27.53 are repeated twice.

In simple I<sup>2</sup>C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.

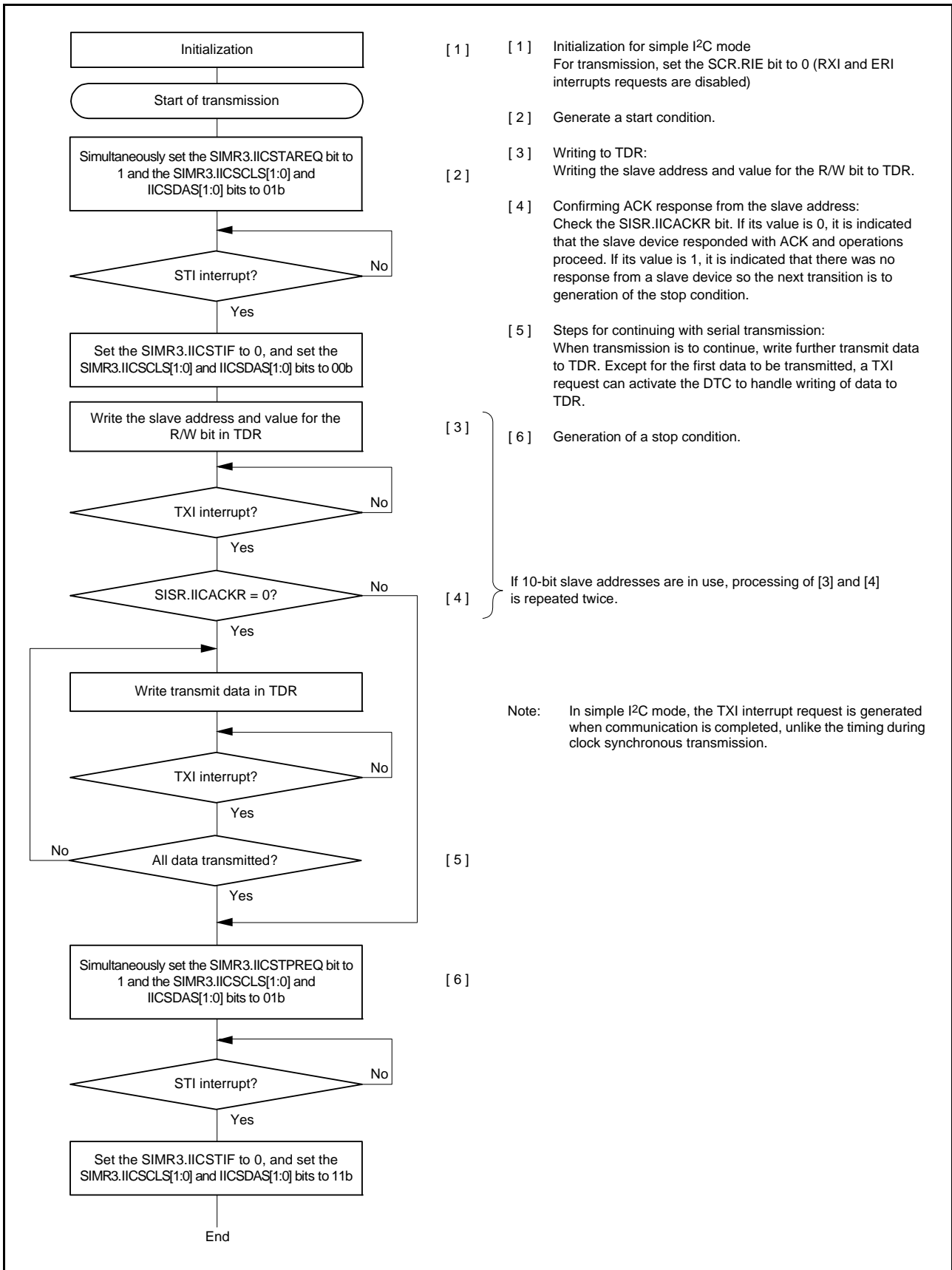


**Figure 27.51 Example 1 of Operations for Master Transmission in Simple I<sup>2</sup>C Bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)**

When the SIMR2.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC is activated by the ACK interrupt as the trigger and necessary number of data bytes are transmitted. When the NACK is received, error processing, such as transmission stop and retransmission, is performed by the NACK interrupt as the trigger.



**Figure 27.52 Example 2 of Operations for Master Transmission in Simple I<sup>2</sup>C Bus Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use)**



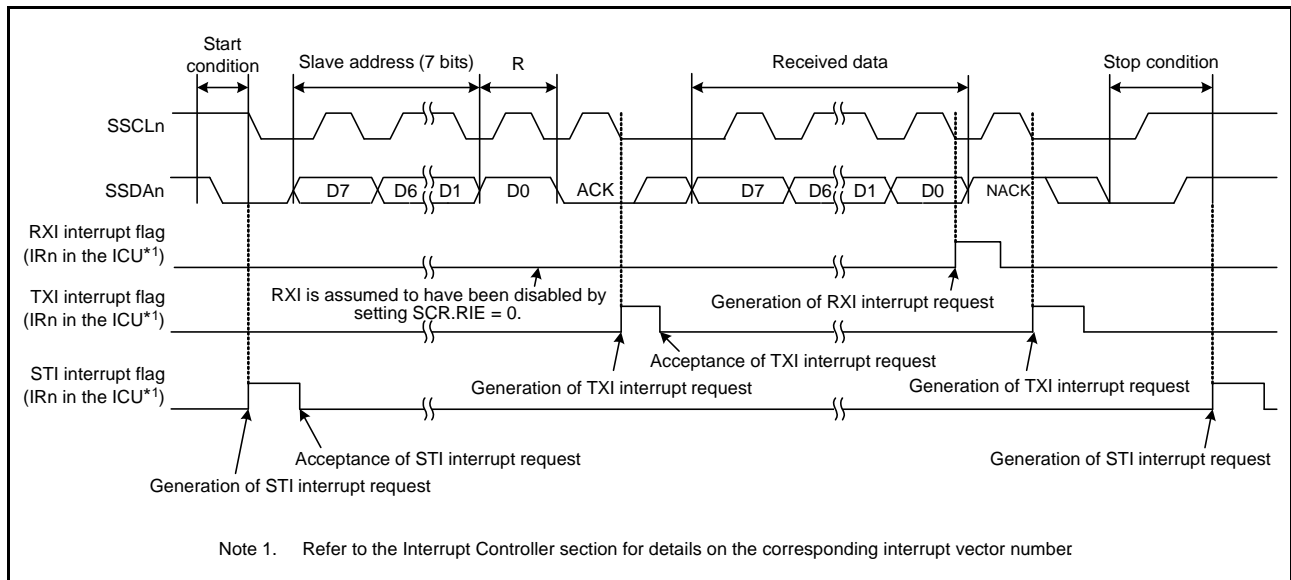
**Figure 27.53 Example of the Procedure for Master Transmission Operations in Simple I<sup>2</sup>C Mode (with Transmission Interrupts and Reception Interrupts in Use)**

### 27.7.6 Master Reception (Simple I<sup>2</sup>C Mode)

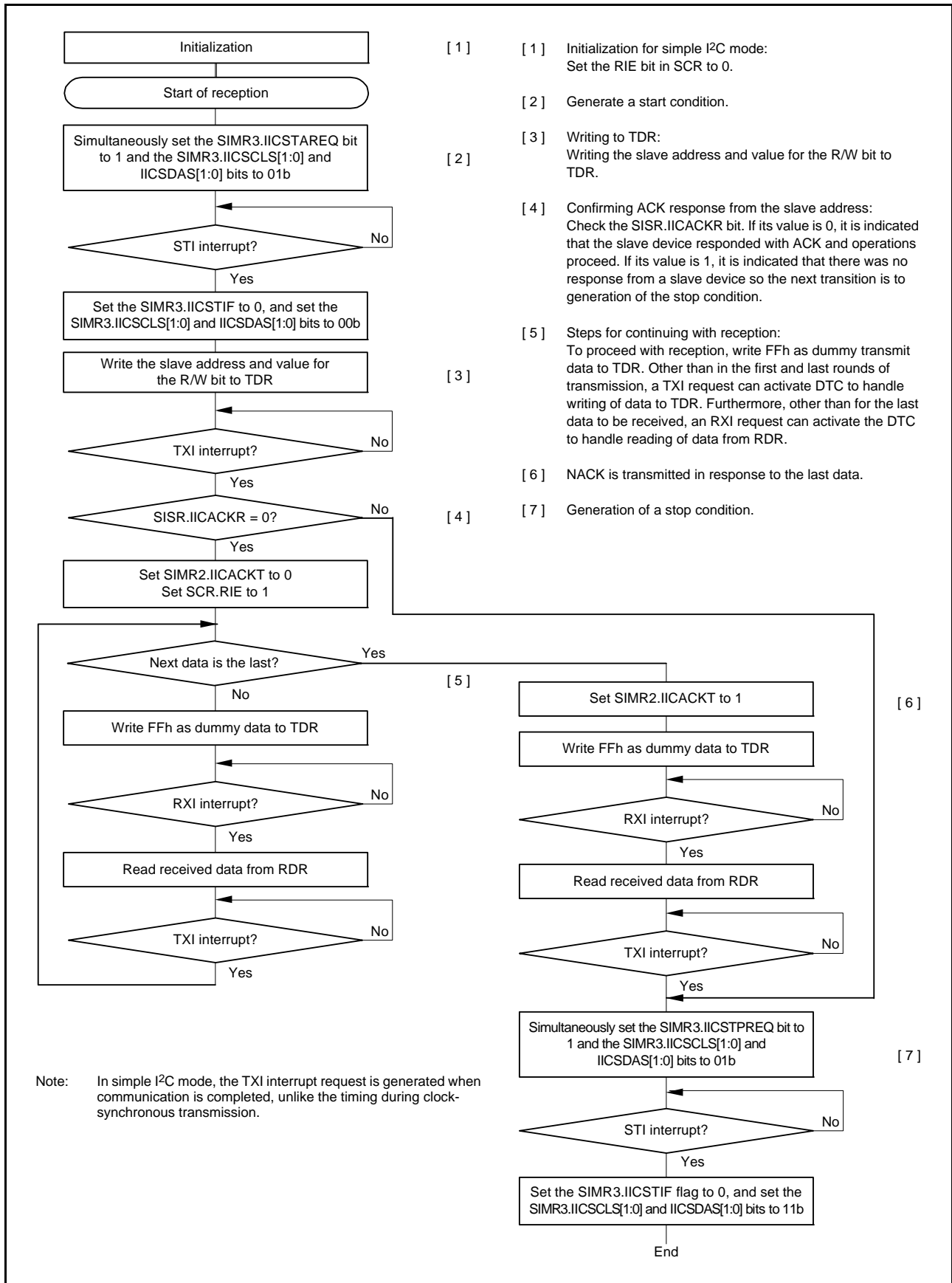
Figure 27.54 shows an example of operations in simple I<sup>2</sup>C mode master reception and Figure 27.55 is a flowchart showing the procedure for master reception.

The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts).

In simple I<sup>2</sup>C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.



**Figure 27.54 Example of Operations for Master Reception in Simple I<sup>2</sup>C Bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)**



**Figure 27.55 Example of the Procedure for Master Reception Operations in Simple I<sup>2</sup>C Mode (with Transmission Interrupts and Reception Interrupts in Use)**

### 27.8 Operation in Simple SPI Mode

As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Making the settings for clock synchronous mode (SCMR.SMIF = 0, SIMR1.IICM = 0, SMR.CM = 1) plus setting the SSE bit in the SPMR 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 27.56 shows an example of connections for simple SPI mode. Control a general port pin to produce the SS output signal from the master.

In simple SPI mode, data are transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended to this. The data can be inverted by setting the SCMR.SINV bit to 1.

Since the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both the transmitter and receiver have a double-buffered structure, writing of further transmit data while transmission is in progress and reading of previously received data while reception is in progress are both possible. Continuous transfer is thus possible.

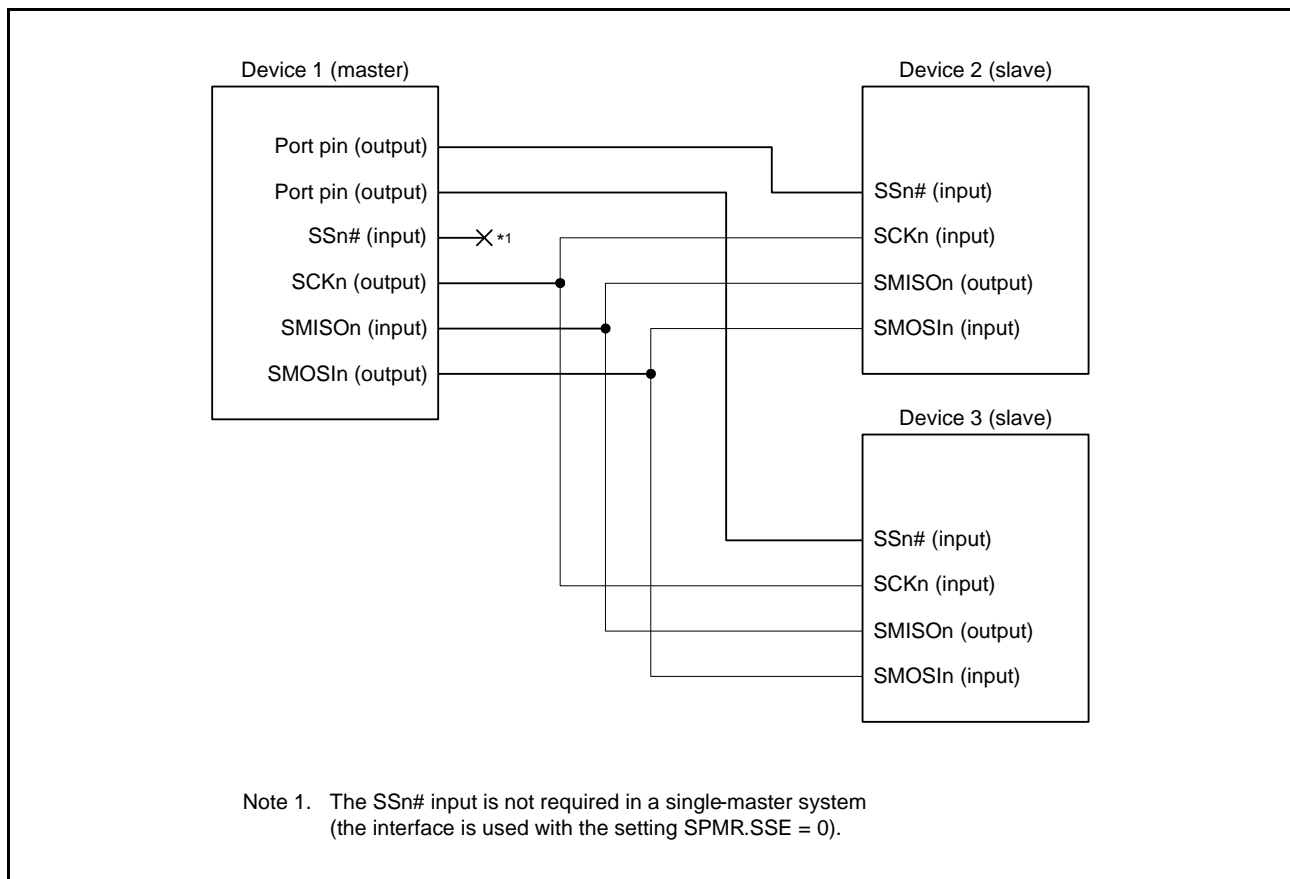


Figure 27.56 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

### 27.8.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (SCR.CKE[1:0] = 00b or 01b and SPMR.MSS = 0) or slave (SCR.CKE[1:0] = 10b or 11b and SPMR.MSS = 1).

Table 27.28 lists the states of pins according to the mode and the level on the SSn# pin.

**Table 27.28 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* <sup>1</sup>	High level (transfer can proceed)	Output for data transmission* <sup>2</sup>	Input for received data	Clock output* <sup>3</sup>
	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 serial transmission is disabled (SCR.TE bit = 0).

Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE and RE bits = 00b) in a multi-master configuration (SPMR.SSE = 1).

### 27.8.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the SCR to 00b 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 pin output will be placed in the high-impedance state after the completion of the transfer.

Control a general port pin to produce the SS output signal from the master.

### 27.8.3 SS Function in Slave Mode

Setting the CKE[1:0] bits in the SCR to 10b and the MSS bit in the SPMR to 1 selects slave operation. When the level on the SSn# pin is high, the SMISOn output pin will be in the high-impedance state and clock input through the SCKn pin will be ignored. When the level on the SSn# pin is low, clock input through the SCKn pin will be effective and transmission or reception can proceed.

If the input on the SSn# pin changes from low to high level during transmission or reception, the SMISOn output pin will be placed in the high-impedance state. Meanwhile, the internal processing for transmission or reception will continue at the rate of the clock input through the SCKn pin until processing for the character currently being transmitted or received is completed, after which it stops. At that time, an interrupt (the appropriate one from among TXI, RXI, and TEI) will be generated.



### 27.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in the SPMR 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 27.57. 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, refer to section 27.8.2, SS Function in Master Mode.

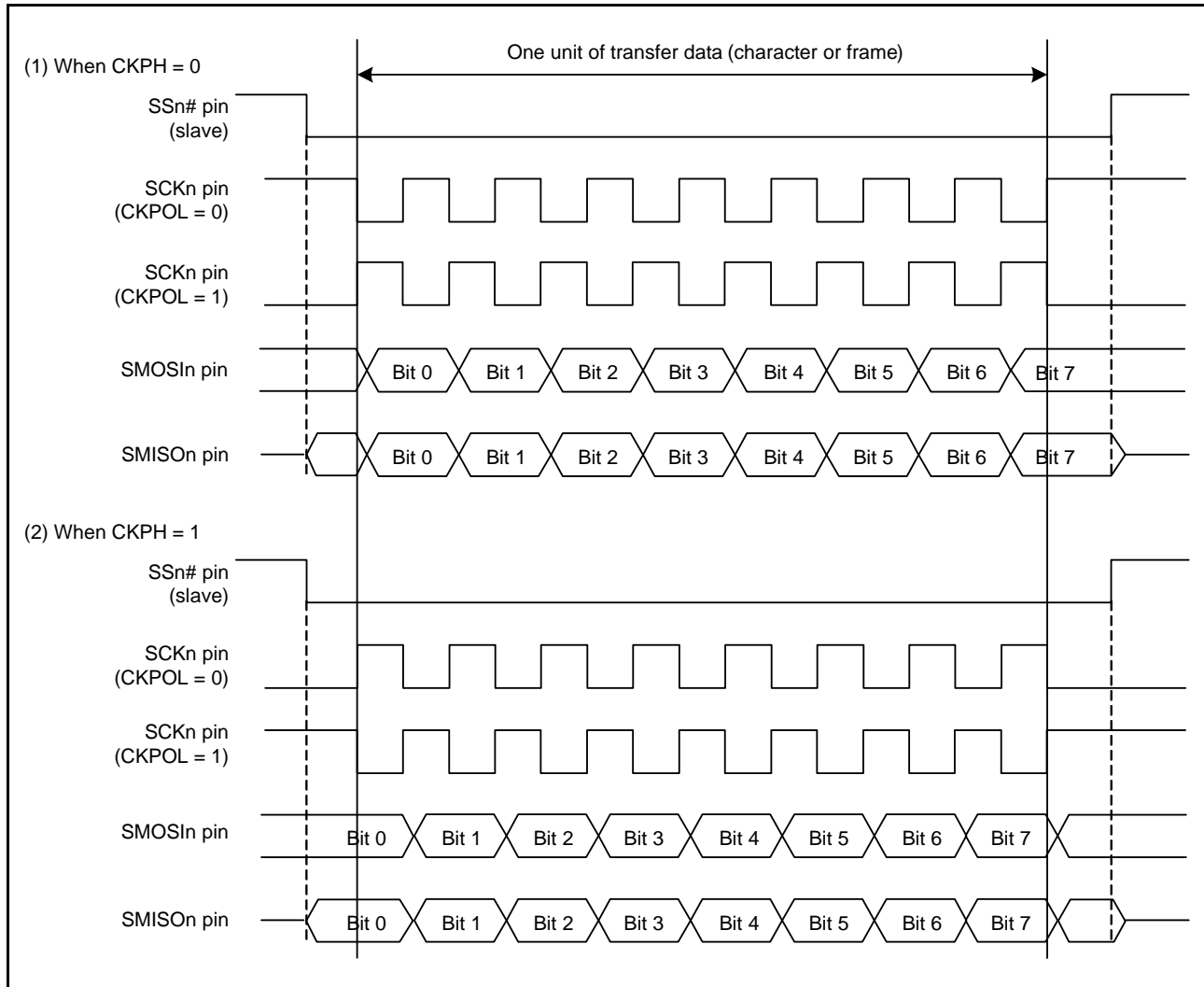


Figure 27.57 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

### 27.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock synchronous mode Figure 27.24, 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.

As well as setting the RE bit to 0, note that the SSR.ORER, FER, and PER flags, as well as the RDR, are not initialized. 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 transmit data empty interrupt (TXI) if the value of the TIE bit in the SCR is 1 at the time.

### 27.8.6 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.

## 27.9 Extended Serial Mode Control Section: Description of Operation

### 27.9.1 Serial Transfer Protocol

In conjunction with the SCIe module, the extended serial mode control section of the SCIf module can realize the serial transfer protocol composed of Start Frames and Information Frames that is shown in Figure 27.58.

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.

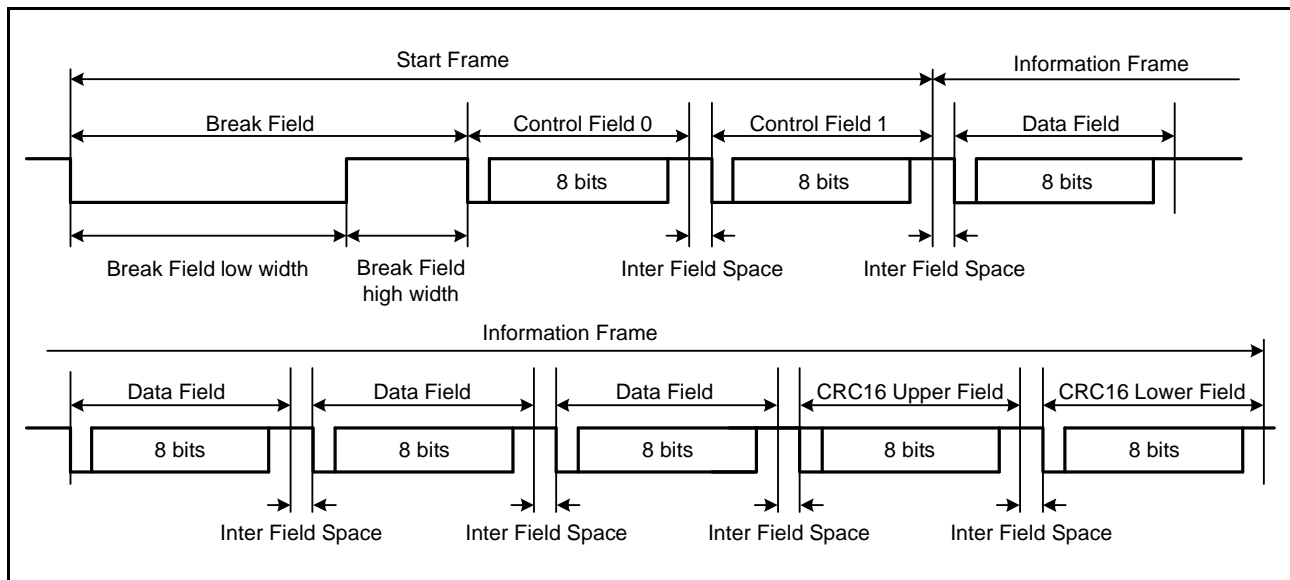


Figure 27.58 Protocol for Serial Transfer by the Extended Serial Mode Control Section

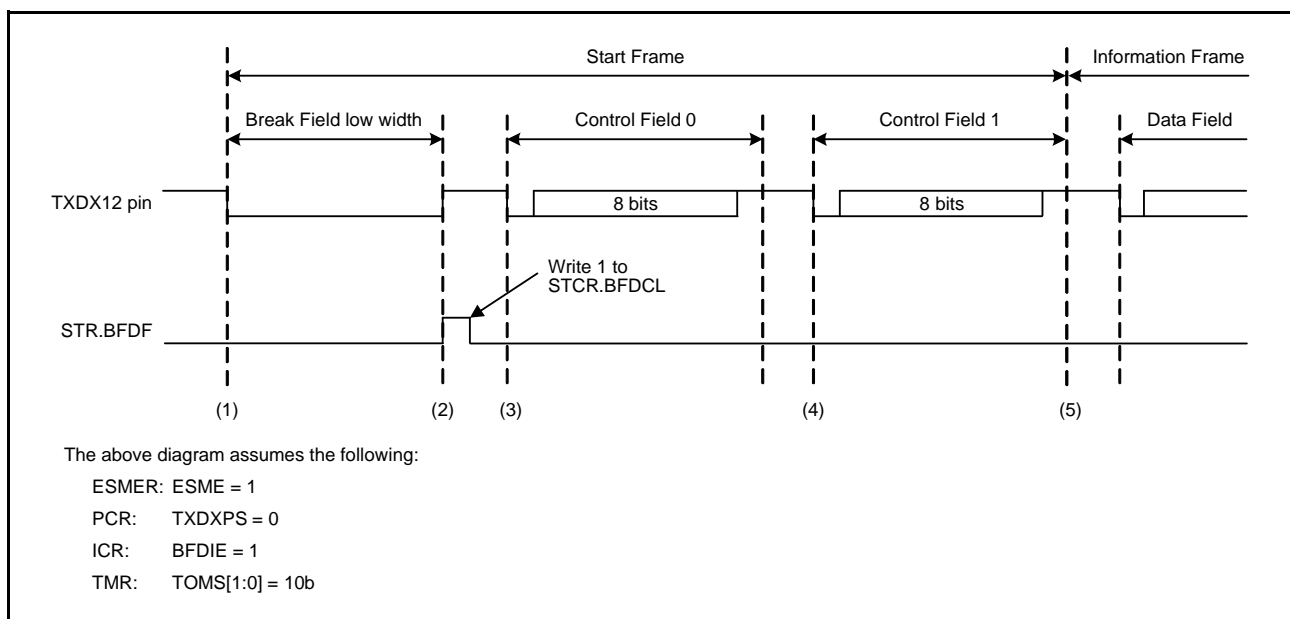
## 27.9.2 Transmitting a Start Frame

Figure 27.59 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 27.60 and Figure 27.61 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 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.
- (3) Writing 0 to the TCST bit in TCR stops counting by the timer, and SCI12 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, SCI12 is used to send the data for Control Field 1.
- (5) Once the data for Control Field 1 have been transmitted, SCI12 is used to send an Information Frame.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.



**Figure 27.59** Example of Operations When Transmitting a Start Frame

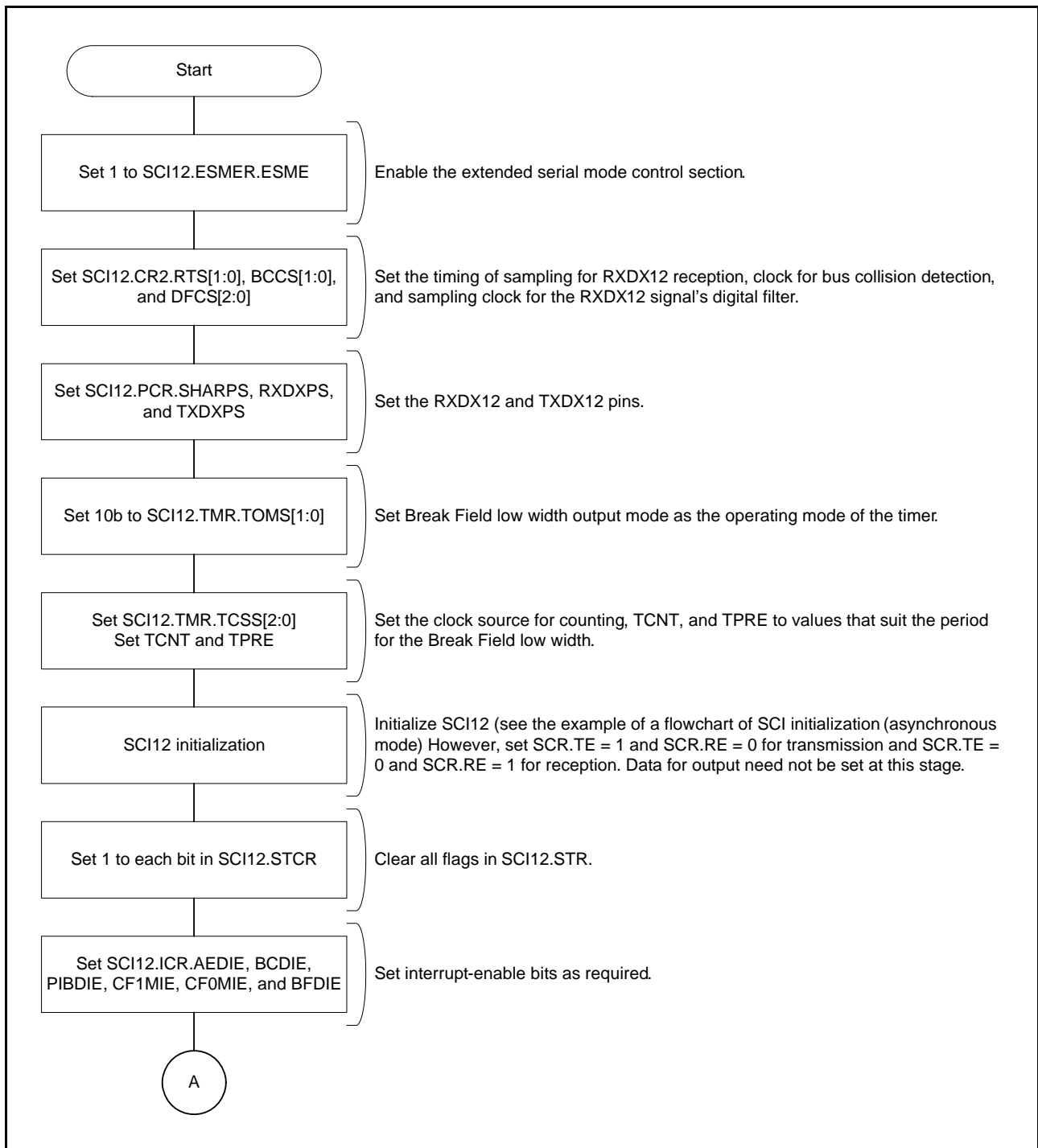


Figure 27.60 Example of Start Frame Transmission (1/2)

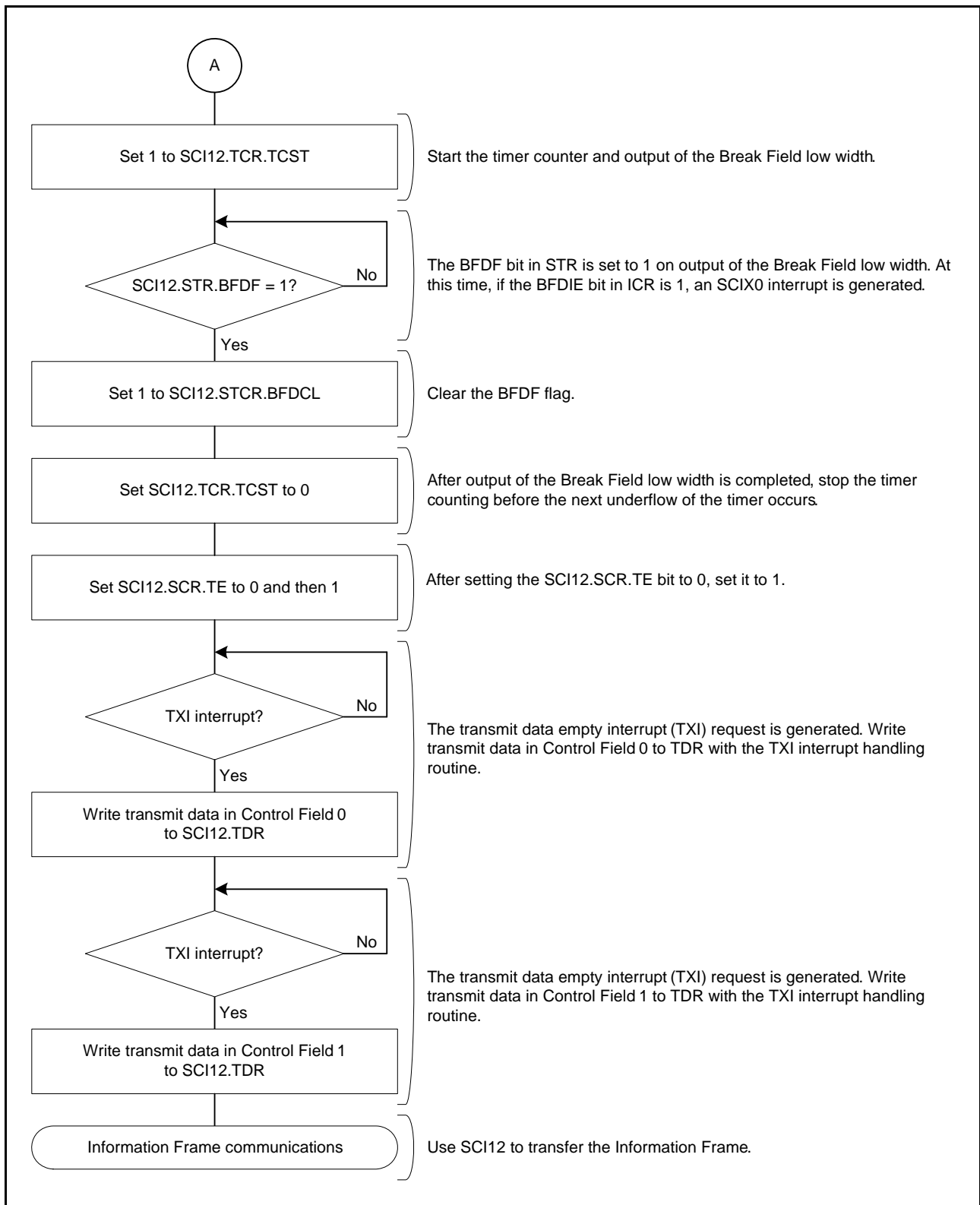


Figure 27.61 Example of Start Frame Transmission (2/2)

### 27.9.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 27.29.

**Table 27.29 Structures of Start Frames**

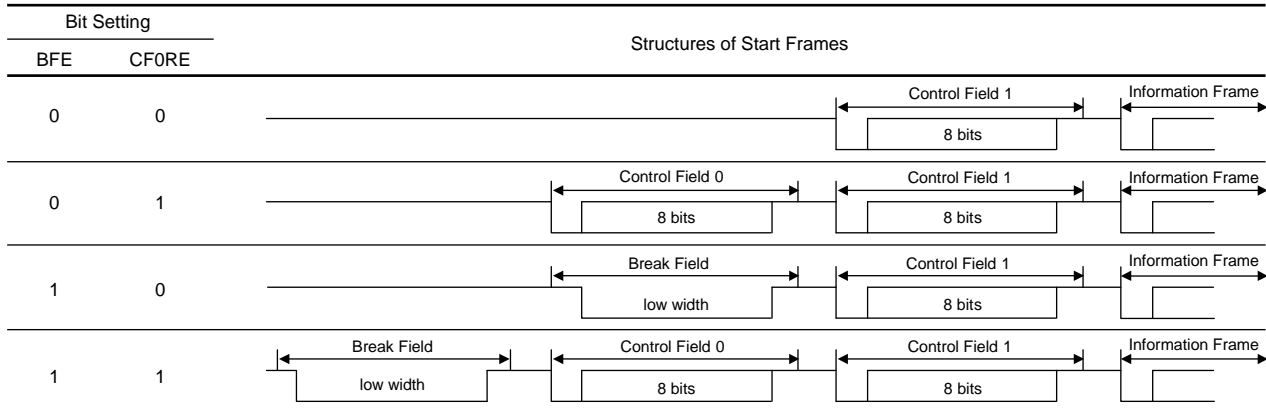
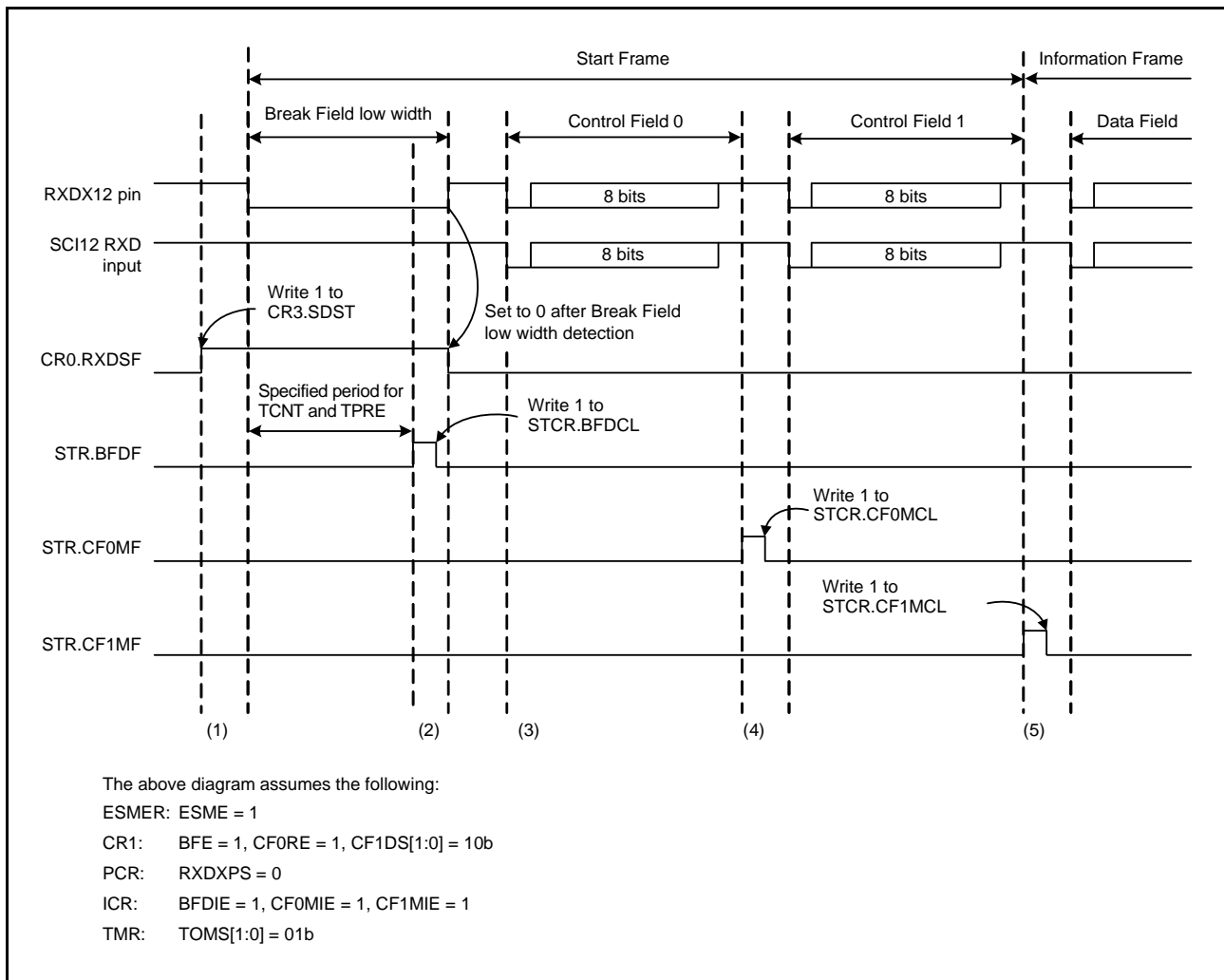


Figure 27.62 shows an example of operations to receive a Start Frame, which is composed of the Break Field low width, Control Field 0, and Control Field 1. Figure 27.63 and Figure 27.64 are flowcharts for the reception of a Start Frame, and Figure 27.65 is a state transition diagram for the extended 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 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 0 and reception of Control Field 0 by the SCI12 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 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 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 27.62 Example of Operations at the Time of Start Frame Reception**



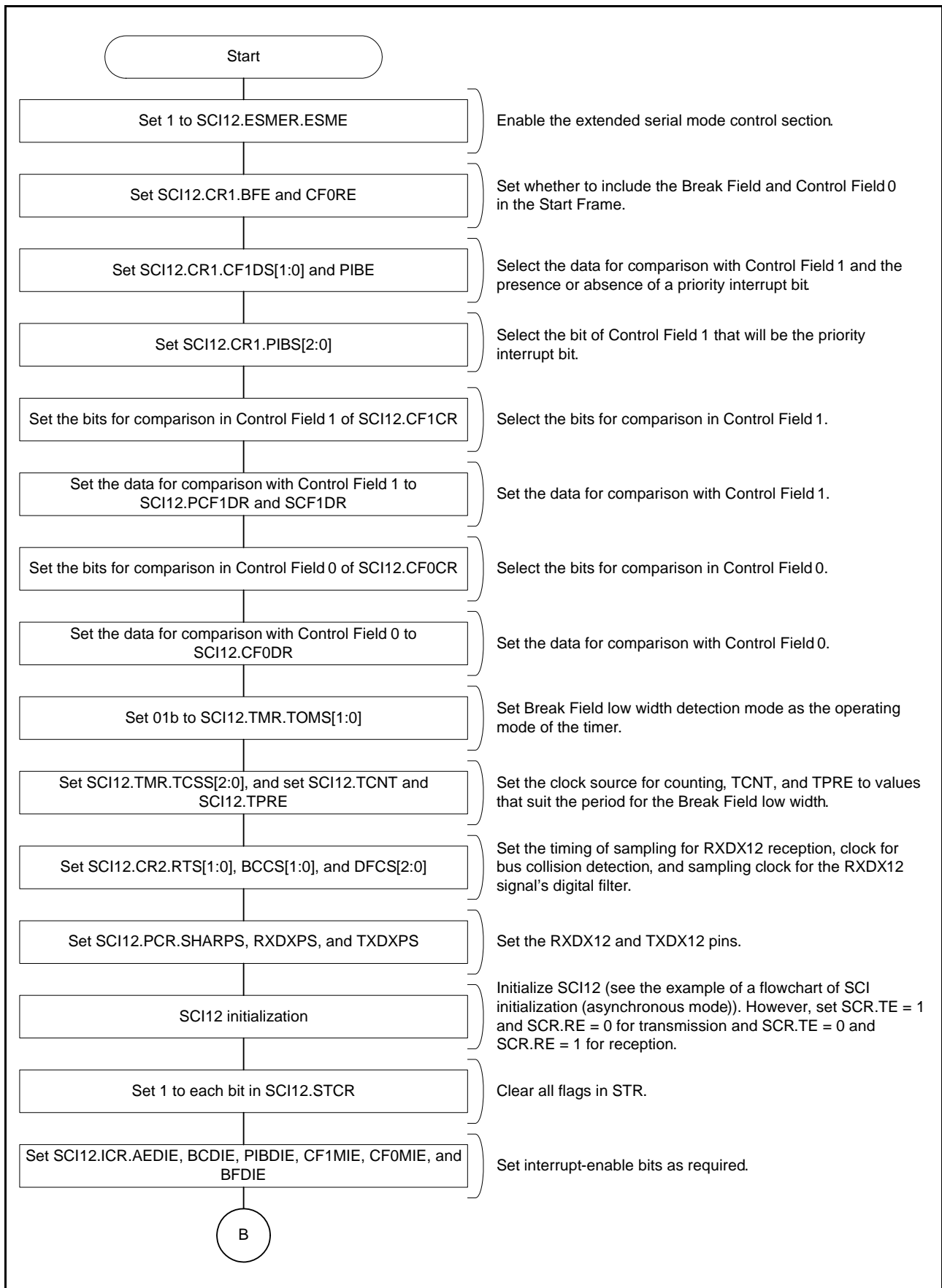


Figure 27.63 Sample Flowchart for Reception of a Start Frame (1)

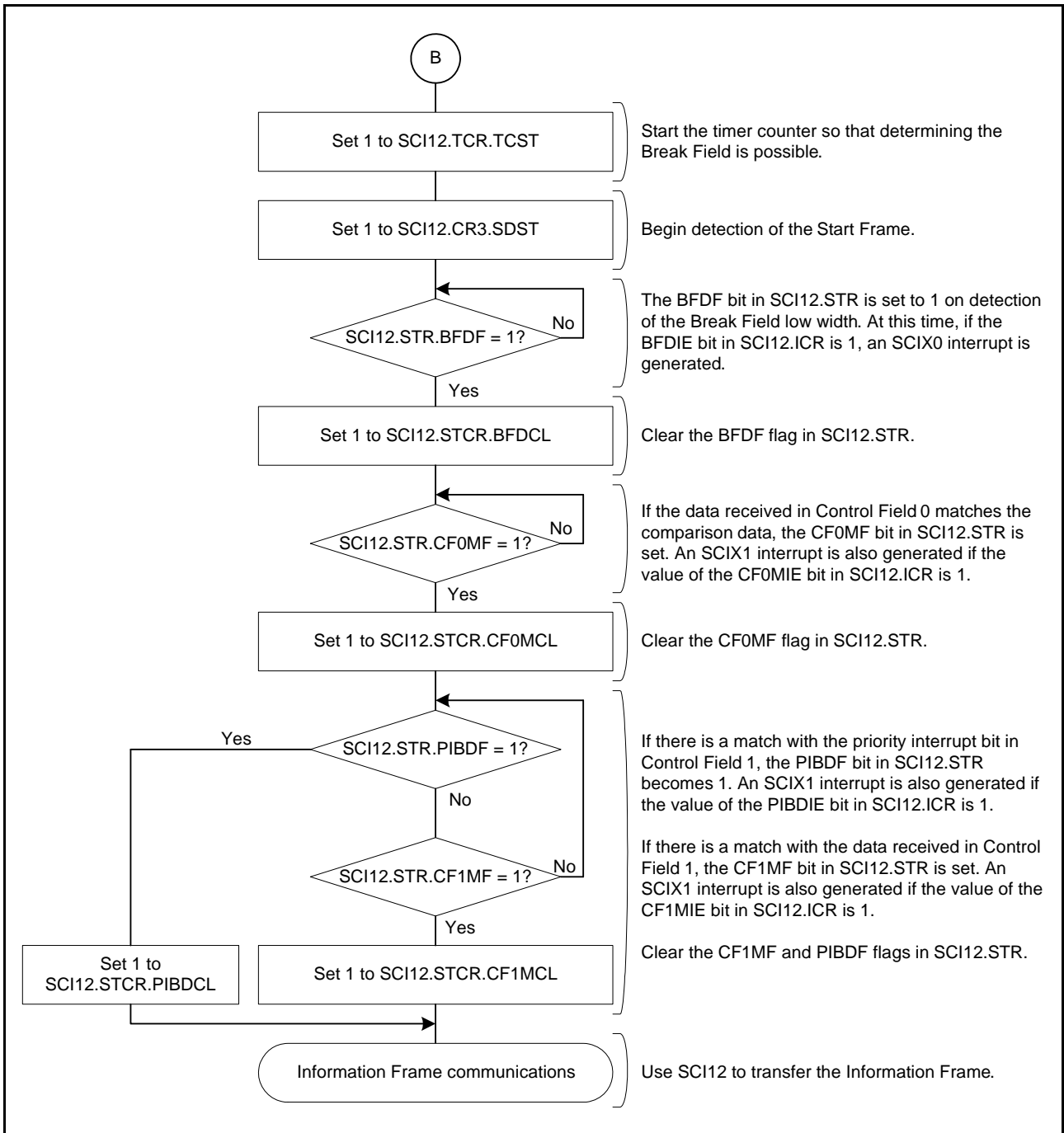


Figure 27.64 Sample Flowchart for Reception of a Start Frame (2)

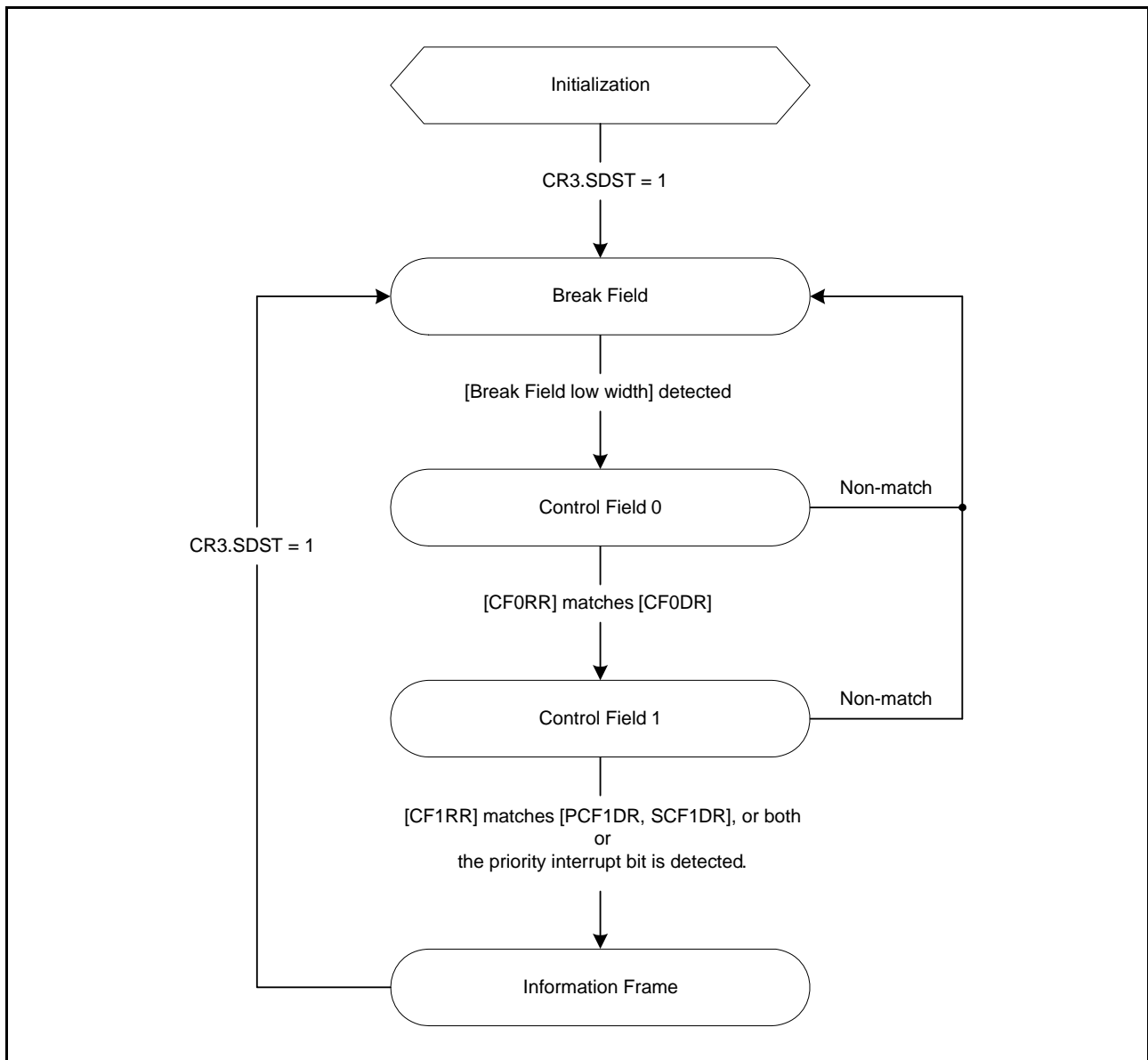


Figure 27.65 State Transitions When Receiving a Start Frame

### 27.9.3.1 Priority Interrupt Bit

Figure 27.66 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 27.62, 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 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.

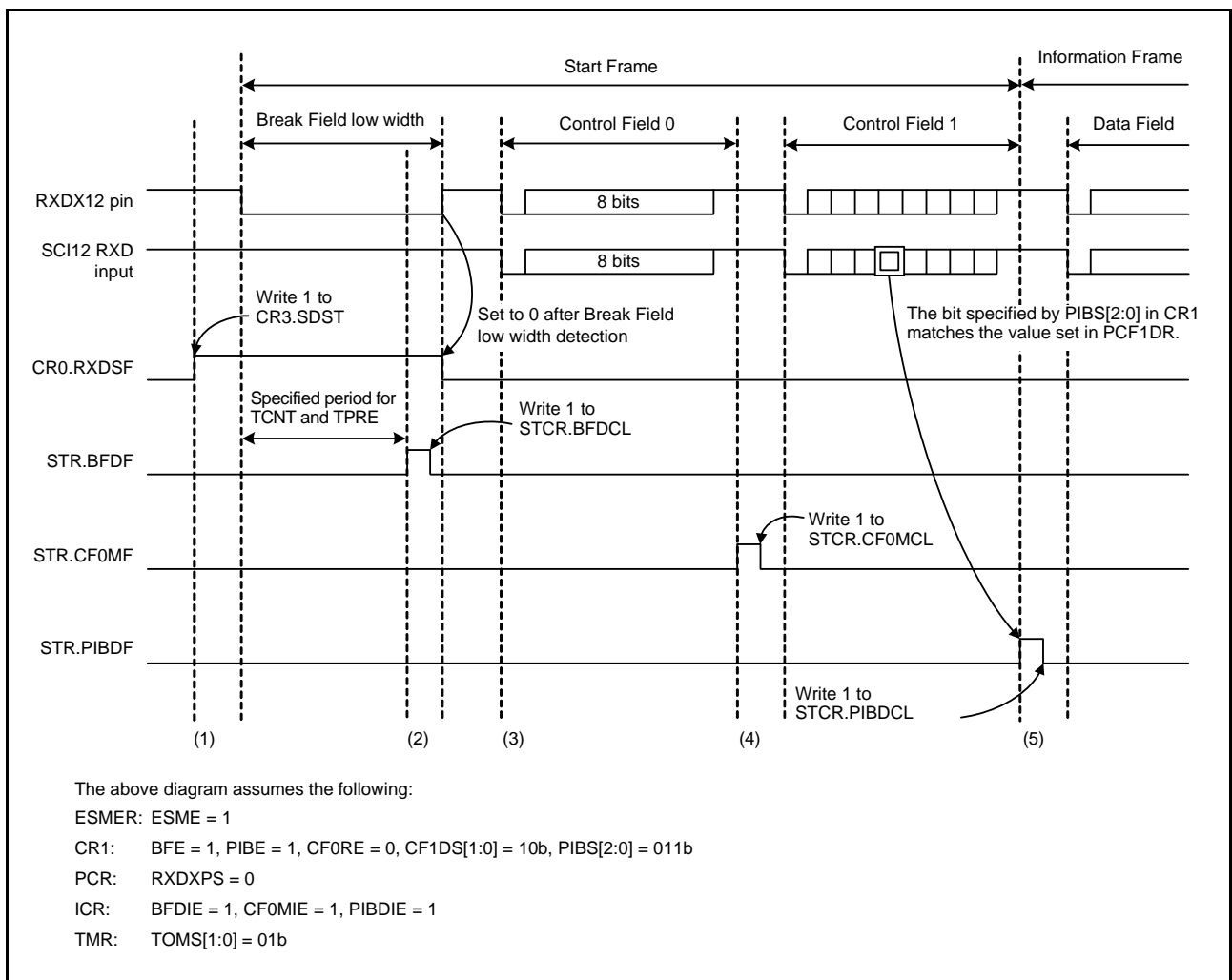


Figure 27.66 Example of Operations When Receiving a Start Frame While the CR1.PIBE Bit is 1

### 27.9.4 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 SCI2 are in progress when the ESMER.ESME and the SCI2.SCI.TE are set to 1.

Figure 27.67 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.

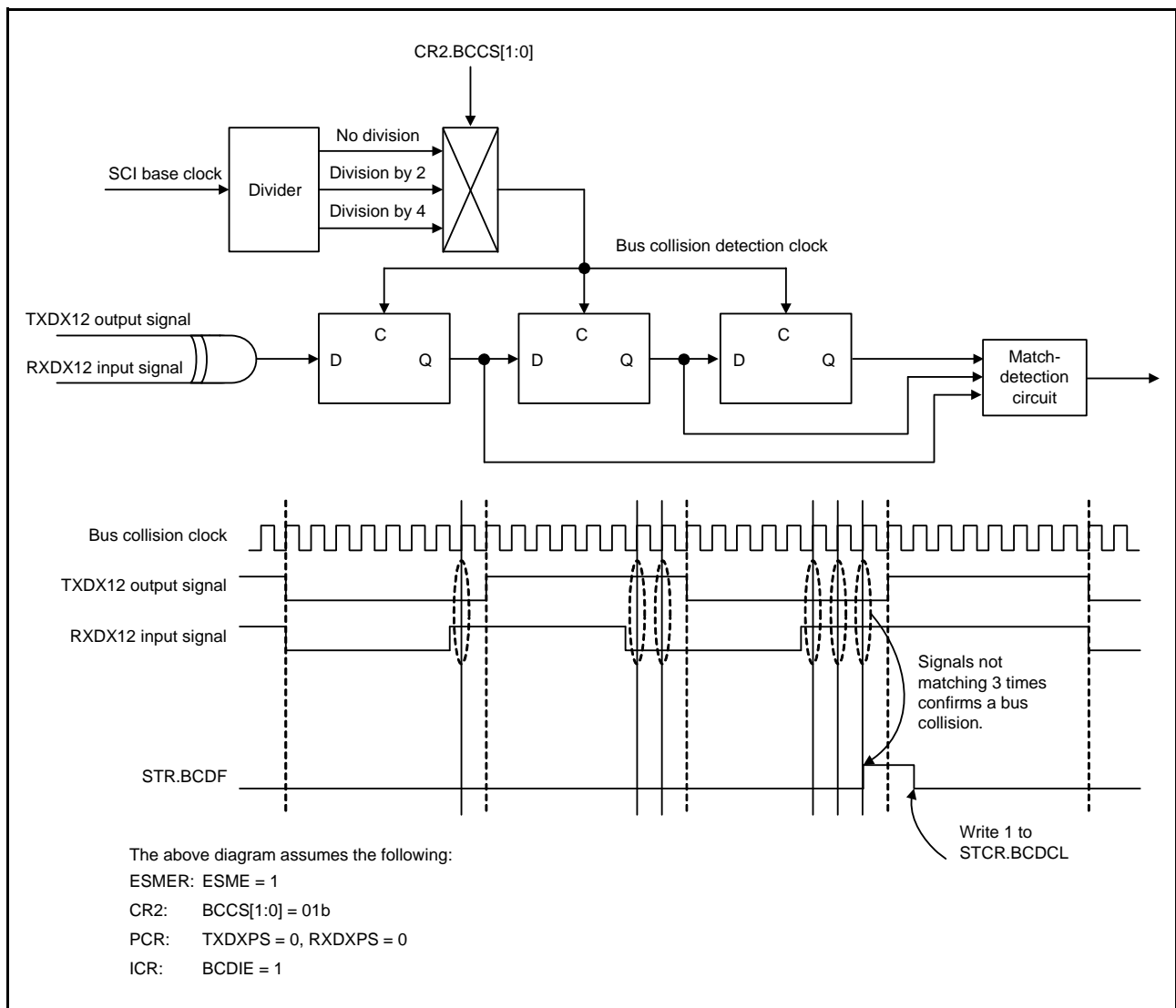


Figure 27.67 Example of Operations with Bus Collision Detection

### 27.9.5 Digital Filter for Input on the RXDX12 Pin

Signals input through the RXDX12 pin can be passed through a digital filter before they are conveyed to the internal circuits. The digital filter consists of three flip-flop circuit stages connected in series and a match-detecting circuit. The 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 27.68 shows an example of operations with the digital filter.

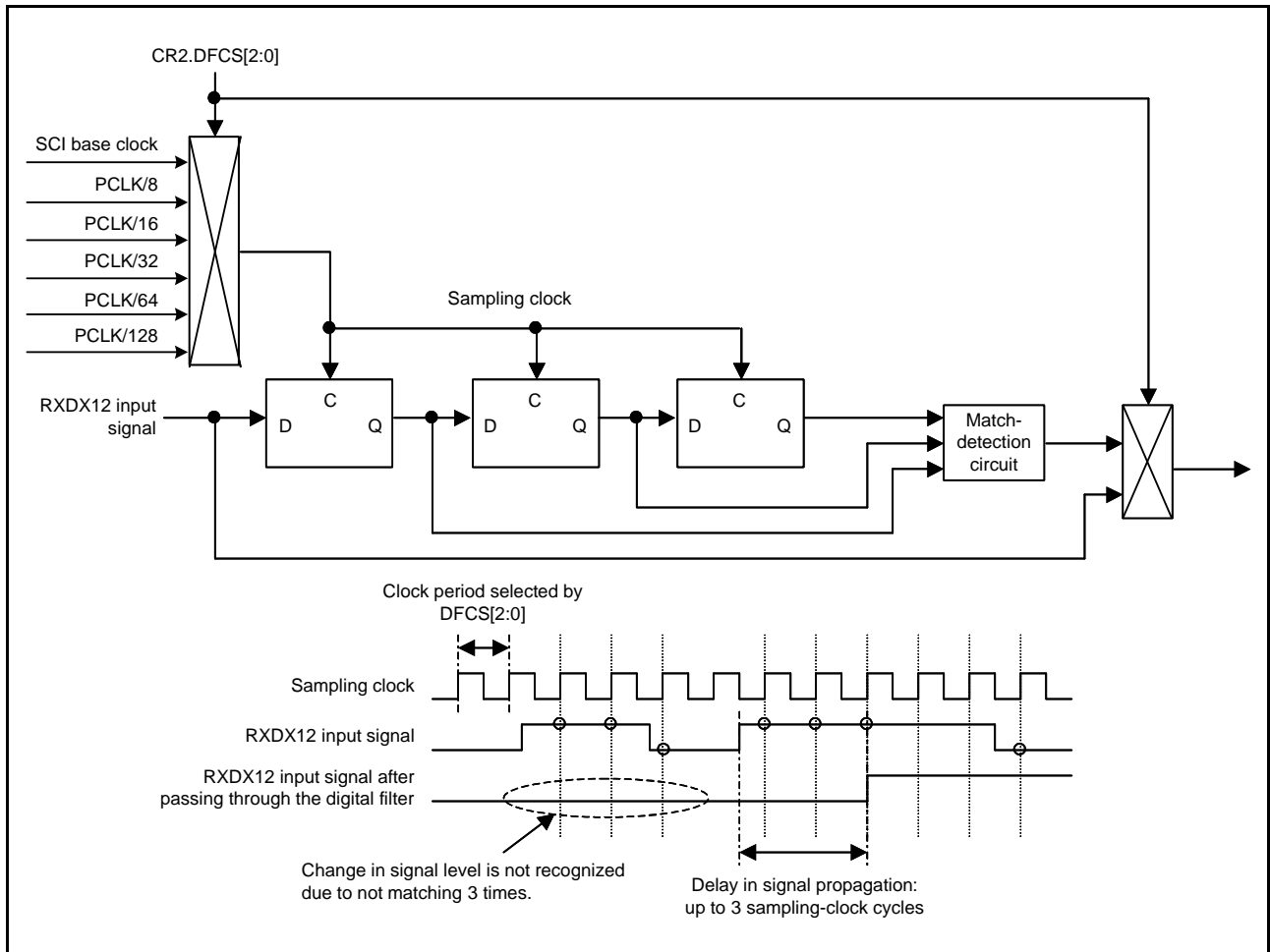


Figure 27.68 Example of Operations with the Digital Filter

### 27.9.6 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 27.69 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. To disable the bit rate measurement after a match with Control Field 1, write 0 to the BRME bit in CR0.

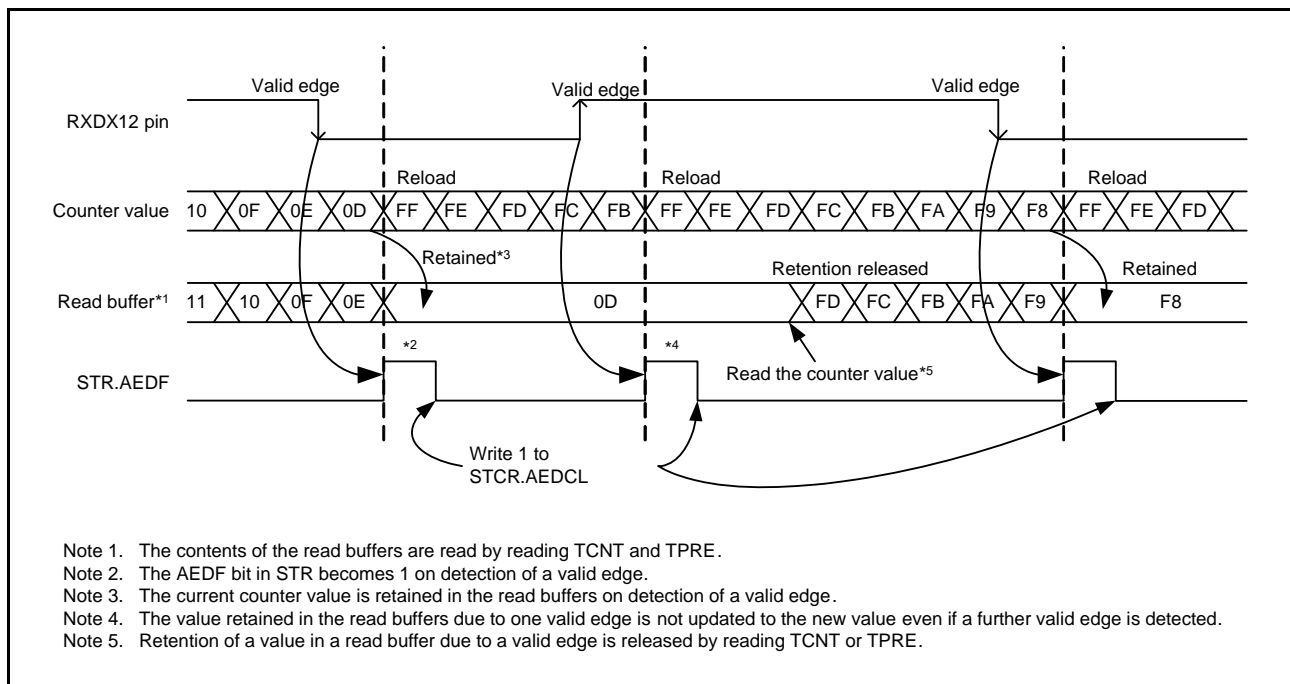
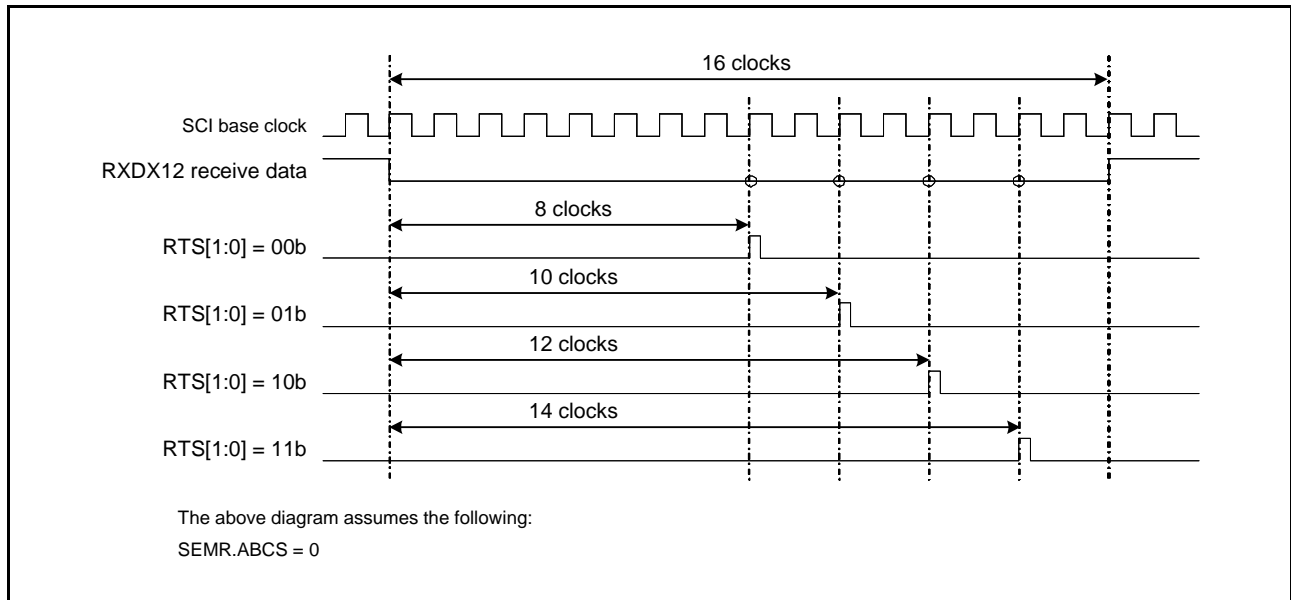


Figure 27.69 Example of Operations for Bit Rate Measurement

### 27.9.7 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 by setting the CRS.RTS[1:0] bits to select the rising edges of 8, 10, 12, or 14 cycles of the 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 PCLK clock of the SCI12. Figure 27.70 shows timing for the sampling of data received through RXDX12.



**Figure 27.70** Timing for Sampling of Data Received through RXDX12



### 27.9.8 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 transmission of a Start Frame. Setting the TMR.TOMS[1:0] bits to 10b 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 BDFD 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 TPRES and TCNT. After output of the Break Field low width is completed, stop the timer before it underflows again. Figure 27.71 shows an example of operations in Break Field low width output mode.

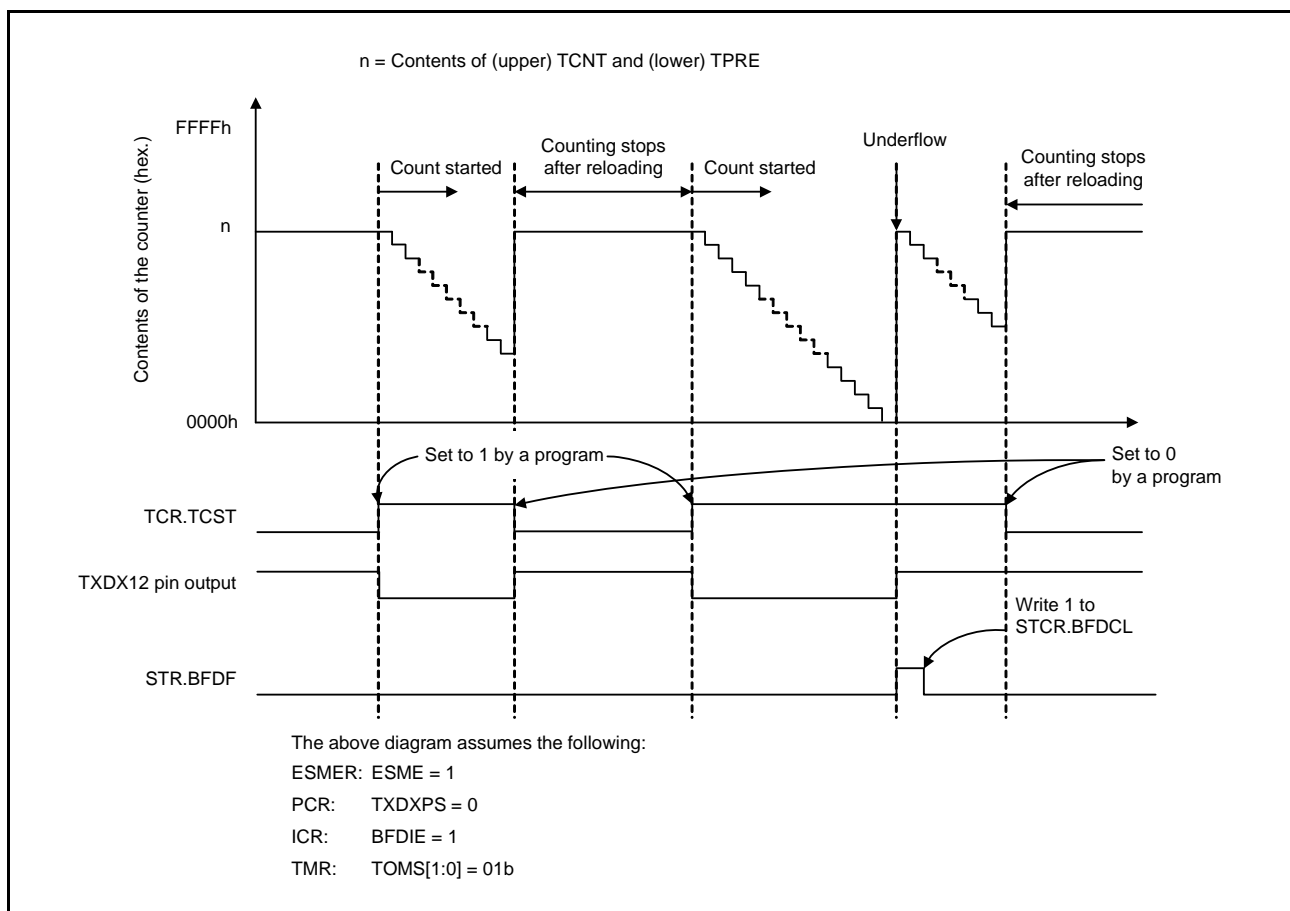


Figure 27.71 Example of Operations in Break Field Low Width Output Mode

## (2) Break Field Low Width Determination Mode

This mode is for determining the Break Field low width in the input signal on the RXDX12 pin at the reception of a Start Frame. Setting 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 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. 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 27.72 shows an example of operations in Break Field low width output mode.

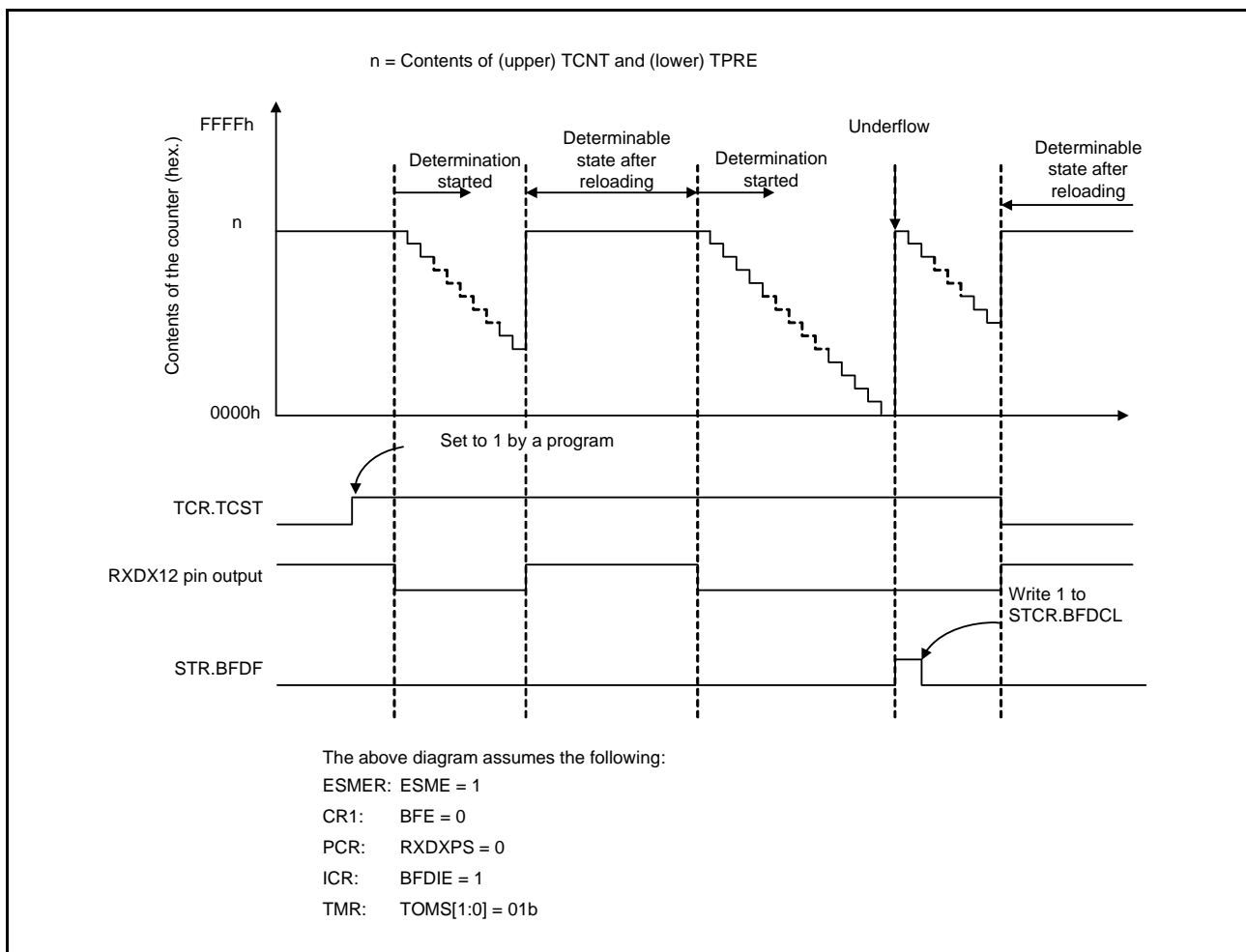


Figure 27.72 Example of Operations in Break Field Low Width Determination Mode

## (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 the counter. Counting starts when 1 is written to the TCST bit in TCR and stops when 0 is written to TCST. TPRES and TCNT both count down. TPRES counts cycles of the clock source for counting, and underflows of TPRES provide the clock source for counting by TCNT. When the timer underflows, 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.

### 27.10 Noise Cancellation Function

Figure 27.73 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.

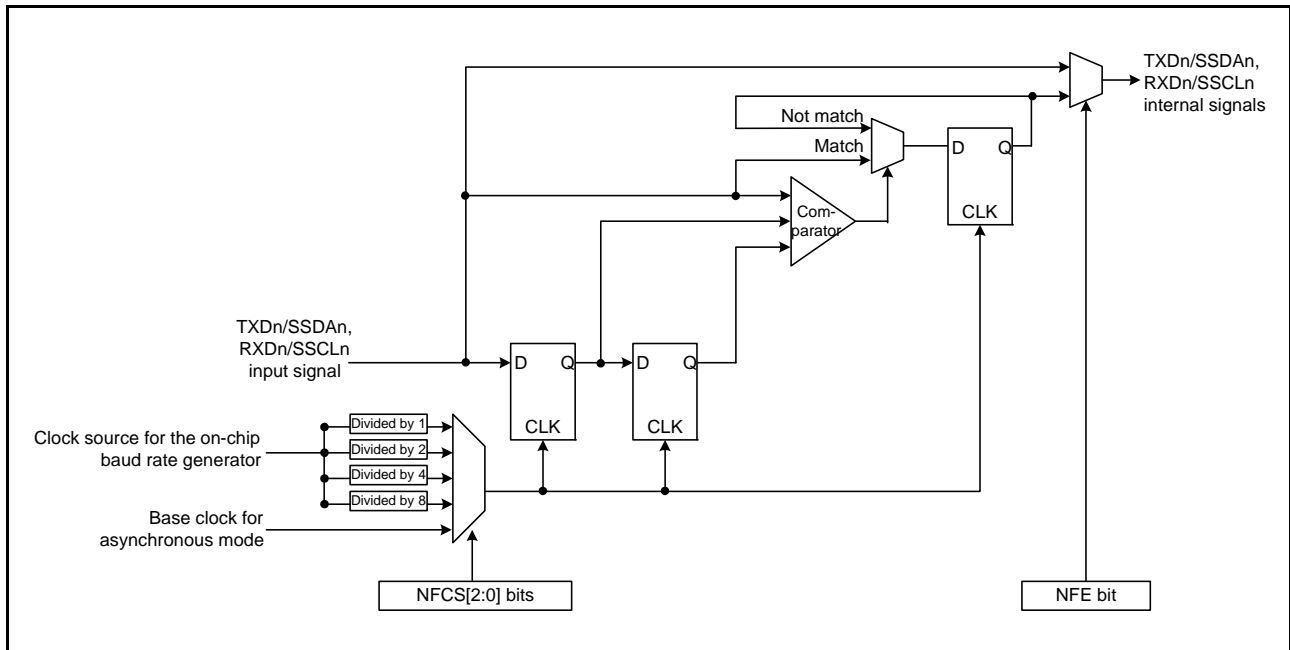


Figure 27.73 Block Diagram of Digital Noise Filter Circuit

## 27.11 Interrupt Sources

### 27.11.1 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.

### 27.11.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode

Table 27.30 lists interrupt sources in asynchronous mode, clock synchronous mode, and simple SPI mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled with the enable bits in SCR.

If the SCR.TIE bit is 1, a TXI interrupt request is generated when transmit data is 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 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.\*<sup>1</sup>

When new data is not written by the time of transmission of the last bit of the current transmit data and the setting of the SCR.TEIE bit is 1, the SSR.TEND flag becomes 1 and a TEI interrupt request is generated. Furthermore, when the setting of the SCR.TE bit is 1, the SSR.TEND flag retains the value 1 until further transmit data are written to the TDR, 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 is stored in the RDR. An RXI interrupt request can activate the DTC to handle data transfer.

Setting of any from among the ORER, FER, and PER flags in the SSR 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 27.30 Interrupt Sources**

Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
ERI	Receive error	ORER, FER, or PER	Not possible	High
RXI	Receive data full	—	Possible	↑
TXI	Transmit data empty	—	Possible	
TEI	Transmit end	TEND	Not possible	Low

### 27.11.3 Interrupts in Smart Card Interface Mode

Table 27.31 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

**Table 27.31 SCI Interrupt Sources**

Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	High
RXI	Receive data full	—	Possible	↑
TXI	Transmit data empty	TEND	Possible	Low

Data transmission/reception using the DTC is also possible in smart card interface mode, similar to in the normal SCI mode. In transmission, when the TEND flag in SSR is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC activation. The TEND flag is automatically set to 0 when the DTC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DTC is not activated. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the 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, be sure to make settings to enable the DTC before making SCI settings. For DTC settings, refer to section 16, 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 allowing transfer of receive data if the RXI request is specified beforehand as a source of DTC activation. If an error occurs, the error flag is set. Therefore, the DTC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

### 27.11.4 Interrupts in Simple I<sup>2</sup>C Mode

The interrupt sources in simple I<sup>2</sup>C mode are listed in Table 27.32. The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DTC can also be used to handle transfer in simple 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 SSCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DTC beforehand, the RXI request will activate the DTC to handle transfer of the received data. Furthermore, a TXI request is generated on the falling edge of the SSCLn signal for the ninth bit (acknowledge bit). If the TXI has been set up as an activating request for the DTC beforehand, the TXI request will activate the DTC to handle transfer of the transmit data.

When the value of the 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 beforehand, the RXI request will activate the DTC to handle transfer of the received data.

Also, if the DTC is used for data transfer in reception or transmission, be sure to set up and enable the DTC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in SIMR3 are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

**Table 27.32 SCI Interrupt Sources**

Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
RXI	Reception, ACK detection	—	Possible	High ↑ Low
TXI	Transmission, NACK detection	—	Possible*1	
STI	Completion of generation of a start, restart, or stop condition	IICSTIF	Not possible	

Note 1. Activation of the DTC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts).

### 27.11.5 Interrupt Requests from the Extended Serial Mode Control Section

The extended serial mode control section has a total of six types of interrupt request for generating the SCIX0 interrupt (Break Field low width detected), SCIX1 interrupt (Control Field 0 match, Control Field 1 match, priority interrupt bit detected), SCIX2 interrupt (bus collision detected), and SCIX3 interrupt (valid edge detected). When any of the interrupt factors is generated, the corresponding status flag is set to 1. Details of all of the interrupt requests are listed in Table 27.33.

**Table 27.33 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

## 27.12 Event Linking

By employing interrupt request signals as event signals, SCIF is able to provide linked operation through the event link controller (ELC) for modules selected in advance.

Event signals can be output regardless of the values of the corresponding interrupt request enable bits.

- (1) Error (receive error, error signal detected) event output
  - Indicates abnormal termination due to a parity error during reception in asynchronous mode.
  - Indicates abnormal termination due to a framing error during reception in asynchronous mode.
  - Indicates abnormal termination due to an overrun error during reception.
  - Indicates detection of the error signal during transmission in smart card interface mode.
- (2) Receive data full event output
  - Indicates that received data have been set in the receive data register (RDR).
  - Indicates that ACK has been detected if the SIMR2.IICINTM bit is 0 in simple I<sup>2</sup>C mode.
  - Indicates that the 8th-bit SSCL5 falling edge has been detected if the SIMR2.IICINTM bit is 1 in simple I<sup>2</sup>C mode.
  - When the SIMR2.IICINTM bit is 1 during master transmission in simple I<sup>2</sup>C mode, set the event link controller (ELC) so that receive data full events are not used.
- (3) Transmit data empty event output
  - Indicates that the SCR.TE bit has been changed from 0 to 1.
  - Indicates that transmit data have been transferred from the transmit data register (TDR) to the transmit shift register (TSR).
  - Indicates that transmission has been completed in smart card interface mode.
  - Indicates that NACK has been detected if the SIMR2.IICINTM bit is 0 in simple I<sup>2</sup>C mode.
  - Indicates that the ninth-bit SSCL5 falling edge has been detected if the SIMR2.IICINTM bit is 1 in simple I<sup>2</sup>C mode.
- (4) Transmit end event output
  - Indicates the completion of transmission.
  - Indicates that the starting condition, resumption condition, or termination condition has been generated in simple I<sup>2</sup>C mode.



## 27.13 Usage Notes

### 27.13.1 Setting the Module Stop Function

Module stop control register B (MSTPCRB) is used to stop and start SCI operations. With the value after a reset, SCI operations are stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

### 27.13.2 Break Detection and Processing

When a framing error is detected, a break can be detected by reading the RXDn pin value directly. In a break, the input from the RXDn pin becomes all 0s, and so the FER flag in SSR is set to 1 (framing error has occurred), and the PER flag in SSR may also be set to 1 (parity error has occurred). The SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is set to 0 (no framing error occurred), it will be set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the SSR.FER flag is set to 0 at this time, the SSR.FER flag retains 0 during the break. When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit at the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

### 27.13.3 Mark State and Production of Breaks

When the SCR.TE bit is 0 (serial transmission is disabled), 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 (serial transmission is enabled), 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 communication line 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 transmitter is initialized regardless of the current state of transmission.

### 27.13.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

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 set the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be set to 0 even if the RE bit in SCR is set to 0 (serial reception is disabled).

### 27.13.5 Writing Data to TDR

Data can 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 handling routine.

### 27.13.6 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

Update TDR by the CPU or DTC and wait for at least five PCLK cycles before allowing the transmit clock to be input (see Figure 27.74).

(2) Continuous transmission

- (a) Write the next transmit data to TDR or TDRL before the falling edge of the transmit clock (bit 7) (see Figure 27.74).
- (b) When updating TDR after bit 7 has started to transmit, update TDR while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock (bit 7) to four PCLK cycles or longer (see Figure 27.74).

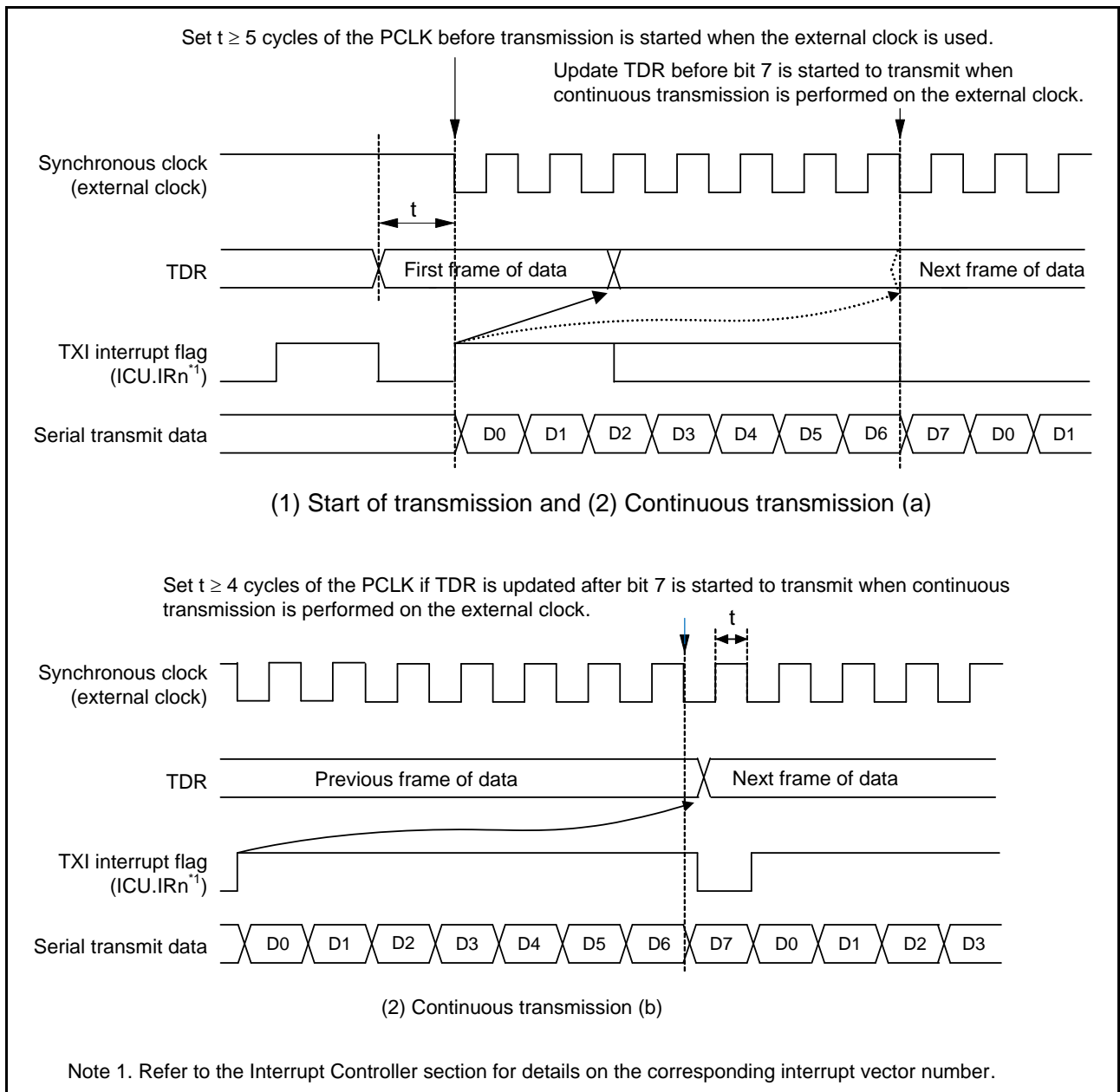


Figure 27.74 Restrictions on Use of External Clock in Clock Synchronous Transmission

### 27.13.7 Restrictions on Using DTC

When using the DTC to read RDR, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant SCI.

### 27.13.8 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR bit) in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1). For details on the interrupt status flag, refer to section 14, Interrupt Controller (ICUb).

- Confirm that transfer has stopped (the setting of the SCR.TE or SCR.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR.TIE or SCR.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has become 0.
- Set the interrupt status flag (IRn.IR bit) in the interrupt controller to 0.

### 27.13.9 SCI Operations during Low Power Consumption State

#### (1) Transmission

When making settings for the module stopped state or in transitions to software standby, stop operations (by setting the TIE, TE, and TEIE bits in the SCR to 0) after switching the TXDn pin to the general I/O port pin function. Setting the TE bit to 0 resets the TSR register and the TEND bit in the SSR. Depending on the port settings, output pins may output the level before a transition to the low power consumption state is made after release from the module stopped state or software standby mode. When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same 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 27.75 shows a sample flowchart for transition to software standby mode during transmission. Figure 27.76 and Figure 27.77 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 27.78 shows a sample flowchart for transition to software standby mode during reception.

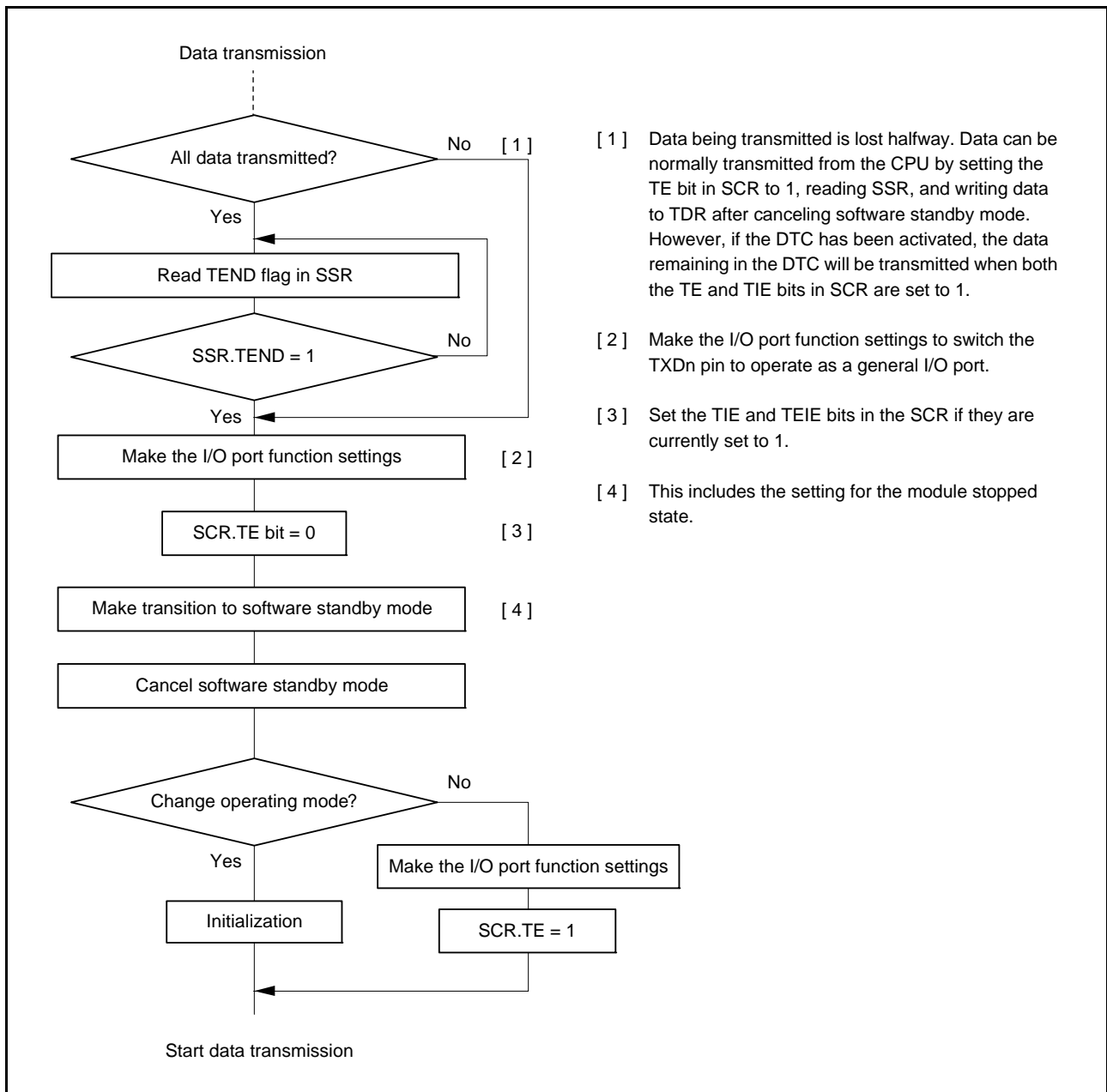
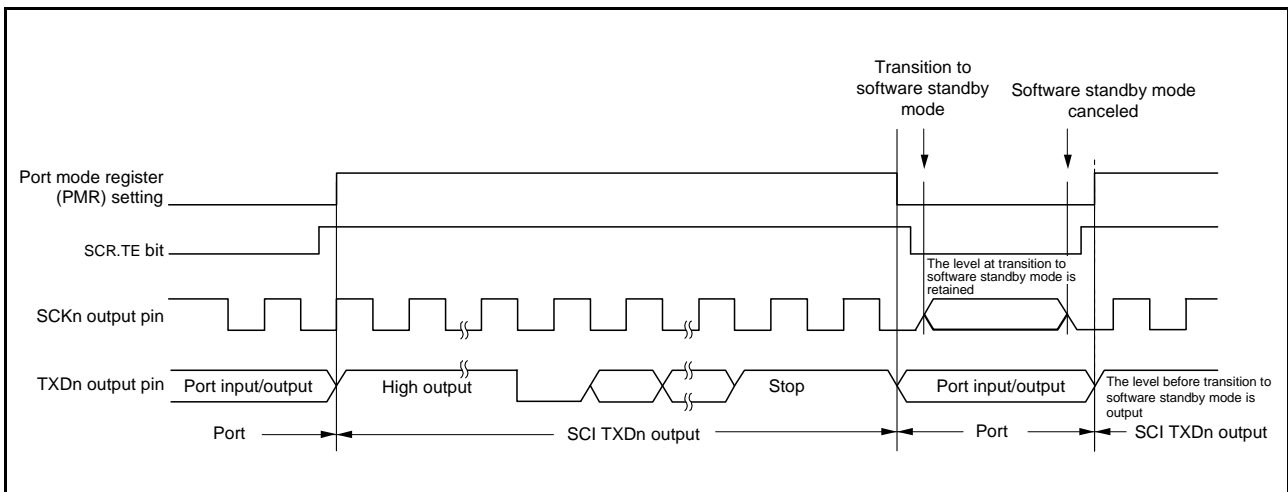
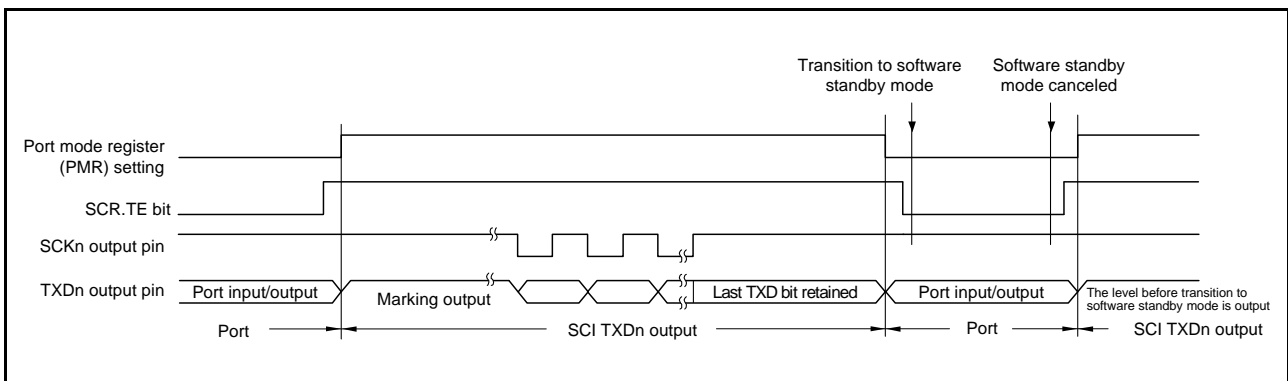


Figure 27.75 Example of Flowchart for Transition to Software Standby Mode during Transmission



**Figure 27.76 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)**



**Figure 27.77 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)**

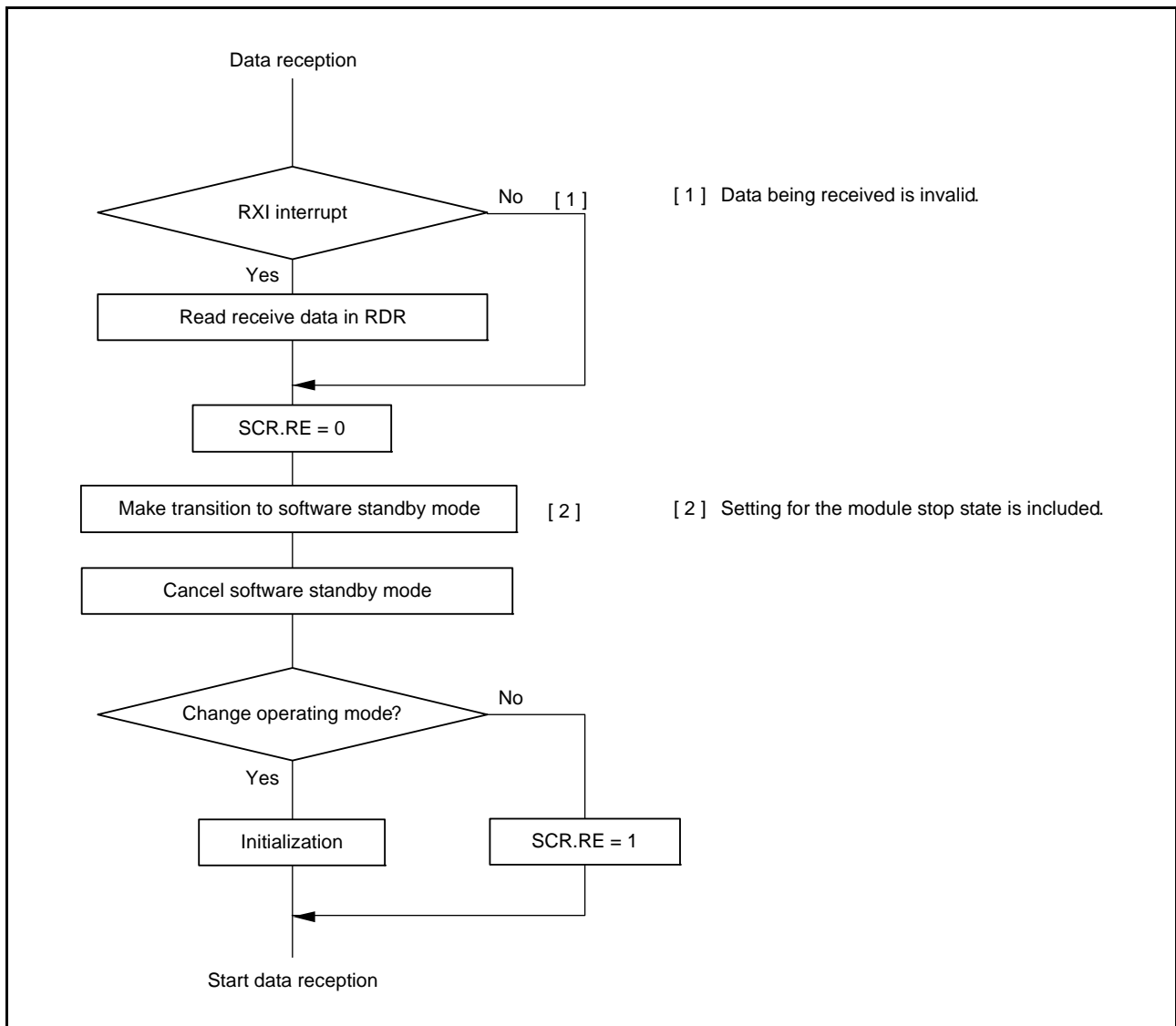


Figure 27.78 Example of Flowchart for Transition to Software Standby Mode during Reception

### 27.13.10 External Clock Input in Clock Synchronous Mode and Simple SPI Mode

In clock synchronous mode and simple SPI mode, the external clock SCKn must be input as follows:  
 High-pulse period, low-pulse period = 2 PCLK cycles or more, period = 6 PCLK cycles or more

### 27.13.11 Limitations on Simple SPI Mode

#### (1) Master Mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set by the SPMR.CKPH and CKPOL bits when the SPMR.SSE bit is 1.  
This prevents the clock line from being placed in the high-impedance state when the SCR.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR.TE bit is changed from 0 to 1. When the SPMR.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not necessary because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- In the case of the setting for clock delay (SPMR.CKPH bit is 1), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 27.79. 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.
- In a multi-master configuration, take care because the SCKn pin output becomes high-impedance while the input on the SSn# pin is at the low level if a mode fault error occurs as the current character is being transferred, stopping supply of the clock signal to the connected slave. Remake the settings for the connected slave to avoid misaligned bits when transfer is restarted.

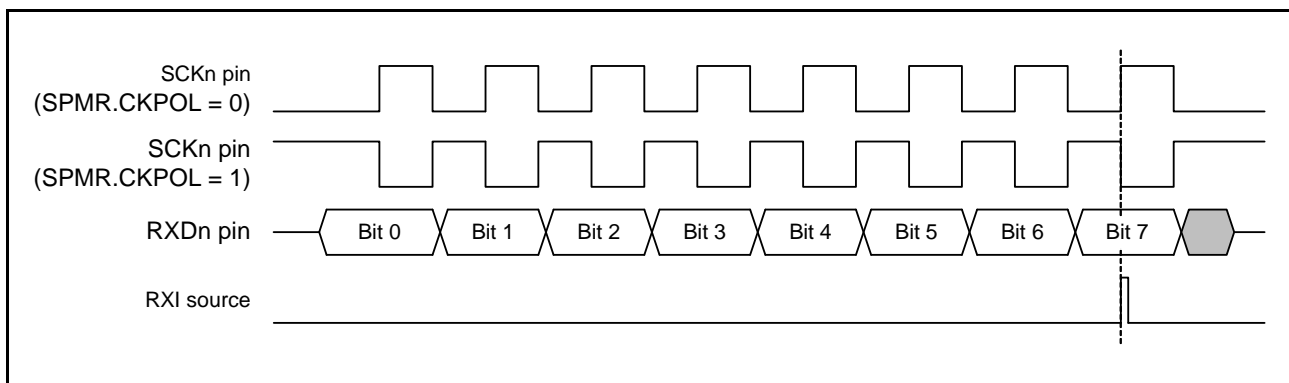


Figure 27.79 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

#### (2) Slave Mode

- 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.

### 27.13.12 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 SCIF 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 SCI12.SCR is 1.

### 27.13.13 Limitation 2 on Usage of the Extended Serial Mode Control Section

An SCIE interrupt request is generated even if the extended serial mode is enabled. However, the SCIE interrupt should not be used during reception of a Start Frame because SCIF uses an SCIE interrupt request.

The two ways of dealing with this are described below. When a receive error is detected, clear the error flag of the SCIE and initialize the control section of the SCIF.

- (1) Set the SCR.RIE bit of the SCIE to 0 to disable the output of interrupt requests. Check the error flags in the SSR register for SCIE on completion of the reception of a Start Frame, because an ERI interrupt is not generated if a receive error occurs. After reception of the Start Frame is completed, set the SCR.RIE bit of the SCIE to 1 by the time the first byte of the Information Frame is received.
- (2) Set the SCR.RIE bit of the SCIE 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.

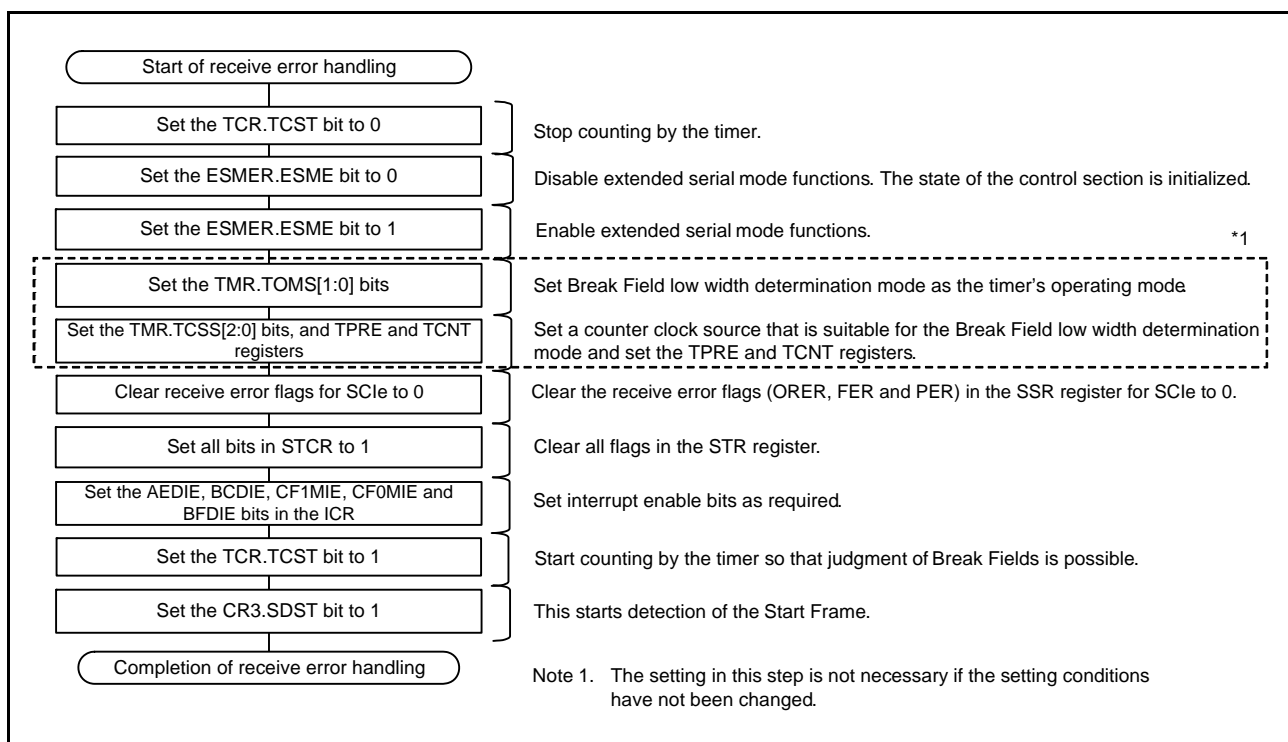


Figure 27.80 Example of Flowchart for Receive Error Handling (during Reception of the Start Frame)



#### 27.13.14 Note on Transmit Enable Bit (TE bit)

When setting the SCR.TE bit to 0 (serial transmission is disabled) while the pin function is “TXDn”, output of the pin becomes high impedance.

Prevent the TXDn line from becoming high impedance by any of the following ways:

- (1) Connect a pull-up resistor to the TXDn line.
- (2) Change the pin function to “general-purpose I/O port, output” before setting the SCR.TE bit to 0.  
Set the SCR.TE bit to 1 before changing the pin function to “TXDn”.

## 28. IrDA Interface

The IrDA interface sends and receives IrDA data communication waveforms in cooperation with the SCI5 based on the IrDA (Infrared Data Association) standard 1.0.

In this section, “PCLK” is used to refer to PCLKB.

### 28.1 Overview

Enabling the IrDA function by using the IRE bit in the IRCR register allows encoding and decoding the TXD5 and RXD5 signals of the SCI5 to the waveforms conforming to the IrDA standard 1.0 (IRTXD5 and IRRXD5 pins). Connecting these waveforms to an infrared transmitter/receiver implements infrared data communication conforming to the IrDA standard 1.0 system.

With the IrDA standard 1.0 system, data transfer can be started at 9600 bps and the transfer rate can be changed whenever necessary. Since the IrDA interface cannot change the transfer rate automatically, the transfer rate should be changed through software.

Figure 28.1 is a block diagram showing cooperation between the IrDA and SCI5.

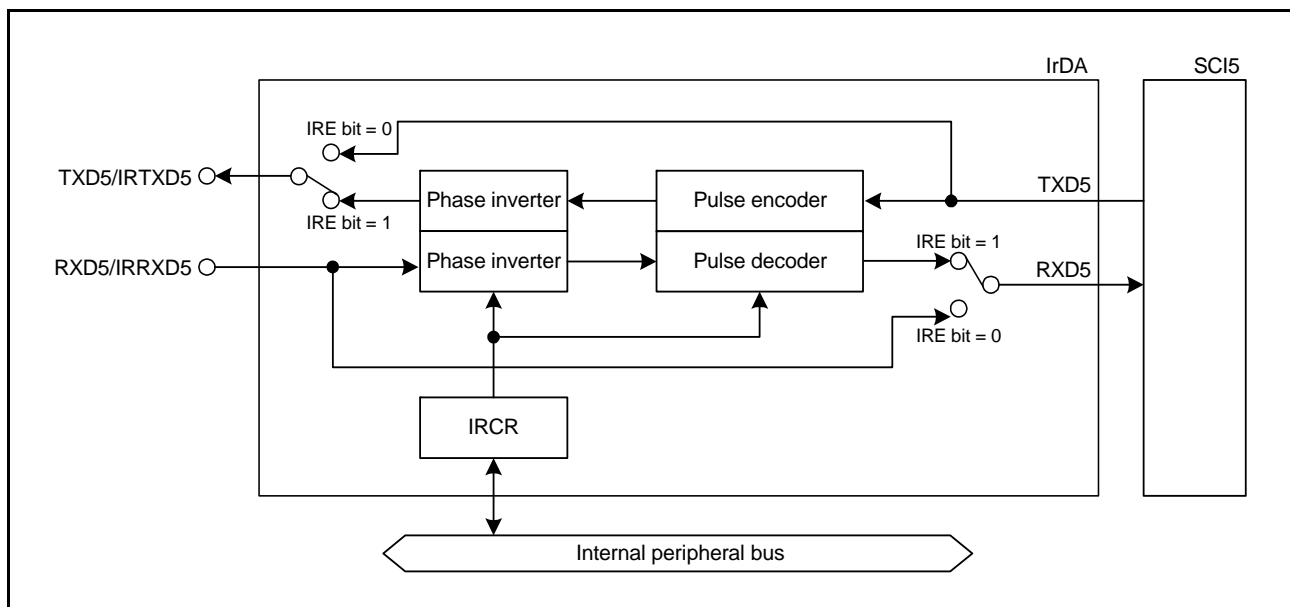


Figure 28.1 Block Diagram Showing Cooperation between IrDA and SCI5

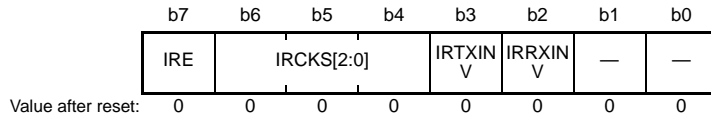
Table 28.1 IrDA Interface I/O Pins

Pin Name	I/O	Function
IRTXD5	Output	Outputs data to be transmitted.
IRRXD5	Input	Inputs received data.

## 28.2 Register Descriptions

### 28.2.1 IrDA Control Register (IRCR)

Address(es): 0008 8410h



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	IRRXINV	IrRX Data Polarity Switching	0: IRRXD5 input is used as received data as is. 1: IRRXD5 input is used as received data after the polarity is inverted.	R/W
b3	IRTXINV	IrTX Data Polarity Switching	0: Data to be transmitted is output to IRTXD5 as is. 1: Data to be transmitted is output to IRTXD5 after the polarity is inverted.	R/W
b6 to b4	IRCKS[2:0]	IrDA Output Pulse Width Select	<div style="display: flex; justify-content: space-between;"> <span>b6</span> <span>b4</span> </div> 0 0 0: B × 3/16 (B = bit period) 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	IRE	IrDA Enable	0: Serial I/O pins are used for normal serial communication. 1: Serial I/O pins are used for IrDA data communication.	R/W

Note 1. The IRCR register values are retained in module stop state, the sleep mode, deep sleep mode, and software standby mode.

#### IRRXINV Bit (IrRX Data Polarity Switching)

This bit inverts the logic level of the IrRXD input.

When IRRXINV = 0, the IrRXD input is used as received data as is.

When IRRXINV = 1, the IrRXD input is used after the polarity is inverted.

#### IRTXINV Bit (IrTX Data Polarity Switching)

This bit inverts the logic level of the IrTXD output.

When IRTXINV = 0, data is transmitted to IrTXD as is, and the high-level pulse width specified by IRCKS is applied.

When IRTXINV = 1, data is transmitted to IrTXD after the polarity is inverted, and the low-level pulse width specified by IRCKS is applied.

#### IRCKS[2:0] Bits (IrDA Output Pulse Width Select)

These bits set the high-level pulse width during IRTXD5 output encoding when the IrDA function is enabled.

Use the following procedure to set the IRCKS[2:0] bits.

- (1) Set the IRCR register to specify the IrDA function (IRE bit = 1 (IrDA is enabled)).
- (2) Set the IRCKS[2:0] bits to 000b.
- (3) Set the SCI5.SCR.TE bit to 1 (transmission is enabled).
- (4) Wait for a duration of 18/(16 × SCI5 bit rate).
- (5) Set the IRCKS[2:0] bits to the target value.

#### IRE Bit (IrDA Enable)

This bit selects either normal serial communication or IrDA data communication as the function of the serial I/O pins.

### 28.3 Operation

#### 28.3.1 Transmission

In transmission, the signals output from the SCI5 (UART frames) are converted to the IR frame data through the IrDA interface (see Figure 28.2). When the IRCR.IRTXINV bit is 0 and data is 0, high-level pulses with 3/16 of the bit period are output (initial setting). The high-level pulse width can be changed by setting the IRCR.IRCKS[2:0] bits. The standard prescribes that the minimum high-level pulse width should be  $1.41 \mu\text{s}$  and the maximum high-level pulse width should be the bit period  $\times (3/16 + 2.5\%)$  or  $(\text{the bit period} \times 3/16) + 1.08 \mu\text{s}$ . When the peripheral module clock PCLK is 20 MHz, the minimum high-level pulse width can be set to  $1.6 \mu\text{s}$  (101b: PCLK/32) as shown in Table 28.2. When data is 1, no pulses are output.

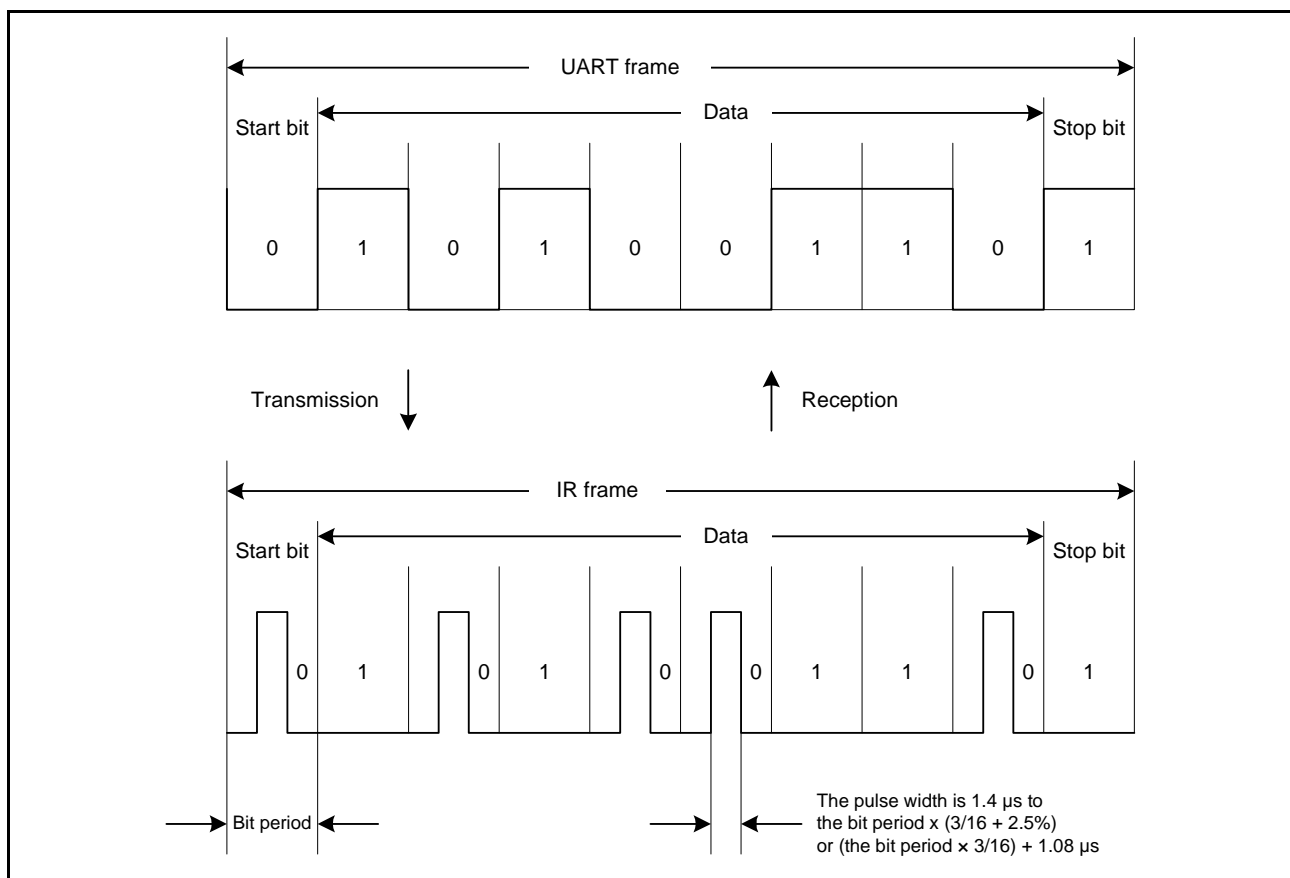


Figure 28.2 IrDA Transmission/Reception

### 28.3.2 Reception

In reception, the IR frame data is converted to the UART frame data through the IrDA interface and is input to the SCI5. Low-level data is input when the IRCR.IRRXINV bit is 0 and a high-level pulse is detected; high-level data is input when no pulse is detected for a 1-bit period.

### 28.3.3 Selecting High-Level Pulse Width

The IRCKS[2:0] bits need to be set to encode waveforms that meet the IrDA standard (the minimum pulse width should be  $1.41 \mu\text{s}$  and the maximum should be the bit period  $\times (3/16 + 2.5\%)$  or  $(\text{the bit period} \times 3/16) + 1.08 \mu\text{s}$ ).

The default value of the IRCKS[2:0] bits is 000b (the bit period  $\times 3/16$ ). Table 28.2 lists IRCKS[2:0] bit settings and the corresponding operating frequencies and bit rates of this module, which can be used when setting the pulse width shorter than the default value.

**Table 28.2 IRCKS[2:0] Bit Settings (When setting the pulse width shorter than IRCKS[2:0] = 000b (the bit period  $\times 3/16$ ))**

Peripheral Module Operating Frequency PCLK (MHz)	Bit Rate (bps) (Upper Row)/Bit Period $\times 3/16$ ( $\mu\text{s}$ ) (Lower Row)*1					
	2400	9600	19200	38400	57600	115200
	78.13	19.53	9.77	4.88	3.26	1.63
4.9152	011b-111b	011b-110b	011b-101b	011b-100b	011b-100b	011b
5	011b-111b	011b-110b	011b-101b	011b-100b	011b-100b	011b
6	100b-111b	100b-110b	100b-101b	100b	100b	—*2
6.144	100b-111b	100b-110b	100b-101b	100b	100b	—*2
7.3728	100b-111b	100b-110b	100b-101b	100b-101b	100b	—*2
8	100b-111b	100b-111b	100b-110b	100b-101b	100b	—*2
9.8304	100b-111b	100b-111b	100b-110b	100b-101b	100b-101b	100b
10	100b-111b	100b-111b	100b-110b	100b-101b	100b-101b	100b
12	101b-111b	101b-111b	101b-110b	101b	101b	—*2
12.288	101b-111b	101b-111b	101b-110b	101b	101b	—*2
14	101b-111b	101b-111b	101b-111b	101b-110b	101b	—*2
14.7456	101b-111b	101b-111b	101b-111b	101b-110b	101b	—*2
16	101b-111b	101b-111b	101b-111b	101b-110b	101b	—*2
16.9344	101b-111b	101b-111b	101b-111b	101b-110b	101b	—*2
17.2032	101b-111b	101b-111b	101b-111b	101b-110b	101b	—*2
18	101b-111b	101b-111b	101b-111b	101b-110b	101b	—*2
19.6608	101b-111b	101b-111b	101b-111b	101b-110b	101b-110b	101b
20	101b-111b	101b-111b	101b-111b	101b-110b	101b-110b	101b
25	110b-111b	110b-111b	110b-111b	110b	110b	—*2

Note 1. Bit rates cannot be set for the IrDA interface. For details on bit rate settings, section 27, Serial Communications Interface (SCI), SCI5.

Note 2. A pulse width which is less than the bit period  $\times 3/16$  cannot be set. Set the IRCKS[2:0] bits to 000b.

## 28.4 Usage Notes

### 28.4.1 Module Stop Function Setting

The IrDA can be enabled and disabled using module stop control register C (MSTPCRC). The IrDA is stopped after a reset. Registers can be accessed by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

### 28.4.2 SCI5 Setting

When using the IrDA, set the SCI5.SEMR.ABCS bit to 0 and SMR.STOP bit to 1 (2 stop bits).

### 28.4.3 Minimum Pulse-Width during Reception

For the input signals (IRRXD5) of this IrDA, input waveforms compliant to IrDA standard 1.0 (minimum pulse width: 1.4 us).

### 28.4.4 Notes on IrDA Initial Setting/Resetting

To change the value of the SCI5.SCR.TE or RE bit, set the IRCR.IRE bit to 1 (IrDA operates) and the IRCR.IRCKS[2:0] bits to 000b and then use the following procedure.

(1) Enabling transmission

- After the SCI.SCR.TE bit is set to 1 (transmission is enabled), wait for a duration of  $18/(16 \times \text{SCI5 bit rate})$  and then switch the I/O port function to the IRTXD5 pin.

(2) Enabling reception

- After the I/O port function is switched to the IRTXD5 pin, wait for a duration of  $18/(16 \times \text{SCI5 bit rate})$  and then set the SCI.SCR.TE bit to 1 (transmission is enabled).

When performing transmission and reception, a duration of  $18/(16 \times \text{SCI5 bit rate})$  can be waited at the same time.

## 29. I<sup>2</sup>C Bus Interface (RIIC)

This MCU has a single-channel I<sup>2</sup>C bus interface (RIIC).

The RIIC module conforms with the NXP I<sup>2</sup>C bus (Inter-IC-Bus) interface and provides a subset of its functions.

In this section, “PCLK” is used to refer to PCLKB.

### 29.1 Overview

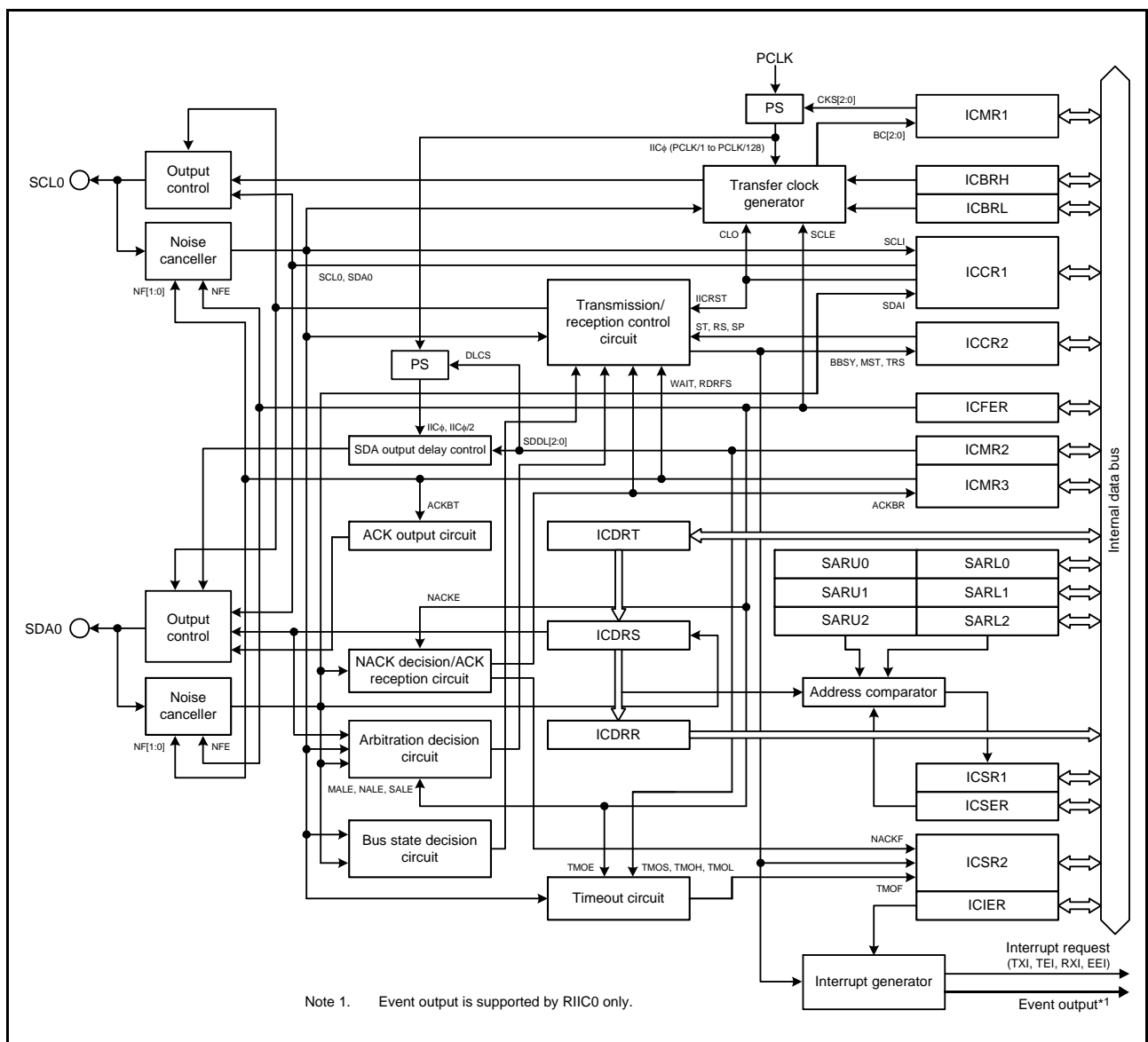
Table 29.1 lists the specifications of the RIIC, Figure 29.1 shows a block diagram of the RIIC, and Figure 29.2 shows an example of I/O pin connections to external circuits (I<sup>2</sup>C bus configuration example). Table 29.2 lists the I/O pins of the RIIC.

**Table 29.1 RIIC Specifications (1/2)**

Item	Description
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 setup times, hold times, and bus-free times for the transfer rate</li> </ul>
Transfer rate	Fast-mode is supported (up to 400 kbps)
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4 to 96%.
Issuing and detecting conditions	Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.
Slave address	<ul style="list-style-type: none"> <li>Up to three different slave addresses can be set.</li> <li>7-bit and 10-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 SCL clock 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</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 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 timeout function is capable of detecting long-interval stop of the SCL clock.
Noise cancellation	The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software.
Interrupt sources	<ul style="list-style-type: none"> <li>Four sources:               <ul style="list-style-type: none"> <li>Error in transfer or occurrence of events</li> <li>Detection of arbitration, NACK, timeout, a start condition including a restart condition, or a stop condition</li> </ul> </li> <li>Receive data full (including matching with a slave address)</li> <li>Transmit data empty (including matching with a slave address)</li> <li>Transmit end</li> </ul>

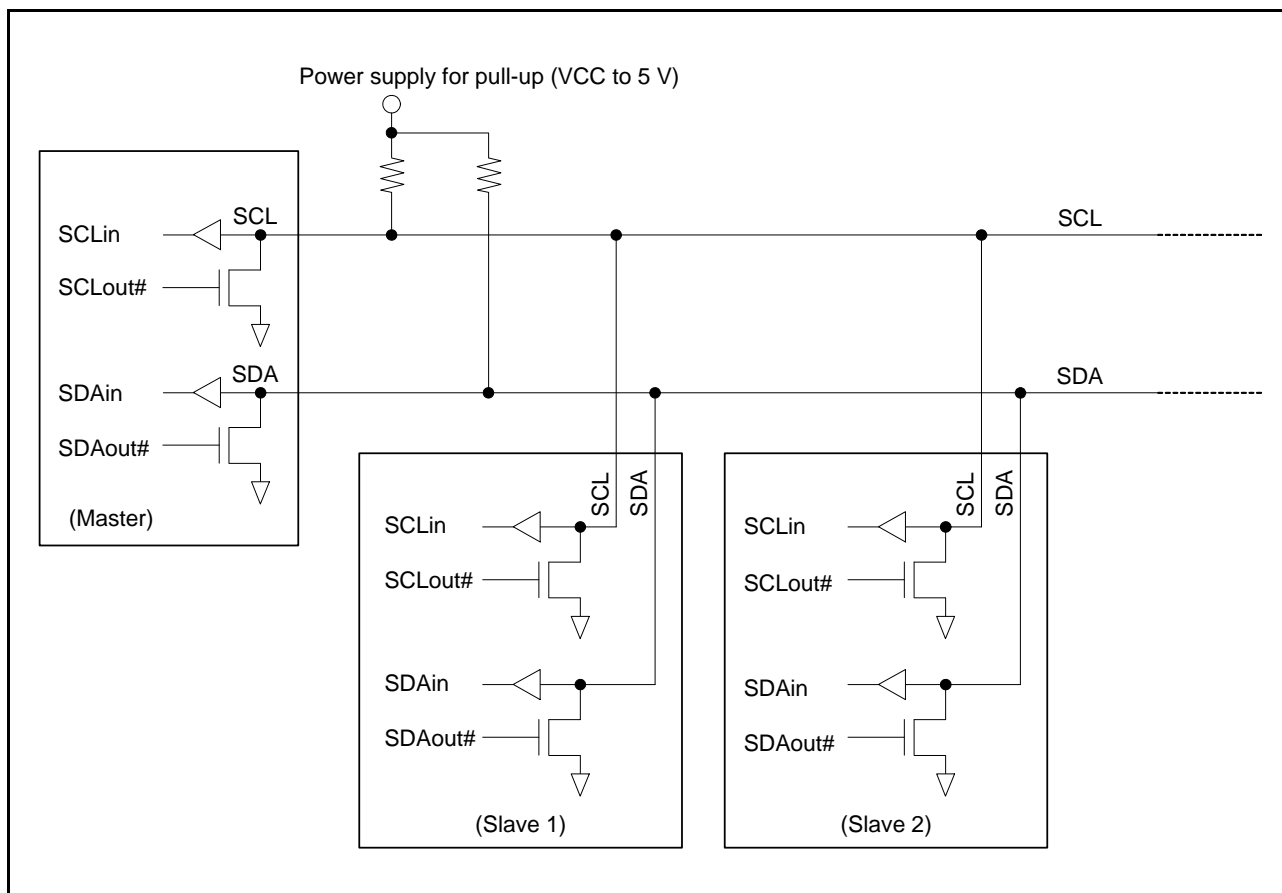
**Table 29.1 RIIC Specifications (2/2)**

Item	Description
Low power consumption function	Module stop state can be set.
RIIC operating modes	<ul style="list-style-type: none"> <li>• Four</li> <li>Master transmit mode, master receive mode, slave transmit mode, and slave receive mode</li> </ul>
Event link function (output)	<ul style="list-style-type: none"> <li>• Four sources (RIIC0):</li> <li>Error in transfer or occurrence of events</li> <li>Detection of arbitration, NACK, timeout, 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>• Transmit end</li> </ul>



**Figure 29.1 RIIC Block Diagram**





**Figure 29.2 I/O Pin Connection to the External Circuit (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 (ICMR3.SMBS bit = 0), or TTL when SMBus is selected (ICMR3.SMBS bit = 1).

**Table 29.2 RIIC Pin Configuration**

Channel	Pin Name	I/O	Function
RIIC0	SCL0	I/O	RIIC0 serial clock I/O pin
	SDA0	I/O	RIIC0 serial data I/O pin

## 29.2 Register Descriptions

### 29.2.1 I<sup>2</sup>C Bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h

b7	b6	b5	b4	b3	b2	b1	b0
ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI

Value after reset: 0 0 0 1 1 1 1 1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAI	SDA Line Monitor	0: SDA0 line is low. 1: SDA0 line is high.	R
b1	SCLI	SCL Line Monitor	0: SCL0 line is low. 1: SCL0 line is high.	R
b2	SDAO	SDA Output Control/Monitor	<ul style="list-style-type: none"> <li>Read:               <ul style="list-style-type: none"> <li>0: The RIIC has driven the SDA0 pin low.</li> <li>1: The RIIC has released the SDA0 pin.</li> </ul> </li> <li>Write:               <ul style="list-style-type: none"> <li>0: The RIIC drives the SDA0 pin low.</li> <li>1: The RIIC releases the SDA0 pin.</li> </ul> </li> </ul>	R/W
b3	SCLO	SCL Output Control/Monitor	<ul style="list-style-type: none"> <li>Read:               <ul style="list-style-type: none"> <li>0: The RIIC has driven the SCL0 pin low.</li> <li>1: The RIIC has released the SCL0 pin.</li> </ul> </li> <li>Write:               <ul style="list-style-type: none"> <li>0: The RIIC drives the SCL0 pin low.</li> <li>1: The RIIC releases the SCL0 pin. (High level output is achieved through an external pull-up resistor.)</li> </ul> </li> </ul>	R/W
b4	SOWP	SCLO/SDAO Write Protect	0: SCLO and SDA0 bits can be written. 1: SCLO and SDA0 bits are protected. (This bit is read as 1.)	R/W
b5	CLO	Extra SCL Clock Cycle Output	0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL 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: Releases the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL0/SDAO output latch)	R/W
b7	ICE	I <sup>2</sup> C Bus Interface Enable	0: Disable (SCL0 and SDA0 pins in inactive state) 1: Enable (SCL0 and SDA0 pins in active state) (Combined with the IICRST bit to select either RIIC or internal reset.)	R/W

#### SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)

These bits are used to directly control the SDA0 and SCL0 signals output from the RIIC.

When writing to these bits, also write 0 to the SOWP bit.

The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a start condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

When reading these bits, the state of signals output from the RIIC can be read.

**CLO Bit (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, refer to section 29.11.2, 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 29.3 lists the resets of the RIIC.

The RIIC reset resets all registers and internal states of the RIIC, and the internal reset resets the bit counter (ICMR1.BC[2:0] bits), 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, refer to section 29.14, Resets and Register and Function States When Issuing Each Condition.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error.

If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the SCL0 pin and SDA0 pin at a high impedance.

**Note:** If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If an internal reset is necessary because the RIIC hangs up with the SCL0 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 29.3 RIIC Resets**

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers and internal states of the RIIC.
	1	Internal reset	Resets the ICMR1.BC[2:0] bits, the ICSR1, ICSR2, and ICDRS registers, and the internal states of the RIIC.

**ICE Bit (I<sup>2</sup>C Bus Interface Enable)**

This bit selects the active or inactive state of the SCL0 and SDA0 pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 29.3, RIIC Resets, for the types of resets.

Set the ICE bit to 1 when using the RIIC. The SCL0 and SDA0 pins are placed in the active state when the ICE bit is set to 1.

Set the ICE bit to 0 when the RIIC is not to be used. The SCL0 and SDA0 pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCL0 or SDA0 pin to the RIIC when setting up the multi-function pin controller (MPC). Note that the slave address comparison operation is carried out if the pins are assigned to the RIIC.

## 29.2.2 I<sup>2</sup>C Bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h

b7	b6	b5	b4	b3	b2	b1	b0
BBSY	MST	TRS	—	SP	RS	ST	—

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

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ST	Start Condition 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: Master 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).	R

Note 1. When the ICMR1.MTWP bit 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 state).

For details on the start condition issuance, refer to section 29.10, 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 (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: Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free state).

Note that arbitration may be lost due to a start condition issuance error if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy state).

**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 state) and the MST bit is set to 1 (master mode).

For details on the restart condition issuance, refer to section 29.10, 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 (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 1 (requests to issue a restart condition) is written to the RS bit in slave mode, the restart condition is not issued but the RS bit remains set to 1. If the operating mode changes to master mode with the bit not being cleared, note that 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 state) and the MST bit is set to 1 (master mode).

For details on the stop condition issuance, refer to section 29.10, 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 (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 state).

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 TRS bit is automatically changed to 1 for transmit mode or 0 for receive mode by issuing or detection of a start condition and setting of the R/W# bit. 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 a restart condition is issued normally according to the restart condition issuance request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit added to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in 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 1 for master mode or 0 for slave mode by issuing of a start condition and issuing or detection of a stop 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 Flag)**

The BBSY flag indicates whether the I<sup>2</sup>C bus is occupied (bus busy state) or released (bus free state).

This bit is set to 1 when the SDA0 line changes from high to low under the condition of SCL0 line = high, assuming that a start condition has been issued.

When the SDA0 line changes from low to high under the condition of SCL0 line = high, this bit is set 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]

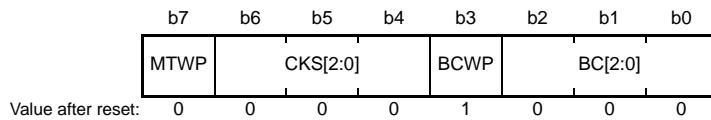
- When a start condition is detected

[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)

### 29.2.3 I<sup>2</sup>C Bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BC[2:0]	Bit Counter	b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
b3	BCWP	BC Write Protect	0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.)	R/W*1
b6 to b4	CKS[2:0]	Internal Reference Clock Select	Select the internal reference clock source (IIC $\phi$ ) for the RIIC. 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 ICCR2.MST and TRS bits. 1: Enables writing to the ICCR2.MST and TRS bits.	R/W

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

#### BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL0 line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

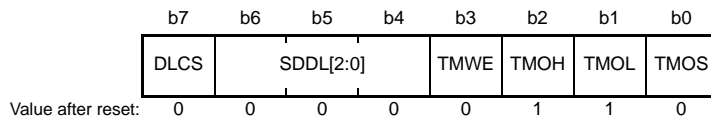
To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred frames when the SCL0 line is at a low level.

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.



### 29.2.4 I<sup>2</sup>C Bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h



Bit	Symbol	Bit Name	Description	R/W																																																						
b0	TMOS	Timeout Detection Time Select	0: Long mode is selected. 1: Short mode is selected.	R/W																																																						
b1	TMOL	Timeout L Count Control	0: Count-up is disabled while the SCL0 line is at a low level. 1: Count-up is enabled while the SCL0 line is at a low level.	R/W																																																						
b2	TMOH	Timeout H Count Control	0: Count-up is disabled while the SCL0 line is at a high level. 1: Count-up is enabled while the SCL0 line is at a high level.	R/W																																																						
b3	TMWE	Timeout Internal Counter Write Enable	0: Writing to internal counter of timeout detection function is disabled. 1: Writing to internal counter of timeout detection function is enabled.	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 style="margin-left: 20px; 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: No output delay</td></tr> <tr><td>0</td><td>0</td><td>1: 1 IIC<math>\phi</math> cycle</td></tr> <tr><td>0</td><td>1</td><td>0: 2 IIC<math>\phi</math> cycles</td></tr> <tr><td>0</td><td>1</td><td>1: 3 IIC<math>\phi</math> cycles</td></tr> <tr><td>1</td><td>0</td><td>0: 4 IIC<math>\phi</math> cycles</td></tr> <tr><td>1</td><td>0</td><td>1: 5 IIC<math>\phi</math> cycles</td></tr> <tr><td>1</td><td>1</td><td>0: 6 IIC<math>\phi</math> cycles</td></tr> <tr><td>1</td><td>1</td><td>1: 7 IIC<math>\phi</math> cycles</td></tr> </table> </li> <li>• When ICMR2.DLCS = 1 (IIC<math>\phi</math>/2)               <table style="margin-left: 20px; 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: No output delay</td></tr> <tr><td>0</td><td>0</td><td>1: 1 or 2 IIC<math>\phi</math> cycles</td></tr> <tr><td>0</td><td>1</td><td>0: 3 or 4 IIC<math>\phi</math> cycles</td></tr> <tr><td>0</td><td>1</td><td>1: 5 or 6 IIC<math>\phi</math> cycles</td></tr> <tr><td>1</td><td>0</td><td>0: 7 or 8 IIC<math>\phi</math> cycles</td></tr> <tr><td>1</td><td>0</td><td>1: 9 or 10 IIC<math>\phi</math> cycles</td></tr> <tr><td>1</td><td>1</td><td>0: 11 or 12 IIC<math>\phi</math> cycles</td></tr> <tr><td>1</td><td>1</td><td>1: 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 Select	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 Select)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE bit = 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit-counter. While the SCL0 line is in the state that enables this counter as specified by the TMOH and TMOL bits, the counter counts up in synchronization with the internal reference clock (IIC $\phi$ ) as a count source. For details on the timeout function, refer to section 29.11.1, Timeout Function.

#### TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL0 line is held low when the timeout function is enabled (ICFER.TMOE bit = 1).

**TMOH Bit (Timeout H Count Control)**

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL0 line is held high when the timeout function is enabled (ICFER.TMOE bit = 1).

**TMWE Bit (Timeout Internal Counter Write Enable)**

This bit is used to select whether or not to allocate the timeout internal counter (TMOCNL/TMOCNTU) to the address of the slave address register (SARL0/SARU0).

When this bit is set to 1, the addresses of the timeout internal counters (TMOCNL and TMOCNTU) are allocated to the addresses of SARL0 and SARU0.

**SDDL[2:0] Bits (SDA Output Delay Counter)**

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledge bit.

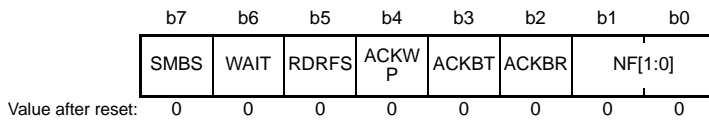
Set the SDA output delay time to meet the I<sup>2</sup>C bus standard (within the data enable time/acknowledge enable time\*<sup>1</sup>) or the SMBus standard (within the data hold time: 300 ns or more, and SCL-clock low-level period - the data setup time: 250 ns). Note that, if a value outside the 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.

For details on this function, refer to section 29.5, SDA Output Delay Function.

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])

### 29.2.5 I<sup>2</sup>C Bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Select	b1 b0 0 0: Noise of up to one 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.	R/W*1
b5	RDRFS	RDRF Flag Set Timing Select	0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle. (The SCL0 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 SCL0 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 Select	0: The I <sup>2</sup> C bus is selected. 1: The SMBus is selected.	R/W

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If it is attempted to write 1 to both the ACKWP and ACKBT bits at the same time, the ACKBT bit will not be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

#### NF[1:0] Bits (Noise Filter Stage Select)

These bits are used to select the number of stages in the digital noise filter.

For details on the digital noise filter function, refer to section 29.6, Digital Noise Filter Circuits.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCL0 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)

**ACKWP Bit (ACKBT Write Protect)**

This bit is used to control the modification of the ACKBT bit.

**RDRFS Bit (RDRF Flag Set Timing Select)**

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL0 line low at the falling edge of the eighth SCL clock cycle.

When the RDRFS bit is 0, the SCL0 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 SCL0 line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCL0 line is released by writing a value to the ACKBT bit.

After data is received with this setting, the SCL0 line is automatically held low before the 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 I<sup>2</sup>C bus receive data register (ICDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL 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 SCL0 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 Select)**

Setting this bit to 1 selects the SMBus and enables the HOAE bit in IC SER.

## 29.2.6 I<sup>2</sup>C Bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOE	Timeout Function Enable	0: The timeout function is disabled. 1: The timeout function is enabled.	R/W
b1	MALE	Master Arbitration-Lost Detection Enable	0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the ICCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the ICCR2.MST and TRS bits automatically when arbitration is lost.)	R/W
b2	NALE	NACK Transmission Arbitration-Lost Detection Enable	0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.	R/W
b3	SALE	Slave Arbitration-Lost Detection Enable	0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.	R/W
b4	NACKE	NACK Reception Transfer Suspension Enable	0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).	R/W
b5	NFE	Digital Noise Filter 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	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

### TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, refer to section 29.11.1, 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, refer to section 29.8.2, 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 set to 0 (no SCL synchronous circuit 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 SCL0 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 set to 0 except for checking the output of the set transfer rate.

### 29.2.7 I<sup>2</sup>C Bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h

b7	b6	b5	b4	b3	b2	b1	b0
HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E

Value after reset: 0 0 0 0 1 0 0 1

Bit	Symbol	Bit Name	Description	R/W
b0	SAR0E	Slave Address Register 0 Enable	0: Slave address in 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 slave address set in SARLy and SARUy.

When this bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address.

When this bit is set to 0, the slave address set in SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

#### GCAE Bit (General Call Address Enable)

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 set to 0, the received slave address is ignored even if it matches the general call address.

#### DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first 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 set 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, refer to section 29.7.3, 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.

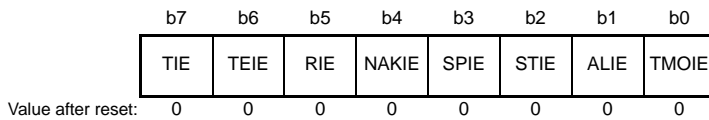
When this bit is set to 1 while the SMBS bit in ICMR3 is 1, if the received slave address matches the host address, the RIIC recognizes the received slave address as the host address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the receive operation.

When the SMBS bit in ICMR3 or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.



## 29.2.8 I<sup>2</sup>C Bus Interrupt Enable Register (ICIER)

Address(es): RIIC0.ICIER 0008 8307h



Bit	Symbol	Bit Name	Description	R/W
b0	TMOIE	Timeout Interrupt Request Enable	0: Timeout interrupt (TMOI) request is disabled. 1: Timeout interrupt (TMOI) request is enabled.	R/W
b1	ALIE	Arbitration-Lost Interrupt Request Enable	0: Arbitration-lost interrupt (ALI) request is disabled. 1: Arbitration-lost interrupt (ALI) request is enabled.	R/W
b2	STIE	Start Condition Detection Interrupt Request Enable	0: Start condition detection interrupt (STI) request is disabled. 1: Start condition detection interrupt (STI) request is enabled.	R/W
b3	SPIE	Stop Condition Detection Interrupt Request Enable	0: Stop condition detection interrupt (SPI) request is disabled. 1: Stop condition detection interrupt (SPI) request is enabled.	R/W
b4	NAKIE	NACK Reception Interrupt Request Enable	0: NACK reception interrupt (NAKI) request is disabled. 1: NACK reception interrupt (NAKI) request is enabled.	R/W
b5	RIE	Receive Data Full Interrupt Request Enable	0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.	R/W
b6	TEIE	Transmit End Interrupt Request Enable	0: Transmit end interrupt (TEI) request is disabled. 1: Transmit end interrupt (TEI) request is enabled.	R/W
b7	TIE	Transmit Data Empty Interrupt Request Enable	0: Transmit data empty interrupt (TXI) request is disabled. 1: Transmit data empty interrupt (TXI) request is enabled.	R/W

### TMOIE Bit (Timeout Interrupt Request Enable)

This bit is used to enable or disable timeout interrupt (TMOI) requests when the TMOF flag in ICSR2 is set to 1. A TMOI interrupt request is canceled by setting the TMOF flag or the TMOIE bit to 0.

### ALIE Bit (Arbitration-Lost Interrupt Request Enable)

This bit is used to enable or disable arbitration-lost interrupt (ALI) requests when the AL flag in ICSR2 is set to 1. An ALI interrupt request is canceled by setting the AL flag or the ALIE bit to 0.

### STIE Bit (Start Condition Detection Interrupt Request Enable)

This bit is used to enable or disable start condition detection interrupt (STI) requests when the START flag in ICSR2 is set to 1. An STI interrupt request is canceled by setting the START flag or the STIE bit to 0.

### SPIE Bit (Stop Condition Detection Interrupt Request Enable)

This bit is used to enable or disable stop condition detection interrupt (SPI) requests when the STOP flag in ICSR2 is set to 1. An SPI interrupt request is canceled by setting the STOP flag or the SPIE bit to 0.

### NAKIE Bit (NACK Reception Interrupt Request Enable)

This bit is used to enable or disable NACK reception interrupt (NAKI) requests when the NACKF flag in ICSR2 is set to 1. An NAKI interrupt request is canceled by setting the NACKF flag or the NAKIE bit to 0.

### RIE Bit (Receive Data Full Interrupt Request Enable)

This bit is used to enable or disable receive data full interrupt (RXI) requests when the RDRF flag in ICSR2 is set to 1.

**TEIE Bit (Transmit End Interrupt Request Enable)**

This bit is used to enable or disable transmit end interrupt (TEI) requests when the TEND flag in ICSR2 is set to 1. An TEI interrupt request is canceled by setting the TEND flag or the TEIE bit to 0.

**TIE Bit (Transmit Data Empty Interrupt Request Enable)**

This bit is used to enable or disable transmit data empty interrupt (TXI) requests when the TDRE flag in ICSR2 is set to 1.

### 29.2.9 I<sup>2</sup>C Bus Status Register 1 (ICSR1)

Address(es): RIIC0.ICSR1 0008 8308h

b7	b6	b5	b4	b3	b2	b1	b0
HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 is not detected. 1: Slave address 0 is detected.	R/(W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected.	R/(W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected.	R/(W) *1
b3	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected.	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DID	Device-ID Address Detection Flag	0: Device-ID command is not detected. 1: Device-ID command is detected. • This bit is set to 1 when the first 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.

#### AAS<sub>y</sub> Flag (Slave Address y Detection Flag) (y = 0 to 2)

[Setting conditions]

For 7-bit address format: SARU<sub>y</sub>.FS = 0

- When the received slave address matches the SVA[6:0] value in SARL<sub>y</sub> with the SAR<sub>y</sub>E bit in ICSE<sub>R</sub> 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: SARU<sub>y</sub>.FS = 1

- When the received slave address matches a value of (11110b + SVA[1:0] in SARU<sub>y</sub>) and the following address matches the SARL<sub>y</sub> value with the SAR<sub>y</sub>E bit in ICSE<sub>R</sub> 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 AAS<sub>y</sub> bit after reading AAS<sub>y</sub> = 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: SARU<sub>y</sub>.FS = 0

- When the received slave address does not match the SVA[6:0] value in SARL<sub>y</sub> with the SAR<sub>y</sub>E bit in ICSE<sub>R</sub> set to 1 (slave address y detection enabled)

This flag is set 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 IC SER set to 1 (slave address y detection enabled)  
This flag is set 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 IC SER set to 1 (slave address y detection enabled)  
This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the frame.

### GCA Flag (General Call Address Detection Flag)

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]) with the GCAE bit in IC SER set to 1 (general call address detection is 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 is enabled)  
This flag is set 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 Flag)

[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 is 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 is enabled)  
This flag is set 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 is enabled)  
This flag is set 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 Flag)

[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 is enabled)  
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

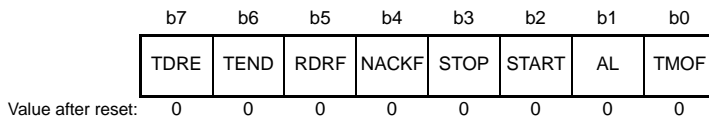
[Clearing conditions]

- 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 is enabled)  
This flag is set 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

### 29.2.10 I<sup>2</sup>C Bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h



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 Flag)

This flag is set to 1 when the RIIC recognizes timeout after the SCL0 line state remains unchanged for a certain period.  
[Setting condition]

- When the SCL0 line state remains unchanged for the period specified by the ICMR2.TMOH, TMOL, and TMOS bits while the ICFER.TMOE bit is 1 (the timeout function is enabled) in master mode or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

#### AL Flag (Arbitration-Lost Flag)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is issued or an address and data are transmitted. The RIIC monitors the level on the SDA0 line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL flag to 1 to indicate that the bus is occupied by another device.

The RIIC can also 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 SDA0 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 SDA0 line is driven low while the internal SDA output is at a high level (the SDA0 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 SDA0 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 SDA0 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 SDA0 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 flag after reading AL = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

**Table 29.4 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 SDA0 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 Flag)

[Setting condition]

- When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

### STOP Flag (Stop Condition Detection Flag)

[Setting condition]

- When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

**NACKF Flag (NACK Detection Flag)**

[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, set the NACKF flag to 0.

**RDRF Flag (Receive Data Full Flag)**

[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 set 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 Flag)**

[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 Flag)**

[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 received slave address matches while the TRS bit is 1

[Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is set to 0
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: When the NACKF flag is set to 1 while the NACKEN 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.



### 29.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC0.SARL1 0008 830Ch, RIIC0.SARL2 0008 830Eh



Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	A slave address is set.	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	A slave address is set.	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 8 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.

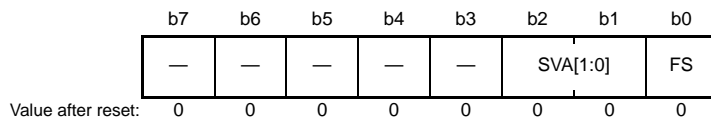
#### SVA[6:0] Bits (7-Bit Address/10-Bit Address Lower Bits)

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 8 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.

### 29.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC0.SARU1 0008 830Dh, RIIC0.SARU2 0008 830Fh



Bit	Symbol	Bit Name	Description	R/W
b0	FS	7-Bit/10-Bit Address Format Select	0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R/W
b2, b1	SVA[1:0]	10-Bit Address Upper Bits	A slave address is set.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### FS Bit (7-Bit/10-Bit Address Format Select)

This bit is used to select 7-bit address or 10-bit address for slave address y (in SARLy and SARUy).

When the SARyE bit in ICSEr is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the settings of the SVA[1:0] bits and the SVA0 bit in SARLy are ignored.

When the SARyE bit in ICSEr is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[1:0] bits and SARLy are valid.

While the SARyE bit in ICSEr is 0 (SARLy and SARUy disabled), the setting of the SARUy.FS bit is invalid.

#### SVA[1:0] Bits (10-Bit Address Upper Bits)

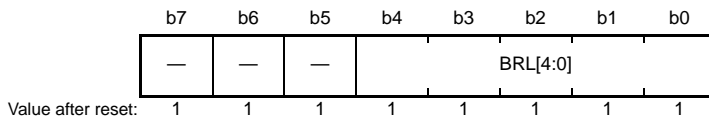
When the 10-bit address format is selected (FS = 1), these bits function as the upper 2 bits of a 10-bit address.

When the SARyE bit in ICSEr 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.

### 29.2.13 I<sup>2</sup>C Bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 0008 8310h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRL[4:0]	Bit Rate Low-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

ICBRL is a 5-bit register to set the low-level period of SCL clock.

It also works to generate the data setup time for automatic SCL low-hold operation (refer to section 29.8, Automatic Low-Hold Function for SCL); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time\*1.

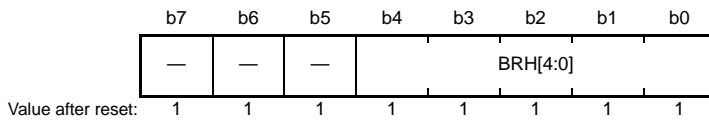
ICBRL counts the low-level period with the internal reference clock source (IIC $\phi$ ) specified by the CKS[2:0] bits in ICMR1.

If the digital noise filter is enabled (the NFE bit in ICFER is 1), set the ICBRL register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

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])

### 29.2.14 I<sup>2</sup>C Bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRH[4:0]	Bit Rate High-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.

If the digital noise filter is enabled (the NFE bit in ICFER is 1), set the ICBRH register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

The I<sup>2</sup>C transfer rate and the SCL clock duty are calculated using the following expression.

$$\text{Transfer rate} = 1 / \{ [(ICBRH + 1) + (ICBRL + 1)] / IIC\phi * 1 + SCL0 \text{ line rising time [tr]} + SCL0 \text{ line falling time [tf]} \}$$

$$\text{Duty cycle} = \{ SCL0 \text{ line rising time [tr]} * 2 + (ICBRH + 1) / IIC\phi \} / \{ SCL0 \text{ line falling time [tf]} * 2 + (ICBRL + 1) / IIC\phi \}$$

Note 1. IIC $\phi$  = PCLK  $\times$  Division ratio

Note 2. The SCL0 line rising time [tr] and SCL0 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 29.5 lists examples of ICBRH/ICBRL settings.

**Table 29.5** Examples of ICBRH/ICBRL Settings for Transfer Rate

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	8			10			12.5		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	100b	22 (F6h)	25 (F9h)	101b	13 (EDh)	15 (EFh)	101b	16 (F0h)	20 (F4h)
50	010b	16 (F0h)	19 (F3h)	010b	21 (F5h)	24 (F8h)	011b	12 (ECh)	15 (EFh)
100	001b	15 (EFh)	18 (F2h)	001b	19 (F3h)	23 (F7h)	001b	24 (F8h)	29 (FDh)
400	000b	4 (E4h)	10 (EAh)	000b	5 (E5h)	12 (ECh)	000b	7 (E7h)	16 (F0h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	16			20			25		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	101b	22 (F6h)	25 (F9h)	110b	13 (EDh)	15 (EFh)	110b	16 (F0h)	20 (F4h)
50	011b	16 (F0h)	19 (F3h)	011b	21 (F5h)	24 (F8h)	100b	12 (ECh)	15 (EFh)
100	010b	15 (EFh)	18 (F2h)	010b	19 (F3h)	23 (F7h)	010b	24 (F8h)	29 (FDh)
400	000b	9 (E9h)	20 (F4h)	000b	11 (EBh)	25 (F9h)	001b	7 (E7h)	16 (F0h)

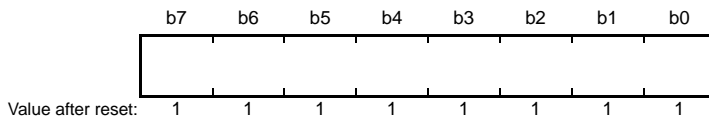
  

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)		
	30		
	CKS[2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)
50	100b	15 (EFh)	18 (F2h)
100	010b	2 (E2h)	3 (E3h)
400	001b	8 (E8h)	19 (F3h)

Note: ICBRH/ICBRL settings in these tables are calculated using the following values:  
SCL0 line rising time (tr): 100 kbps or less, [Sm]: 1000 ns, 400 kbps or less, [Fm]: 300 ns  
SCL0 line falling time (tf): 400 kbps or less, [Sm/Fm]: 300 ns  
For the specified values of SCL0 line rising time (tr) and SCL0 line falling time (tf), see the I<sup>2</sup>C bus standard from NXP Semiconductors.

### 29.2.15 I<sup>2</sup>C Bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h



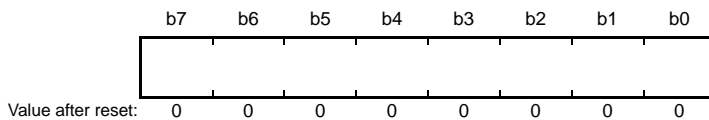
When ICDRT detects a space in the I<sup>2</sup>C bus shift register (ICDRS), it transfers the transmit data that has been written to ICDRT to ICDRS and starts transmitting data in transmit mode.

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 (TXI) request is generated.

### 29.2.16 I<sup>2</sup>C Bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h



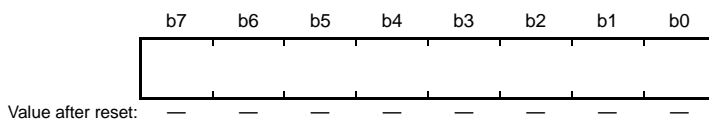
When 1 byte of data has been received, the received data is transferred from the I<sup>2</sup>C bus shift register (ICDRS) to ICDRR to enable the next data to be received.

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 (RXI) request is generated.

If ICDRR receives the next receive data before the current data is read from ICDRR (while the RDRF flag in ICSR2 is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

### 29.2.17 I<sup>2</sup>C Bus Shift Register (ICDRS)



ICDRS is an 8-bit shift register to transmit and receive data.

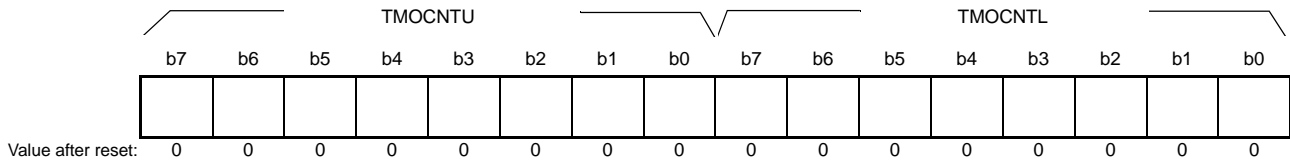
During transmission, transmit data is transferred from ICDRT to ICDRS and is sent from the SDA0 pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data has been received.

ICDRS cannot be accessed directly.

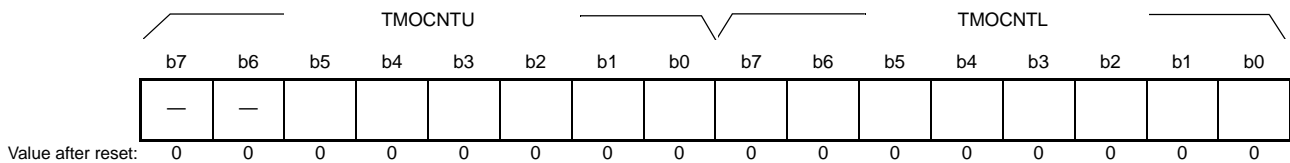
### 29.2.18 Timeout Internal Counter (TMOCNT)

Address(es): RIIC0.TMOCNTL 0008 830Ah, RIIC0.TMOCNTU 0008 830Bh

- TMOS = 0 (Long mode)



- TMOS = 1 (Short mode)



Note: Same address with ones of the slave address registers, SARL0, SARU0. Care should be taken.

- TMOCNTL register

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TMOCNTL	Timeout internal counter	Timeout internal counter low-order	W*1

Note 1. The value in timeout internal counter cannot be read. When the value is read, the read value is FFh.

- TMOCNTU register

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TMOCNTU	Timeout internal counter	Timeout internal counter high-order*1	W*2

Note 1. When TMOS = 1 (short mode is selected), bits b7 and b6 are reserved bits. They are writable. However, the value written is disabled.

Note 2. The value in timeout internal counter cannot be read. When the value is read, the read value is FFh.

The timeout internal counter (TMOCNTL/TMOCNTU) is initialized (TMOCNTL = 00h, TMOCNTU = 00h) after a reset, while ICCR1.IICRST = 1 or ICFER.TMOE = 1 and PCLK/1 is selected with ICMR1.CKS[2:0] = 000b setting, and when the counter clear conditions specified by the ICMR2.TMOH and TMOL bits (SCL rising edge/falling edge detection) are met.

Since the timeout internal counter is not automatically initialized when the ICMR1.CKS[2:0] bits are not 000b (PCLK/1), write 00h to the TMOCNTL and TMOCNTU counters as necessary for initialization.

The TMOCNTL and TMOCNTU counters can be accessed as 16-bit registers in 16-bit units.

For 16-bit access, access the address shown in the table below.

**Table 29.6 Register Allocation for 16-Bit Access**

Address	Upper 8 Bits	Lower 8 Bits
0008 830Ah	RIIC0.TMOCNTU	RIIC0.TMOCNTL

## 29.3 Operation

### 29.3.1 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 29.3 shows the I<sup>2</sup>C bus format, and Figure 29.4 shows the I<sup>2</sup>C bus timing.

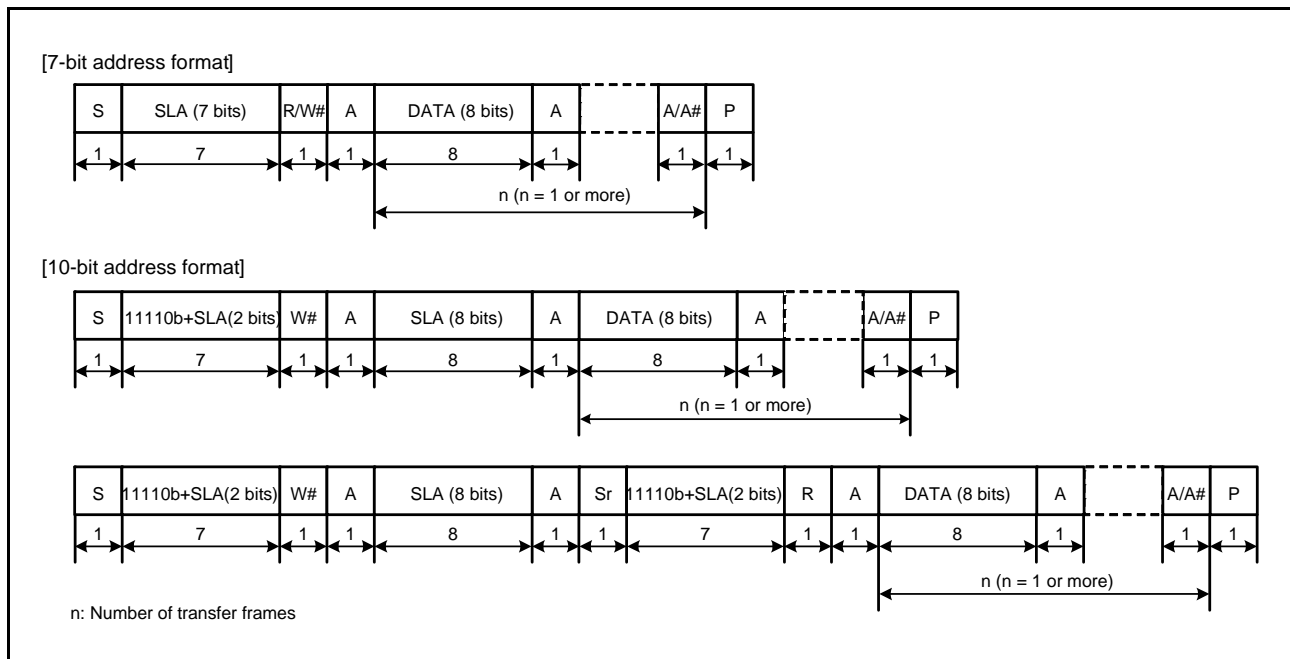


Figure 29.3 I<sup>2</sup>C Bus Format

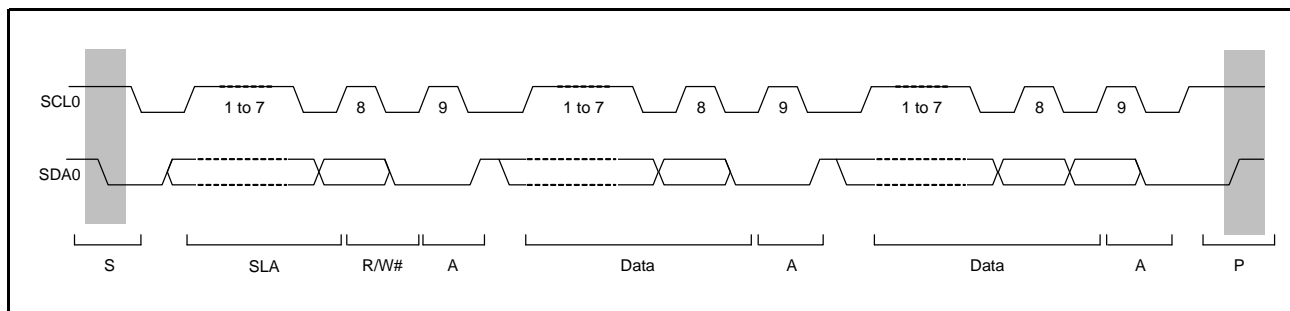


Figure 29.4 I<sup>2</sup>C Bus Timing (SLA = 7 Bits)

- S: Start condition. The master device drives the SDA0 line low from high level while the SCL0 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 SDA0 line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)
- A#: Not Acknowledge. The receive device drives the SDA0 line high.
- Sr: Restart condition. The master device drives the SDA0 line low from the high level after the setup time has elapsed with the SCL0 line at the high level.
- DATA: Transmitted or received data
- P: Stop condition. The master device drives the SDA0 line high from low level while the SCL0 line is at a high level.



### 29.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 29.5. Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (RIIC reset) with the ICCR1.ICE bit set to 0 (SCL0 and SDA0 pins in inactive state). This initializes the various flags and internal state of the ICSR1 register. After that, set registers SARLy, SARUy, ICSEr, ICMR1, ICBRH, and ICBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see Figure 29.5). When the necessary register settings have been completed, set the ICCR1.IICRST bit to 0 (releases the RIIC reset). This step is not necessary if initialization of the RIIC has already been completed.

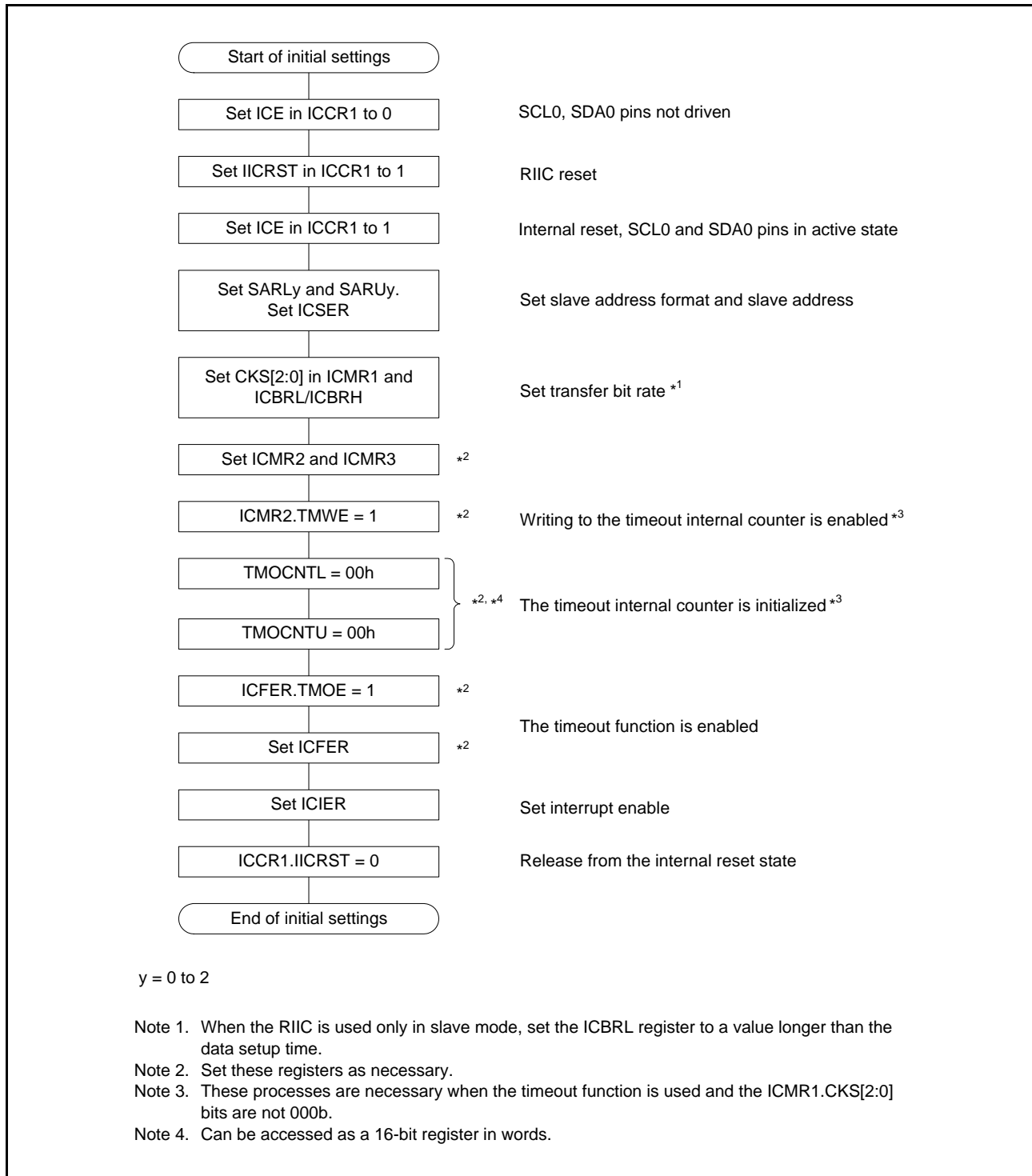


Figure 29.5 Example of RIIC Initialization Flowchart

### 29.3.3 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 29.6 shows an example of usage of master transmission and Figure 29.7 to Figure 29.9 show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

- (1) Initial settings. For details, refer to section 29.3.2, Initial Settings.
- (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 set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA0 line have matched while the ST bit is 1, the RIIC recognizes that 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 set 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 2 higher-order bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the 8 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 SCL0 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 sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Furthermore, it automatically sets the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
- (7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

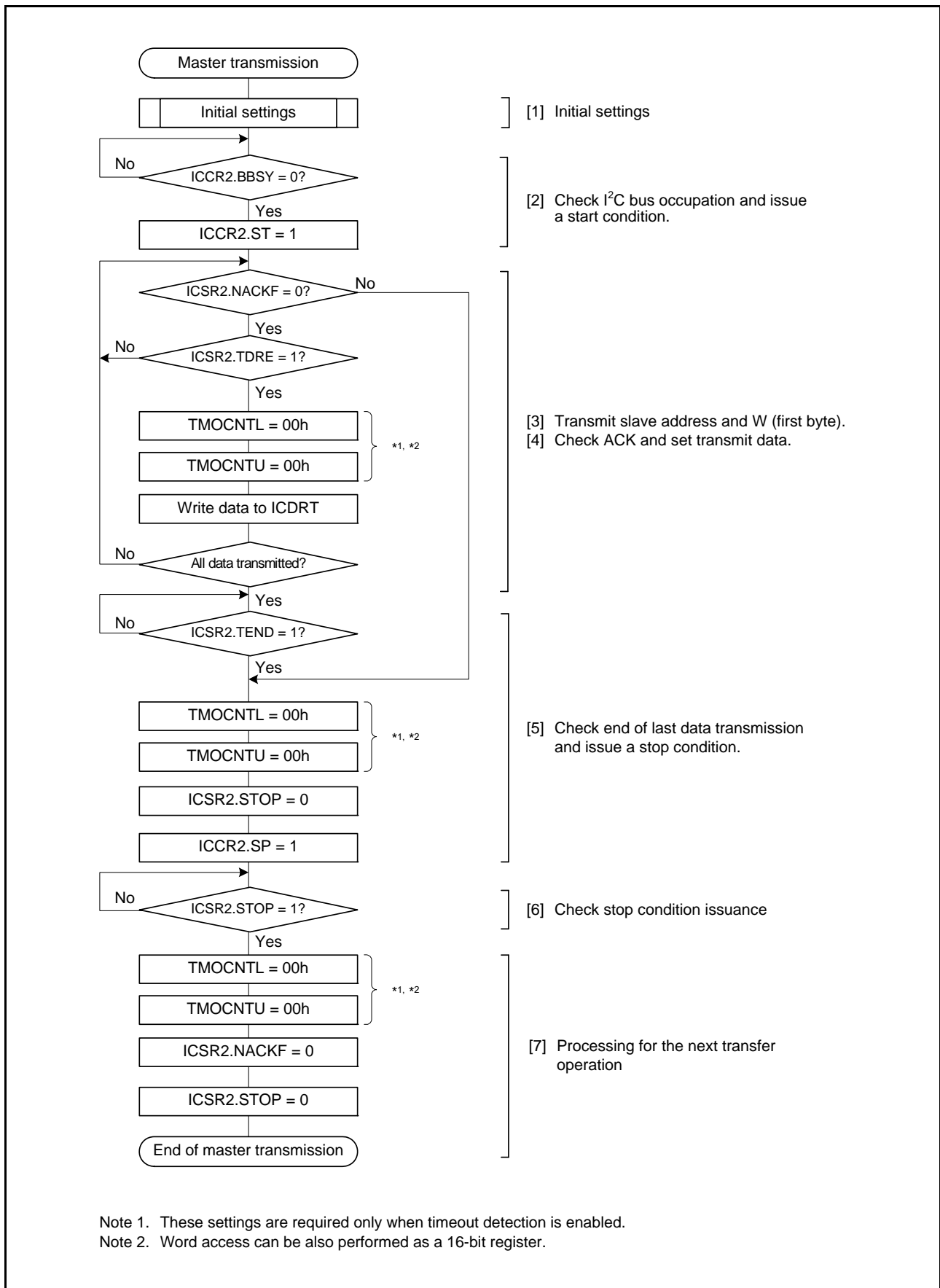


Figure 29.6 Example of Master Transmission Flowchart

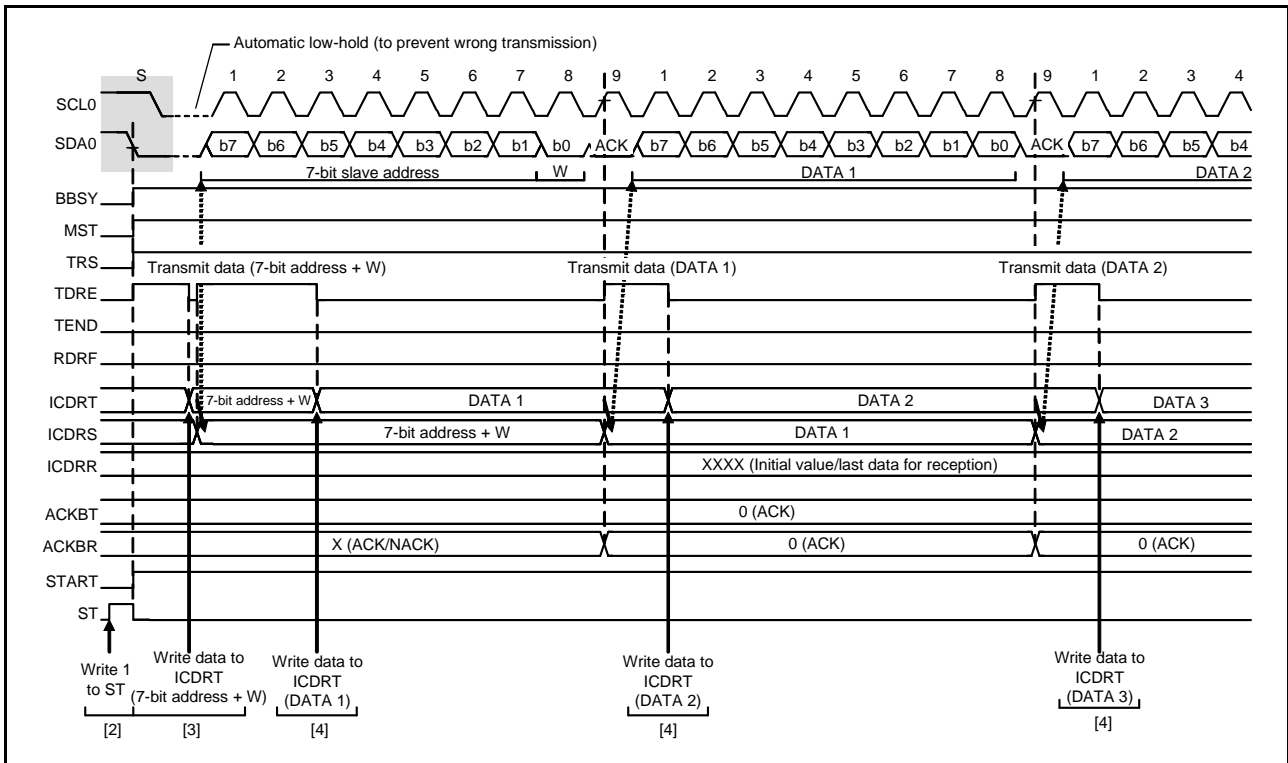


Figure 29.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

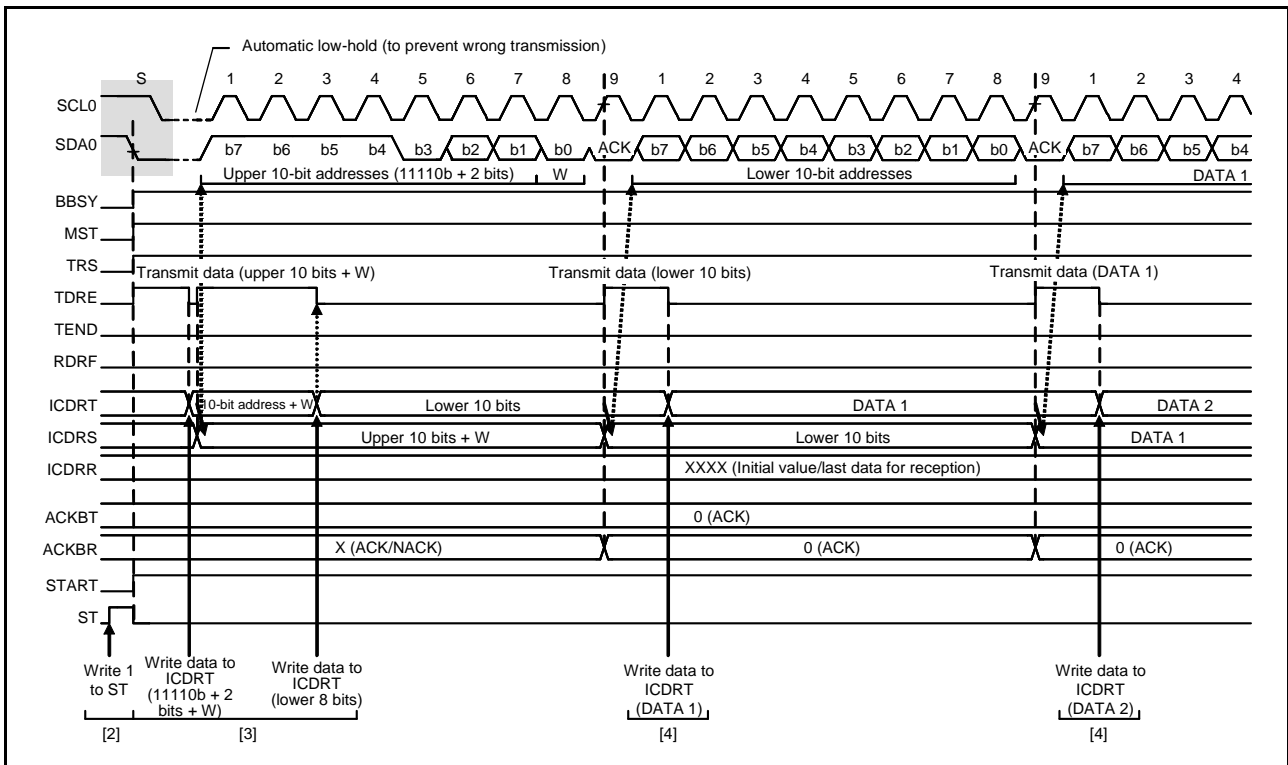
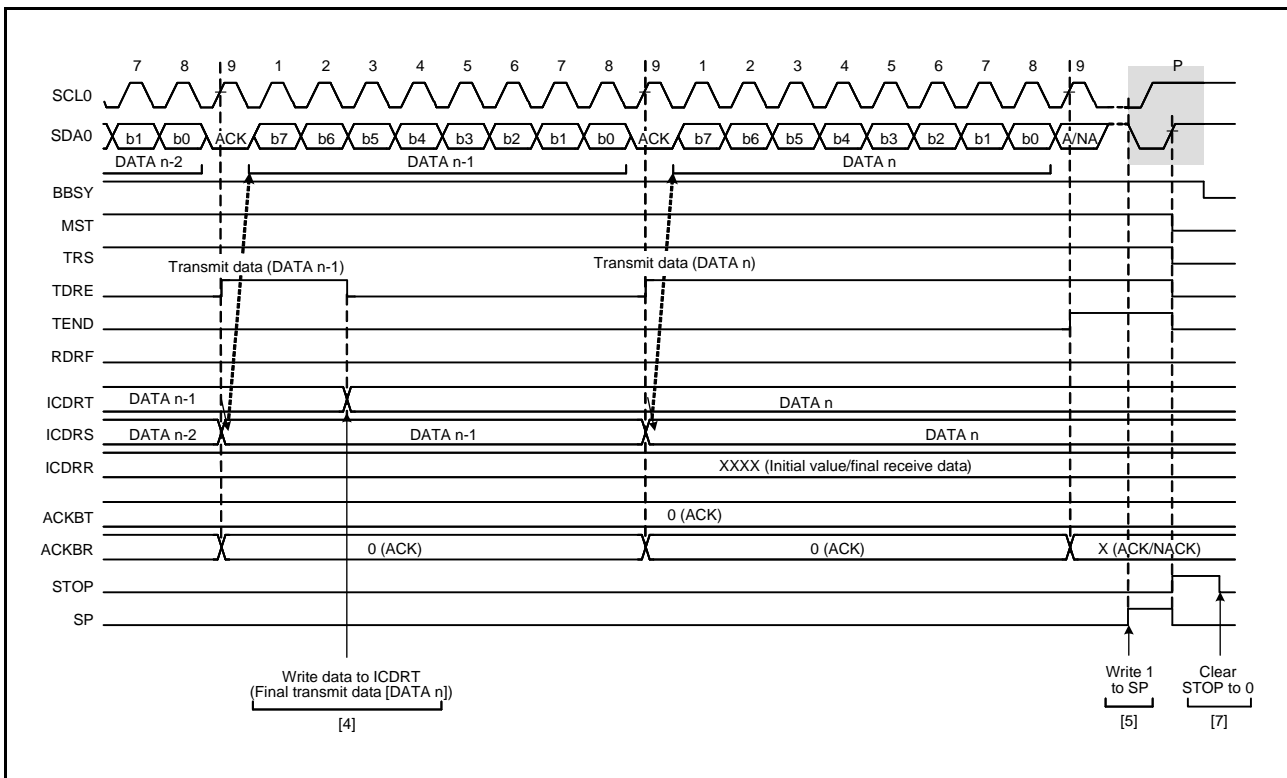


Figure 29.8 Master Transmit Operation Timing (2) (10-Bit Address Format)



**Figure 29.9 Master Transmit Operation Timing (3)**

### 29.3.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL clock, 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 29.10 and Figure 29.11 show examples of usage of master reception (7-bit address format) and Figure 29.12 to Figure 29.14 show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

- (1) Initial settings. For details, refer to section 29.3.2, Initial Settings.
- (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 set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA0 line have matched while the ST bit is 1, the RIIC recognizes that 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 set 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 set to 0 on the rising edge of the ninth cycle of SCL clock, placing the RIIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag is automatically set to 1.

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 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 ICDRR at this time will produce the received data, and the RDRF flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgement field received during the ninth cycle of SCL clock is returned as the value set in the ICMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the ICMR3.WAIT bit to 1 (for wait insertion) before reading the ICDRR (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ICMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL0 line to the low level on the falling 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 ICMR3.ACKBT bit to 1 (NACK).
- (7) After reading the byte before last from the ICDRR register, if the value of the ICSR2.RDRF flag is confirmed to be 1, write 1 to the 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 SCL0 line is released from the low-hold state.
- (8) Upon detecting the stop condition, the RIIC automatically sets 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 ICSR2.STOP flag to 1.
- (9) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

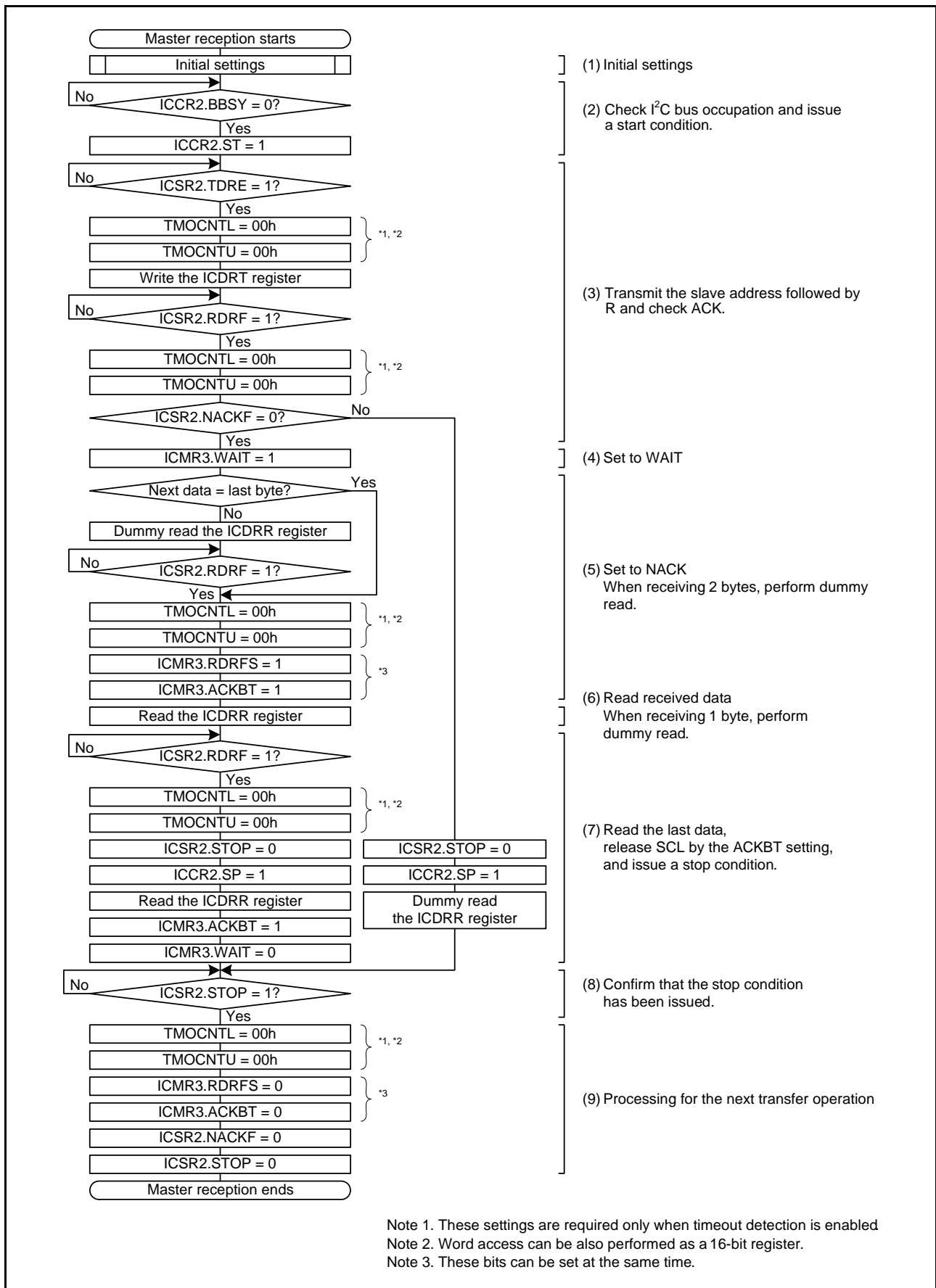


Figure 29.10 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes)

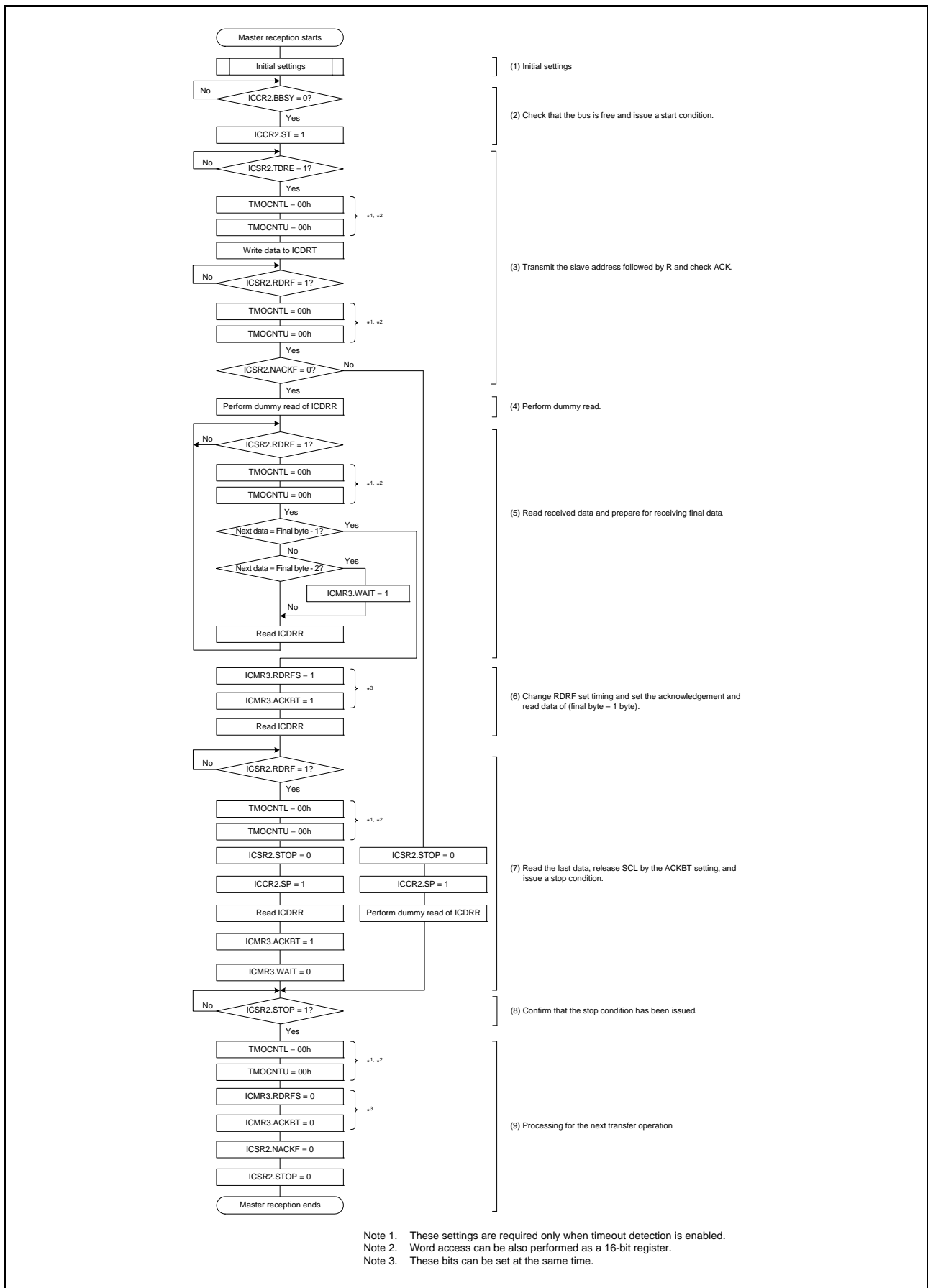


Figure 29.11 Example of Master Reception (7-Bit Address Format, 3 Bytes or More)



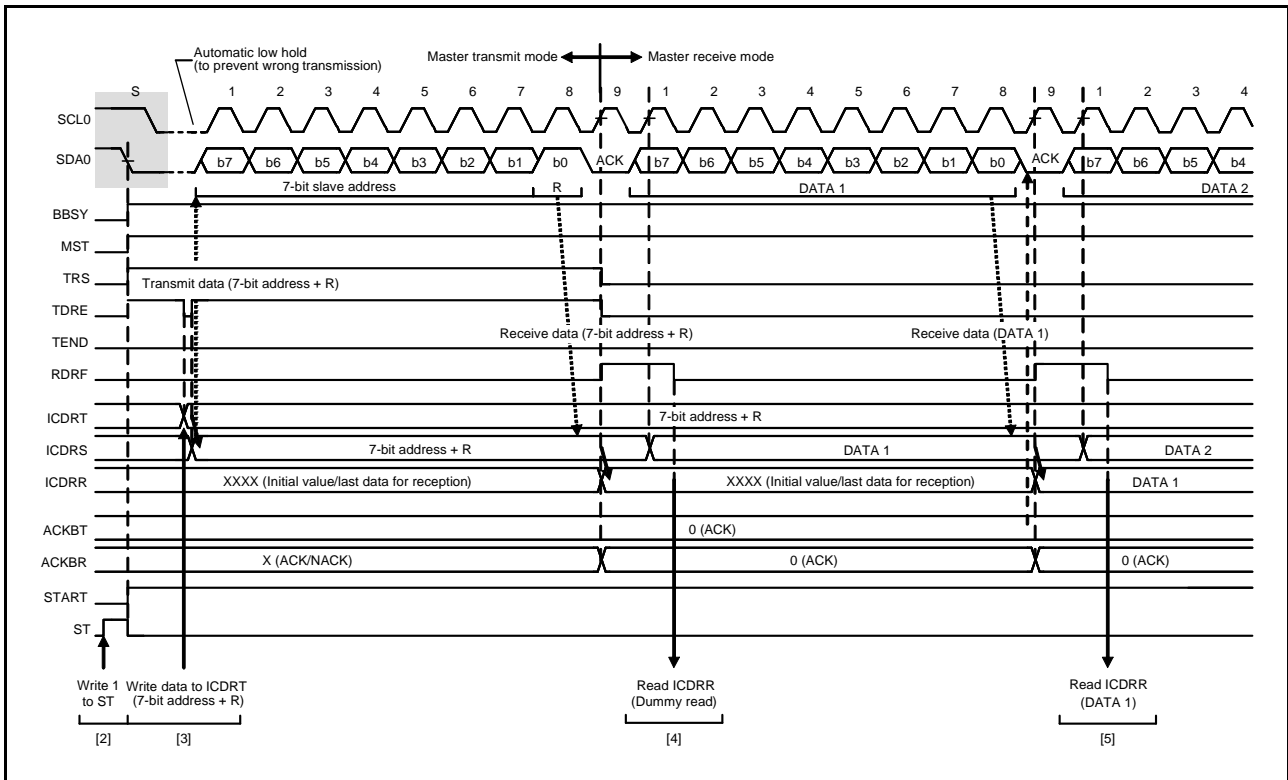


Figure 29.12 Master Receive Operation Timing (1) (7-Bit Address Format, When RDRFS = 0)

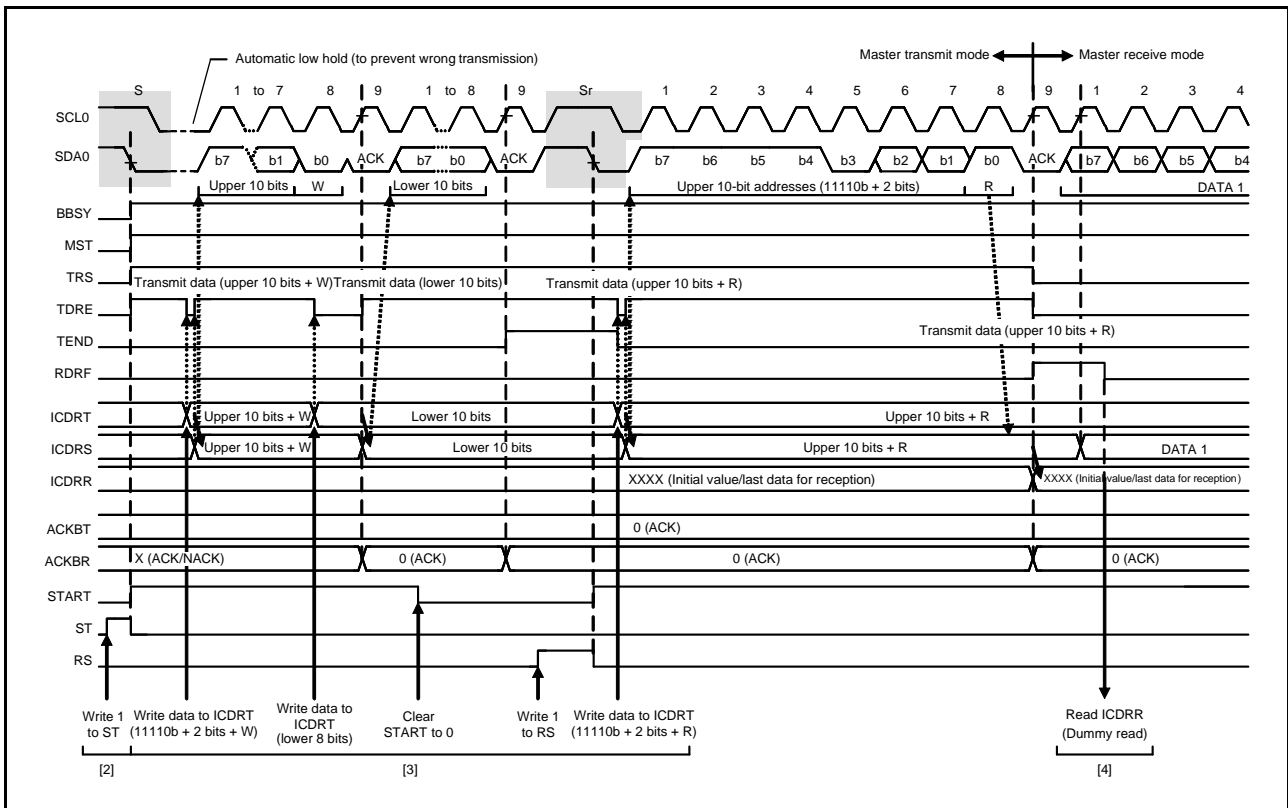


Figure 29.13 Master Receive Operation Timing (2) (10-Bit Address Format, When RDRFS = 0)

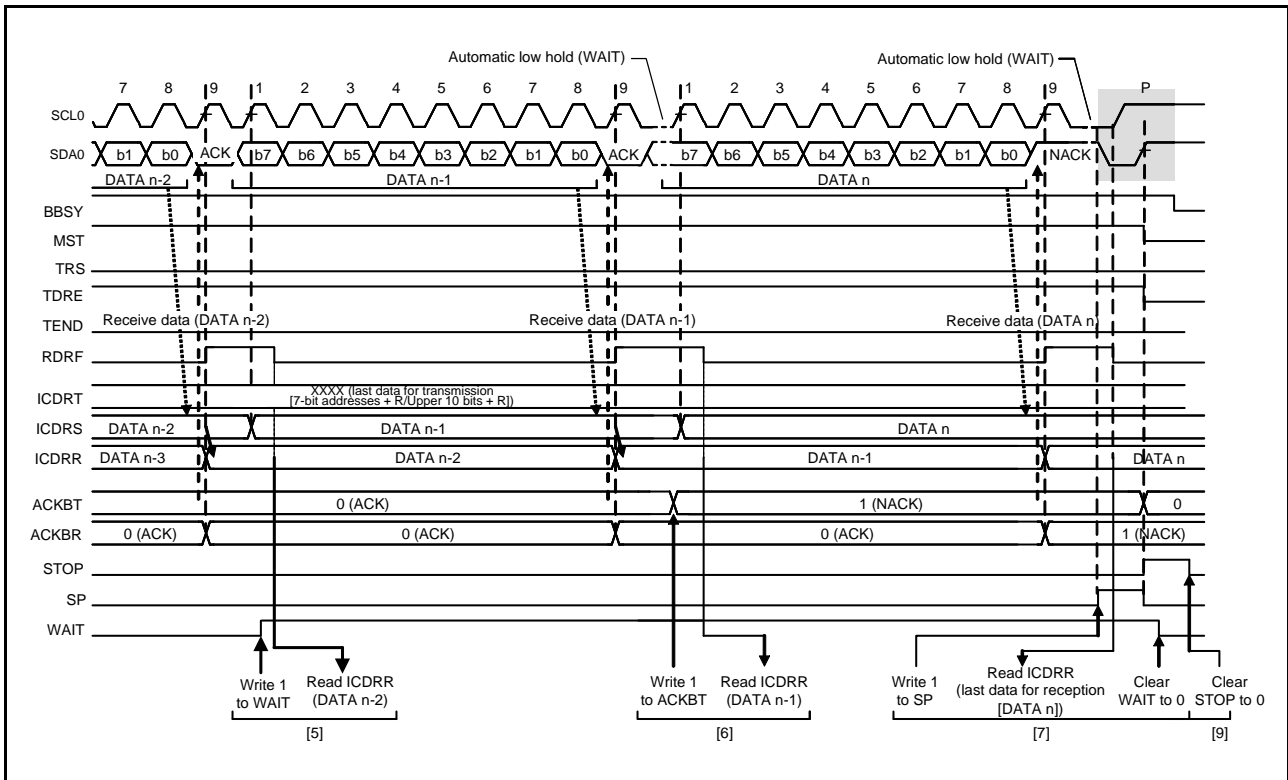


Figure 29.14 Master Receive Operation Timing (3) (When RDRFS = 0)

### 29.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL clock, the RIIC transmits data as a slave device, and the master device returns acknowledgements.

Figure 29.15 shows an example of usage of slave transmission and Figure 29.16 and Figure 29.17 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

- (1) Initial settings. For details, refer to section 29.3.2, Initial Settings.  
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.
- (2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AAS<sub>y</sub> (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 ICCR2.TRS bit and the ICSR2.TDRE flag to 1.
- (3) After the ICSR2.TDRE flag is confirmed to be 1, write the data for transmission to the ICDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives a NACK signal) while the ICFER.NACKE bit is 1, the RIIC suspends transfer of the next data.
- (4) Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCL0 line low on the 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 SCL0 line.
- (6) Upon detecting the stop condition, the RIIC automatically sets bits ICSR1.HOA, GCA, and AAS<sub>y</sub> (y = 0 to 2), flags ICSR2.TDRE and TEND, and the ICCR2.TRS bit to 0, and enters slave receive mode.
- (7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

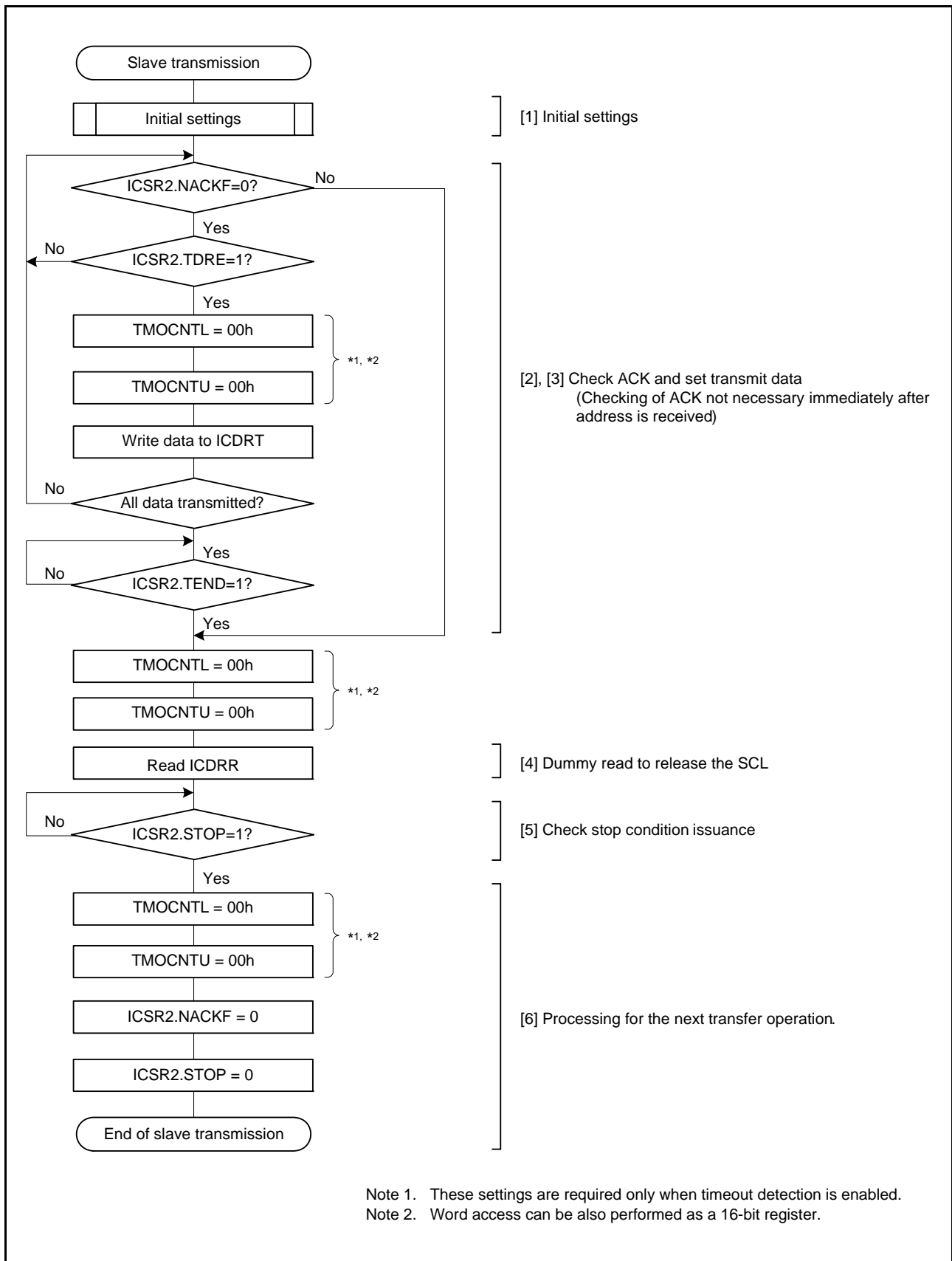


Figure 29.15 Example of Slave Transmission Flowchart

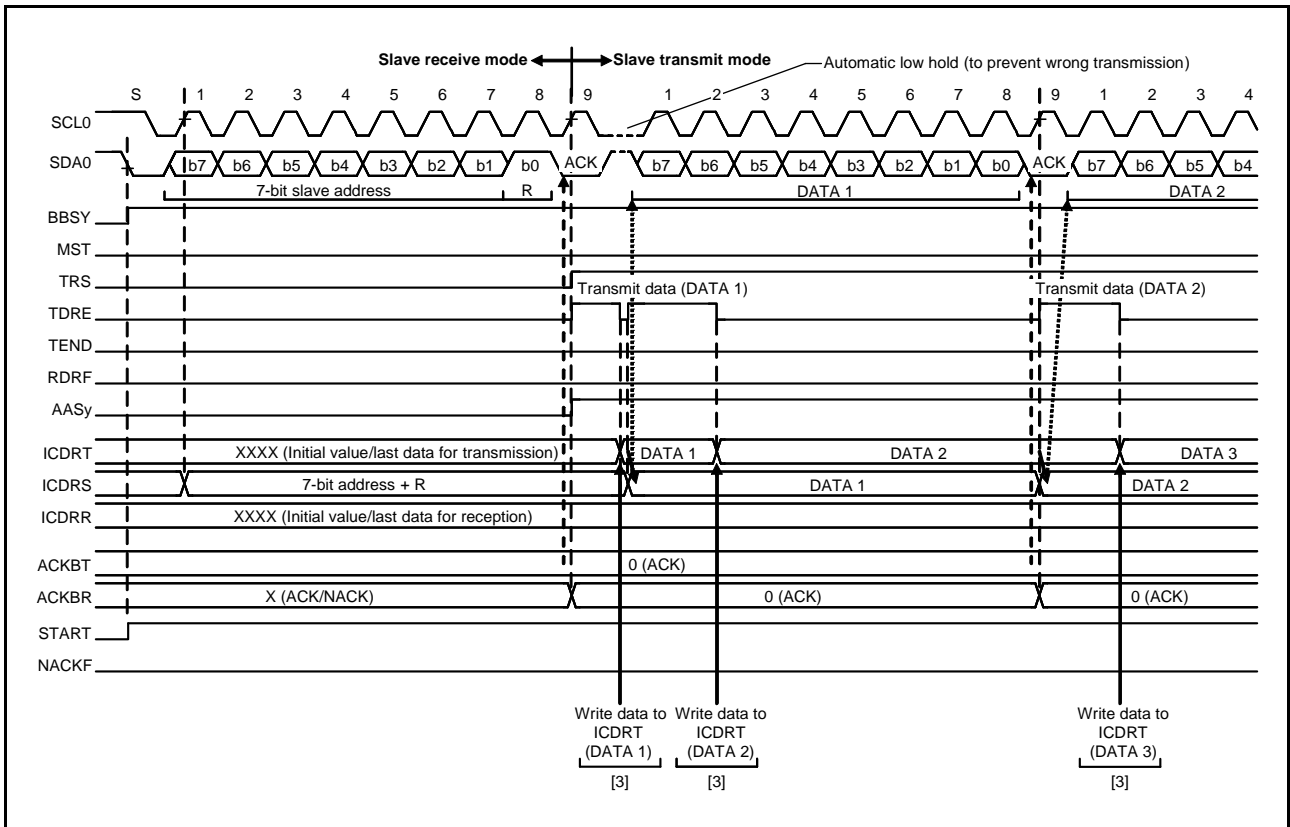


Figure 29.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

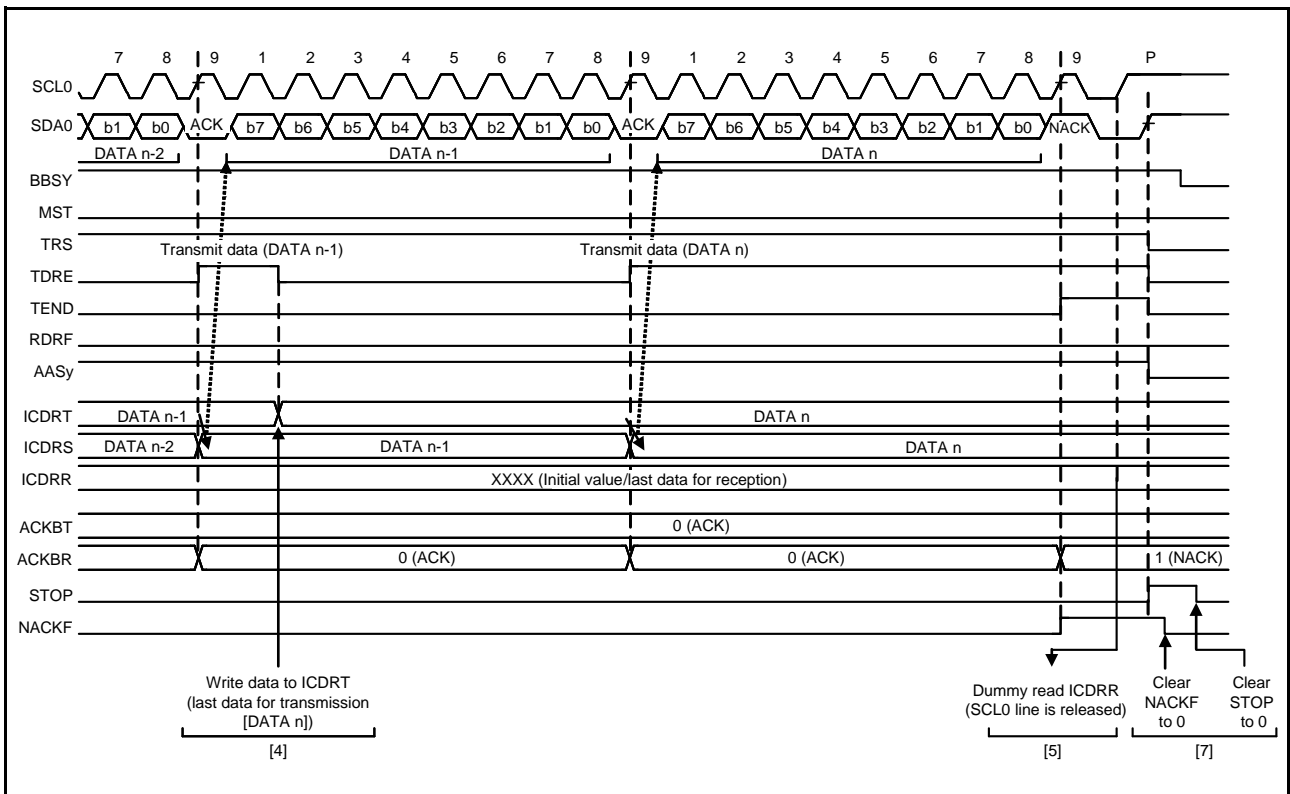


Figure 29.17 Slave Transmit Operation Timing (2)

### 29.3.6 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 29.18 shows an example of usage of slave reception and Figure 29.19 and Figure 29.20 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

- (1) Initial settings. For details, refer to section 29.3.2, Initial Settings.  
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.
- (2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AAS<sub>y</sub> (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 8 bits when the 10-bit address format is selected).
- (4) When ICDRR is read, the RIIC automatically sets 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 SCL0 line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading ICDRR releases the SCL0 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 AAS<sub>y</sub> (y = 0 to 2) to 0.
- (6) After checking that the ICSR2.STOP flag is 1, set the ICSR2.STOP flag to 0 for the next transfer operation.

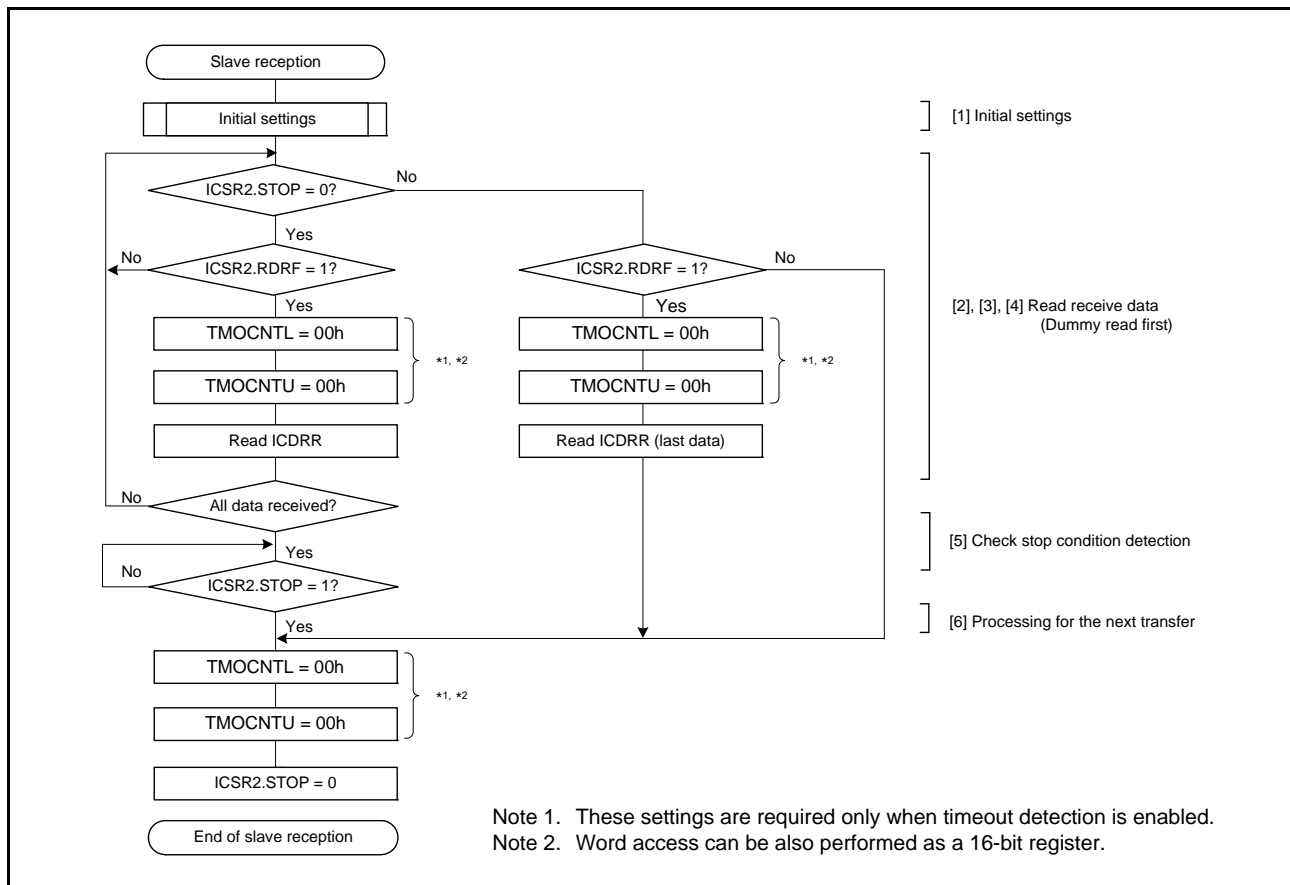


Figure 29.18 Example of Slave Reception Flowchart

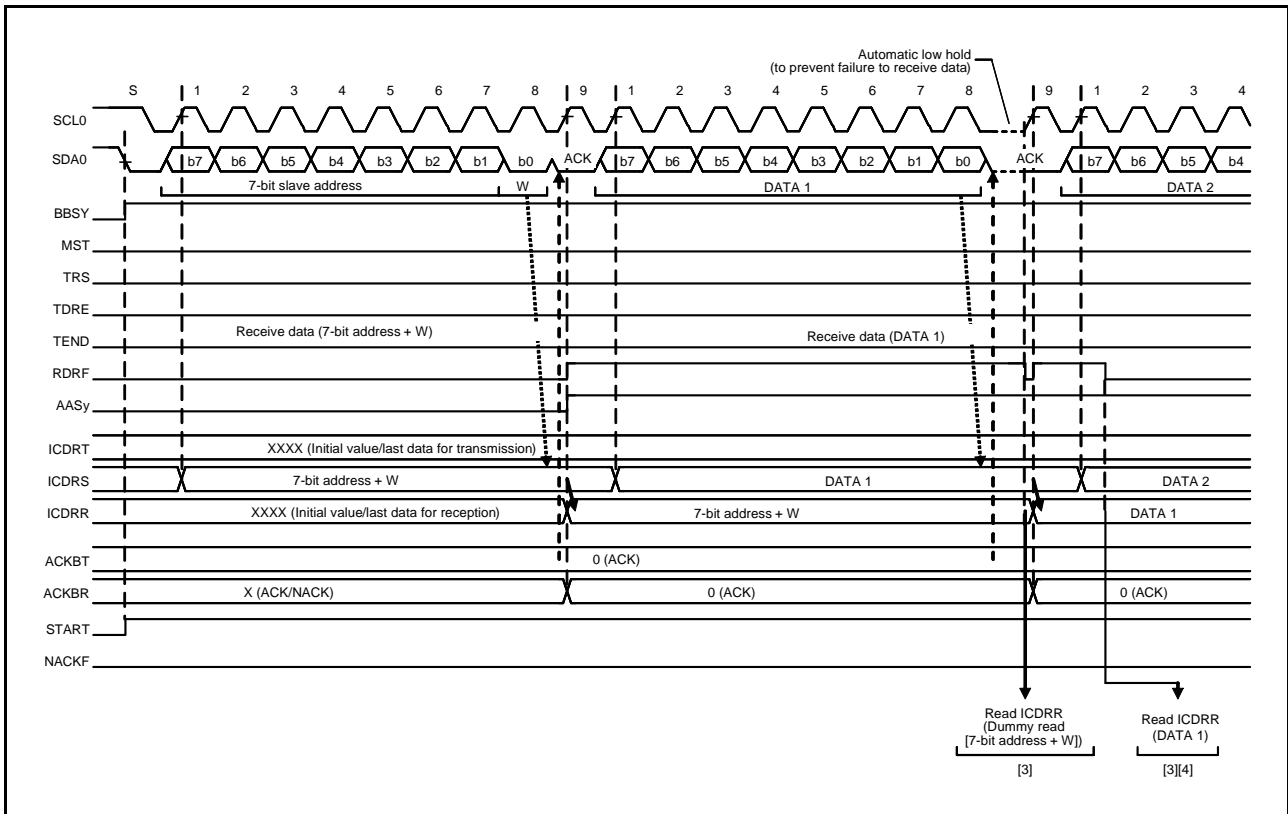


Figure 29.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

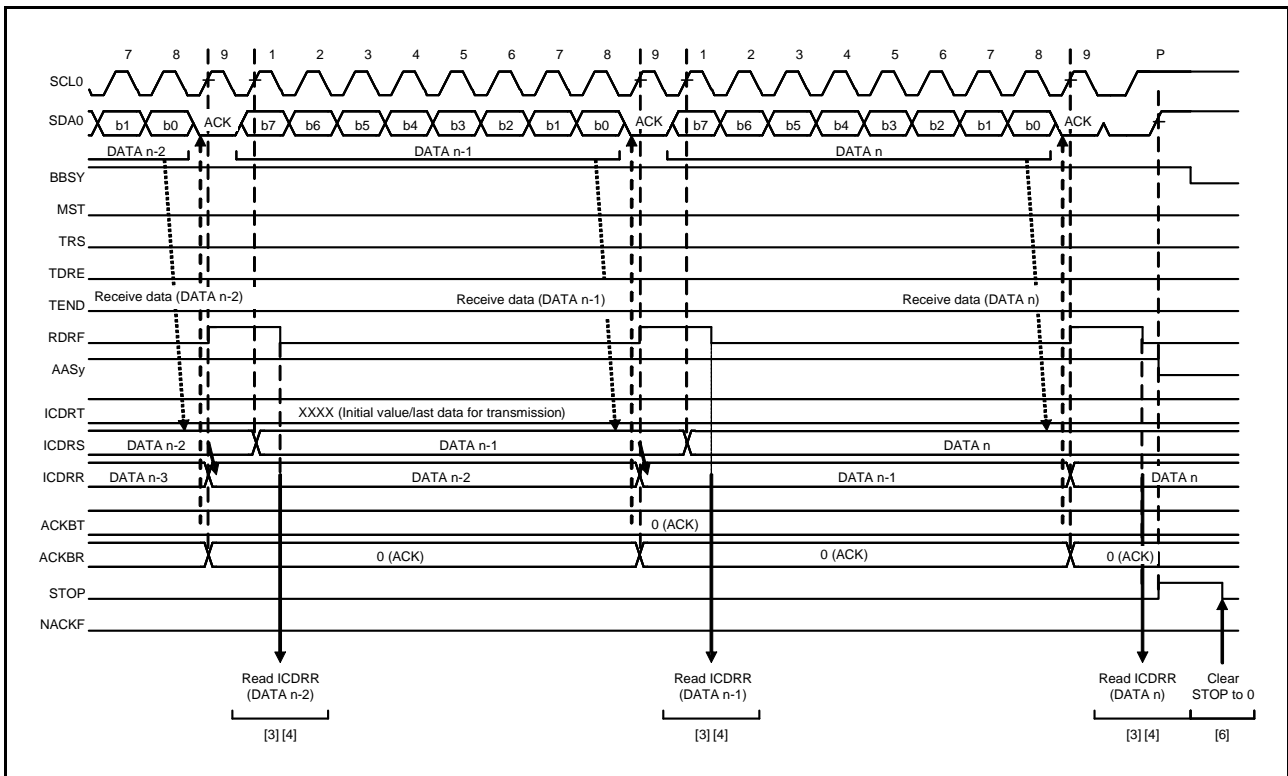


Figure 29.20 Slave Receive Operation Timing (2) (when RDRFS = 0)

## 29.4 SCL Synchronization Circuit

In generation of the SCL clock, the RIIC starts counting out the value for width at high level specified in ICBRH when it detects a rising edge on the SCL0 line and drives the SCL0 line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL0 line, it starts counting out the width at low level period specified in ICBRL, and then stops driving the SCL0 line (releases the line) once counting of the width at low level is complete. The SCL clock 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 SCL0 line while in master mode.

When the RIIC has detected a rising edge on the SCL0 line and thus started counting out the width at high level specified in ICBRH, and the level on the SCL0 line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL0 line low, and starts counting out the width at low level specified in ICBRL. When the RIIC finishes counting out the width at low level, it stops driving the SCL0 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 SCL0 line has been released. When the RIIC finishes outputting the low-level period of the SCL clock, the SCL0 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.

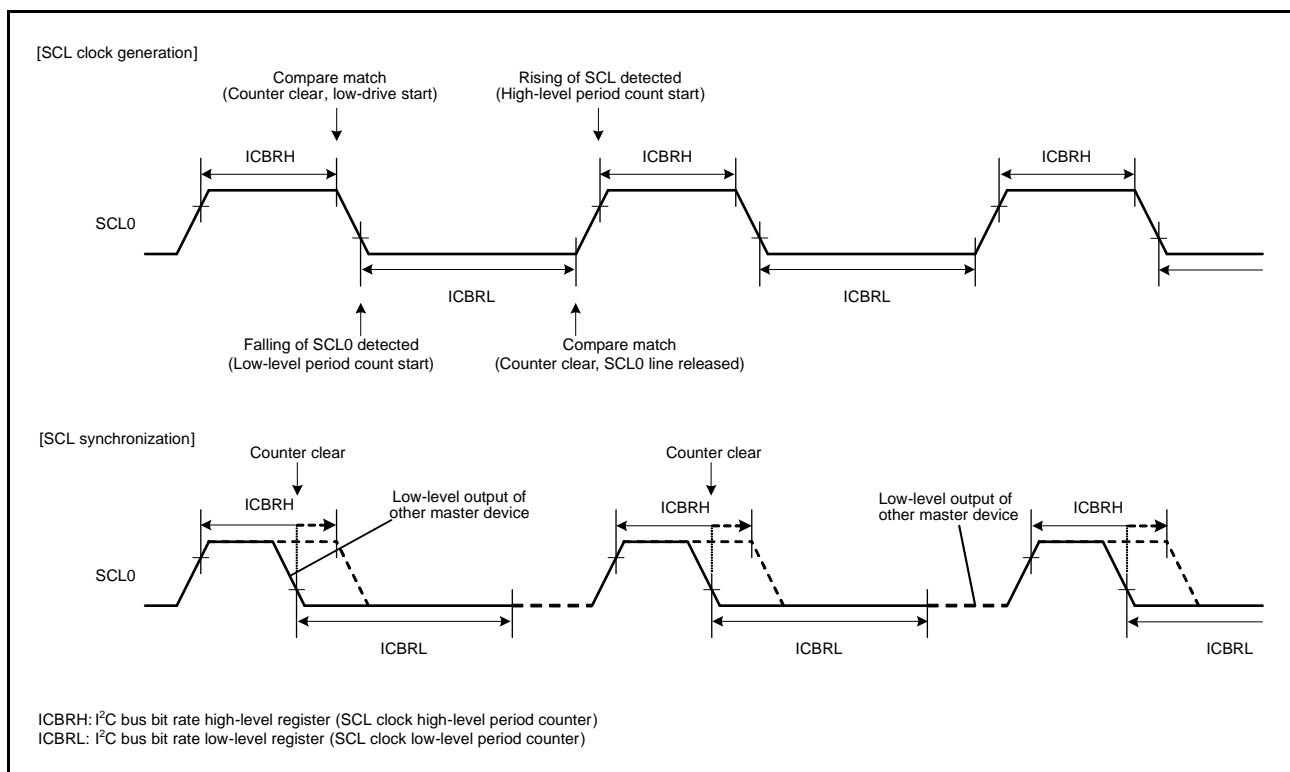


Figure 29.21 Generation and Synchronization of the SCL Signal from the RIIC



### 29.5 SDA Output Delay Function

The RIIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay function, 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 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 function is enabled by setting the SDDL[2:0] bits in ICMR2 to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay function is enabled (i.e. while the SDDL[2:0] bits in ICMR2 are set to any value other than 000b), the DLCS bit in ICMR2 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 ICMR2. 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.

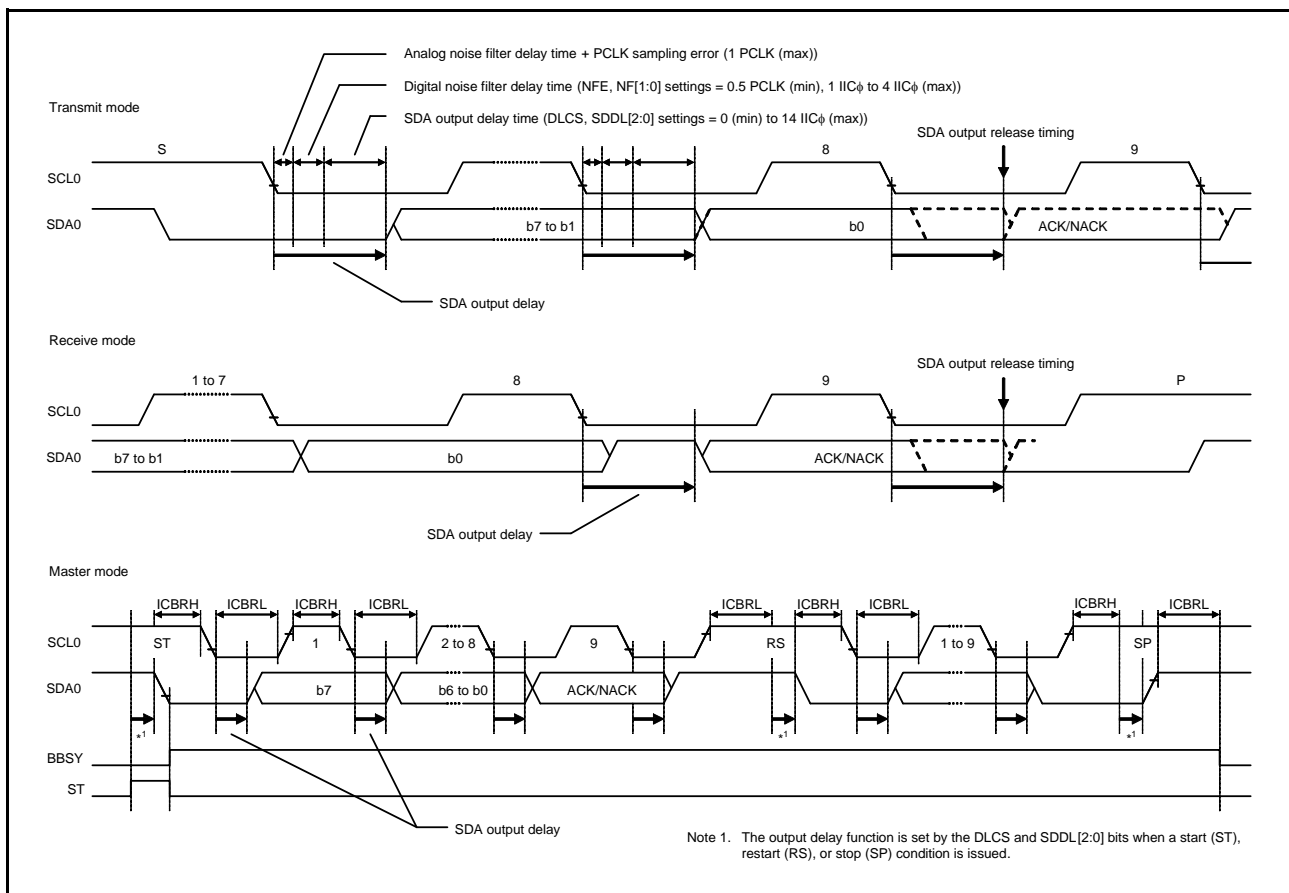


Figure 29.22 SDA Output Delay Function

## 29.6 Digital Noise Filter Circuits

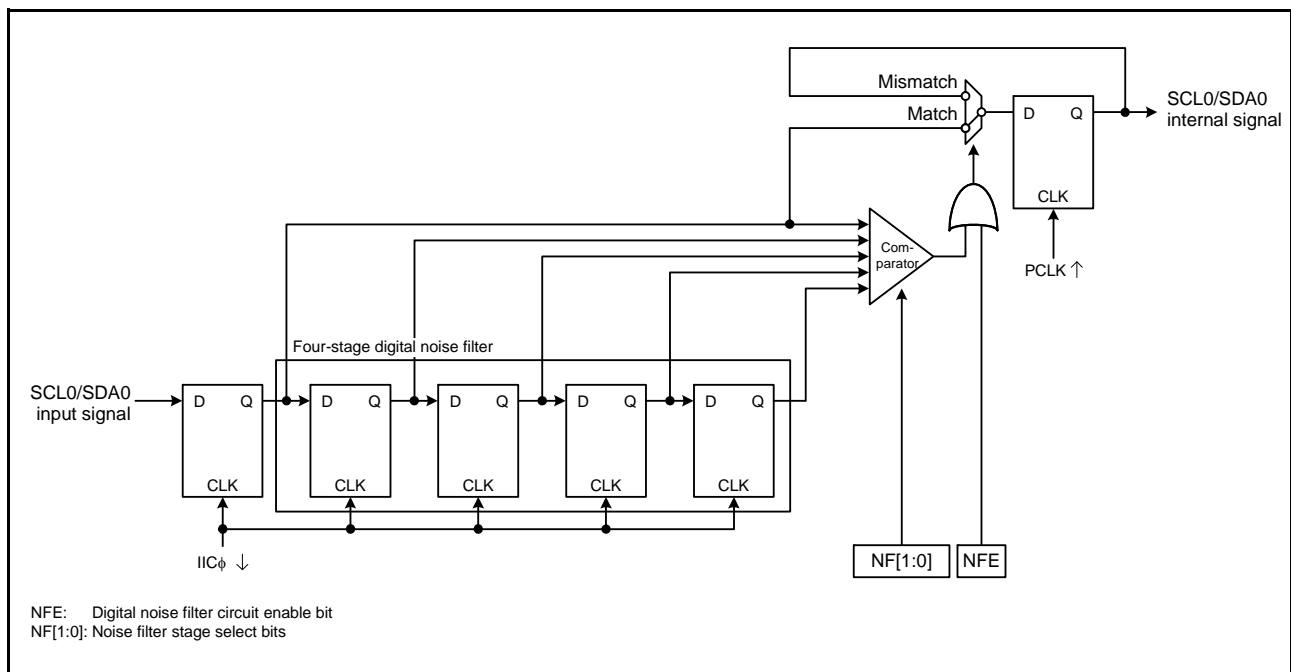
The states of the SCL0 and SDA0 pins are conveyed to the internal circuitry through analog noise-filter and digital noise-filter circuits. Figure 29.23 is a block diagram of the digital noise-filter circuit.

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 SCL0 pin (or SDA0 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 setting the ICFER.NFE bit to 0) and use only the analog noise filter circuit.



**Figure 29.23** Block Diagram of Digital Noise Filter Circuit

### 29.7 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.

#### 29.7.1 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 rising 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 (RXI) or transmit data empty interrupt (TXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 29.24 to Figure 29.26 show the AASy flag set timing in three cases.

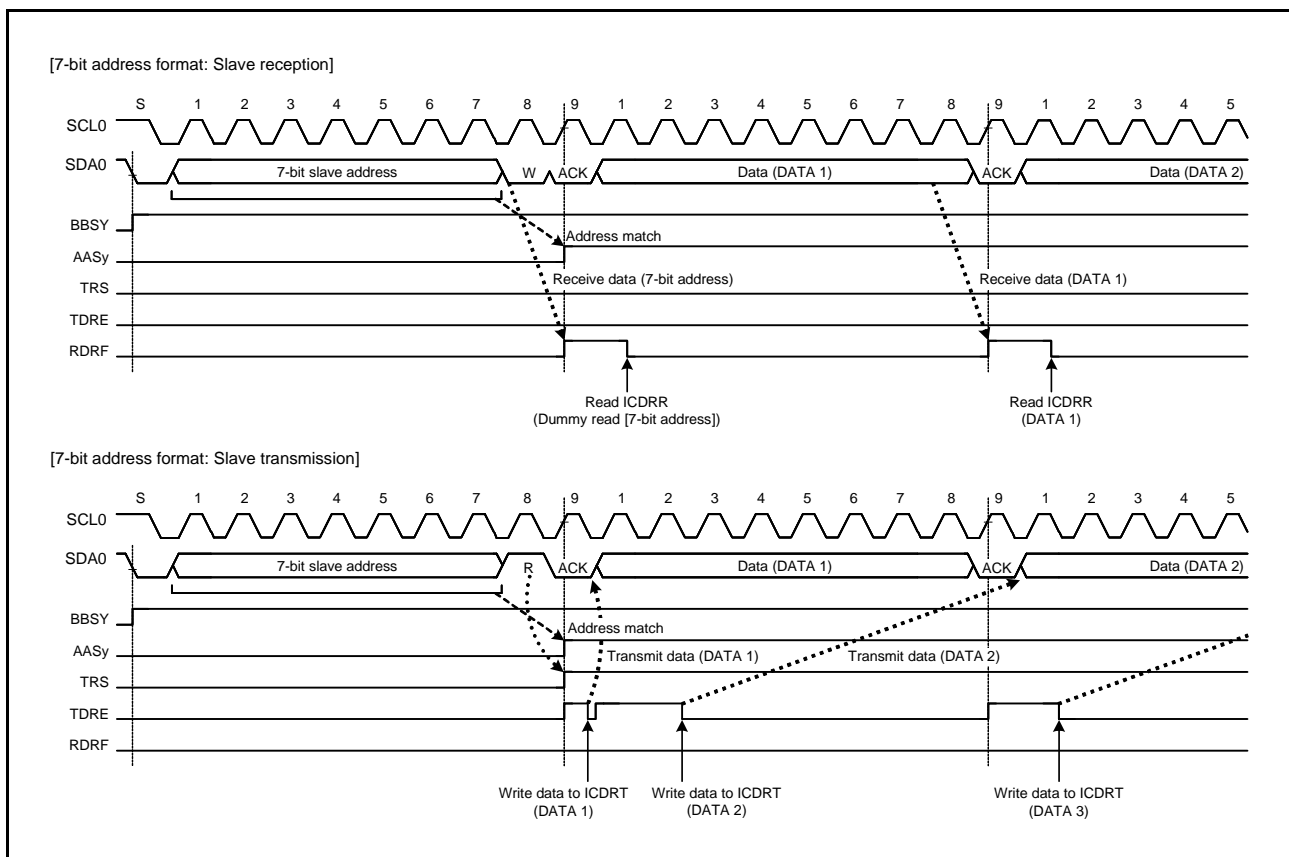
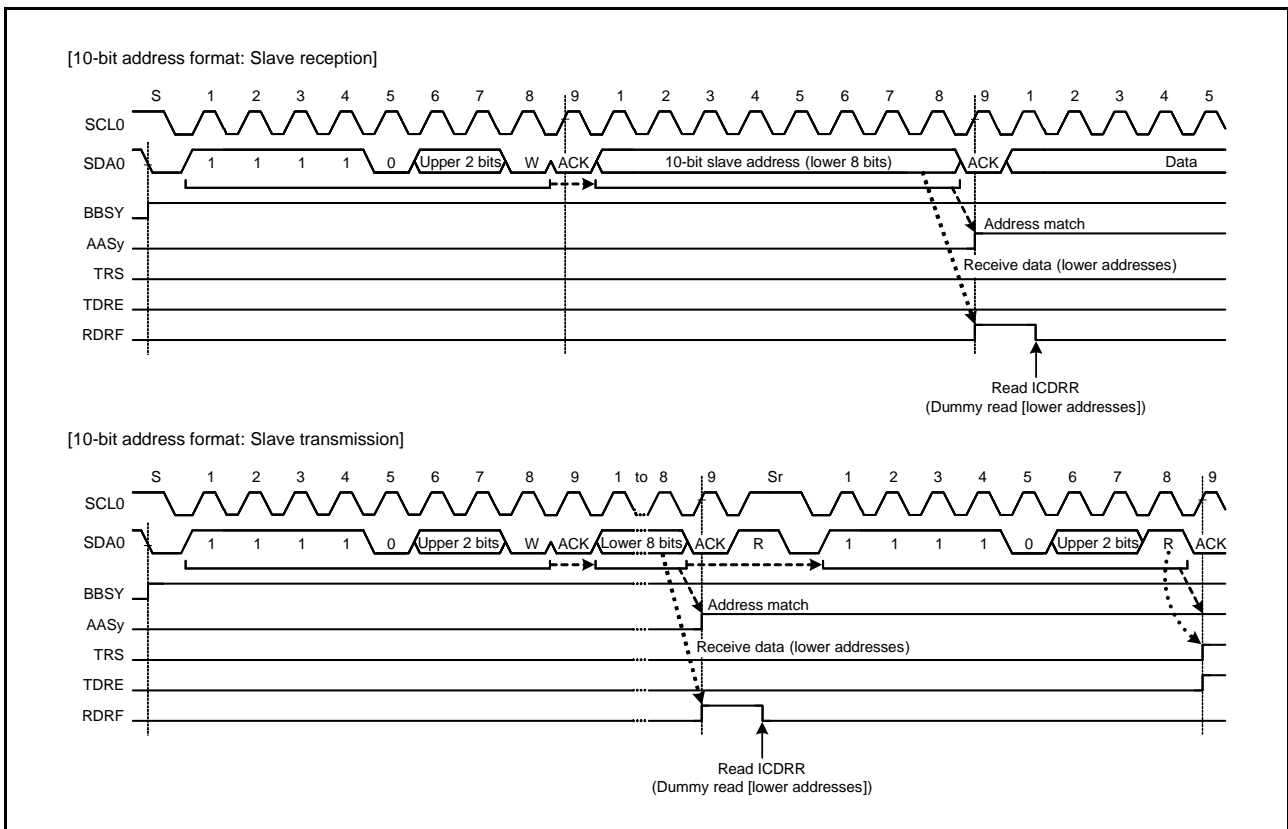
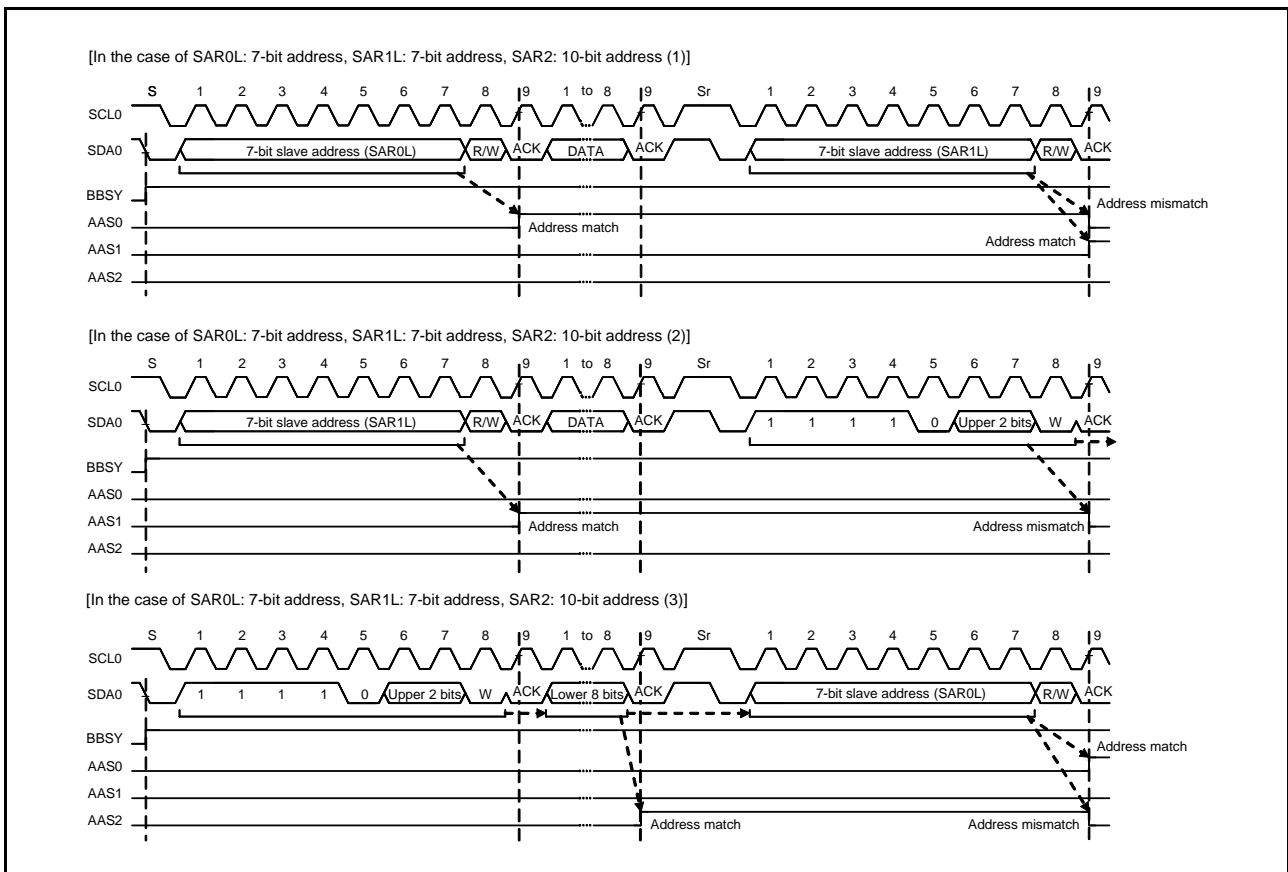


Figure 29.24 AASy Flag Set Timing with 7-Bit Address Format Selected



**Figure 29.25 AASy Flag Set Timing with 10-Bit Address Format Selected**



**Figure 29.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed**

### 29.7.2 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 IC SER 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 IC SER1 and the RDRF flag in IC SER2 are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a receive data full interrupt (RXI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

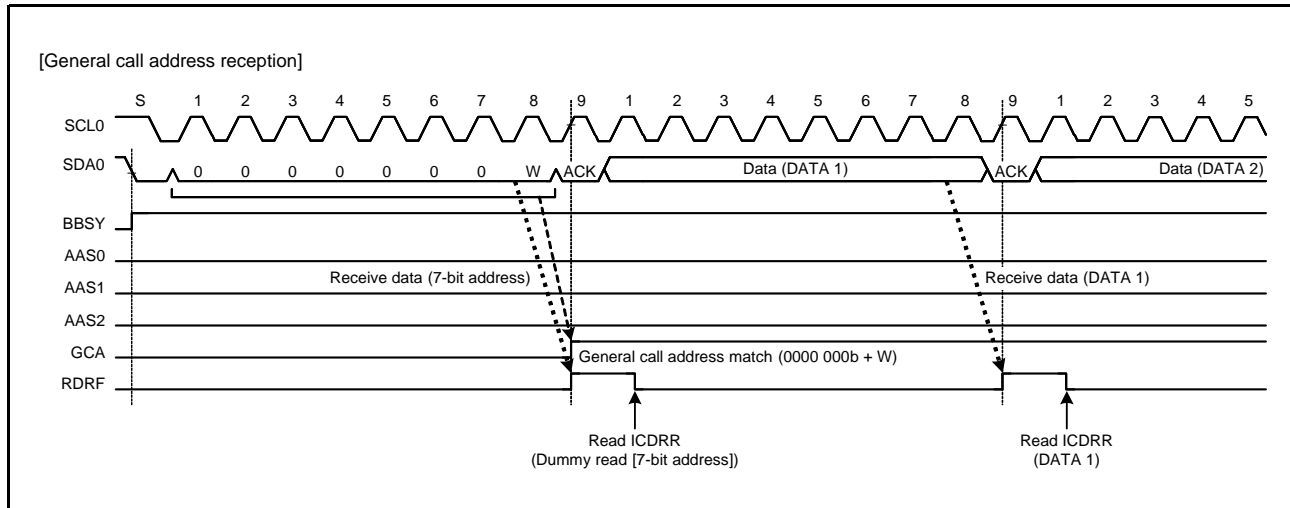


Figure 29.27 Timing of GCA Flag Setting during Reception of General Call Address

### 29.7.3 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 eighth 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 sets the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Furthermore, prepare the device-ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

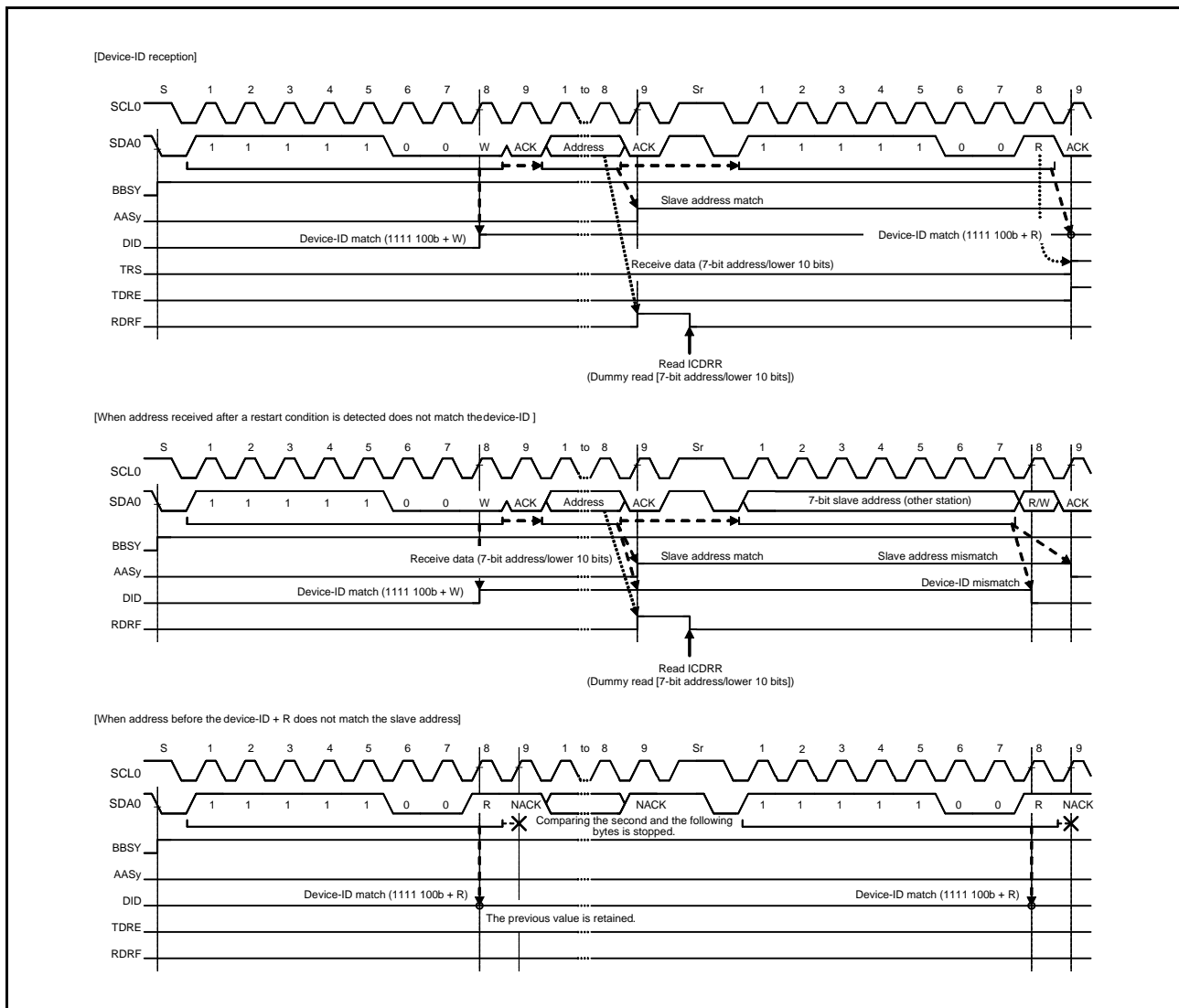


Figure 29.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

### 29.7.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the HOAE bit in IC SER 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 IC SR1 is set to 1 at the rising edge of the ninth SCL clock cycle, and at the same time, the RDRF flag in IC SR2 is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (RXI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit = 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.

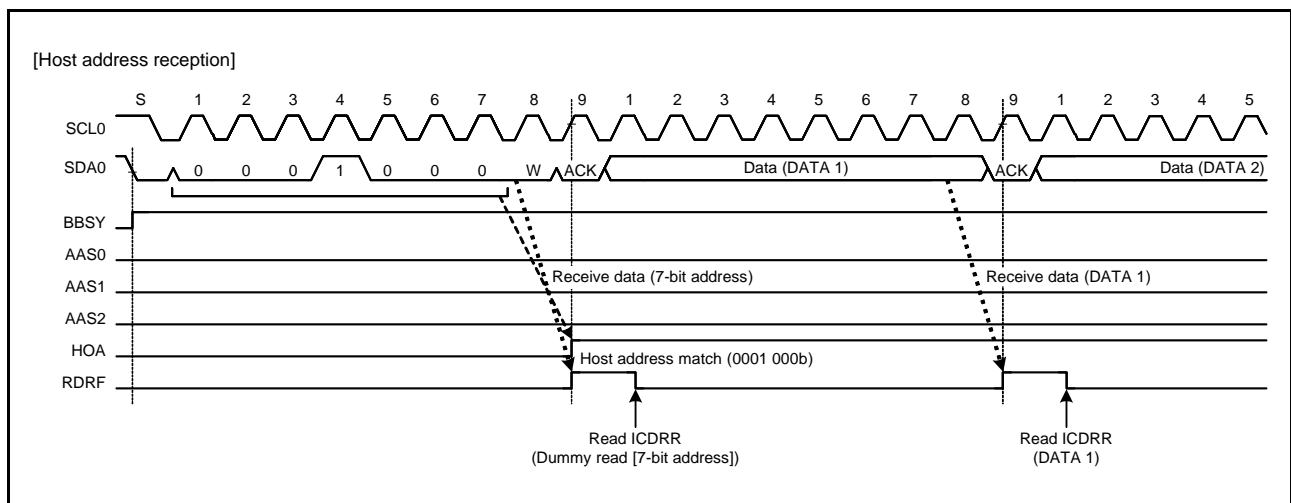


Figure 29.29 HOA Flag Set Timing during Reception of Host Address



## 29.8 Automatic Low-Hold Function for SCL

### 29.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (ICDRS) is empty when data have not been written to the I<sup>2</sup>C bus transmit data register (ICDRT) with the RIIC in transmission mode (TRS bit = 1 in ICCR2), the SCL0 line 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 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

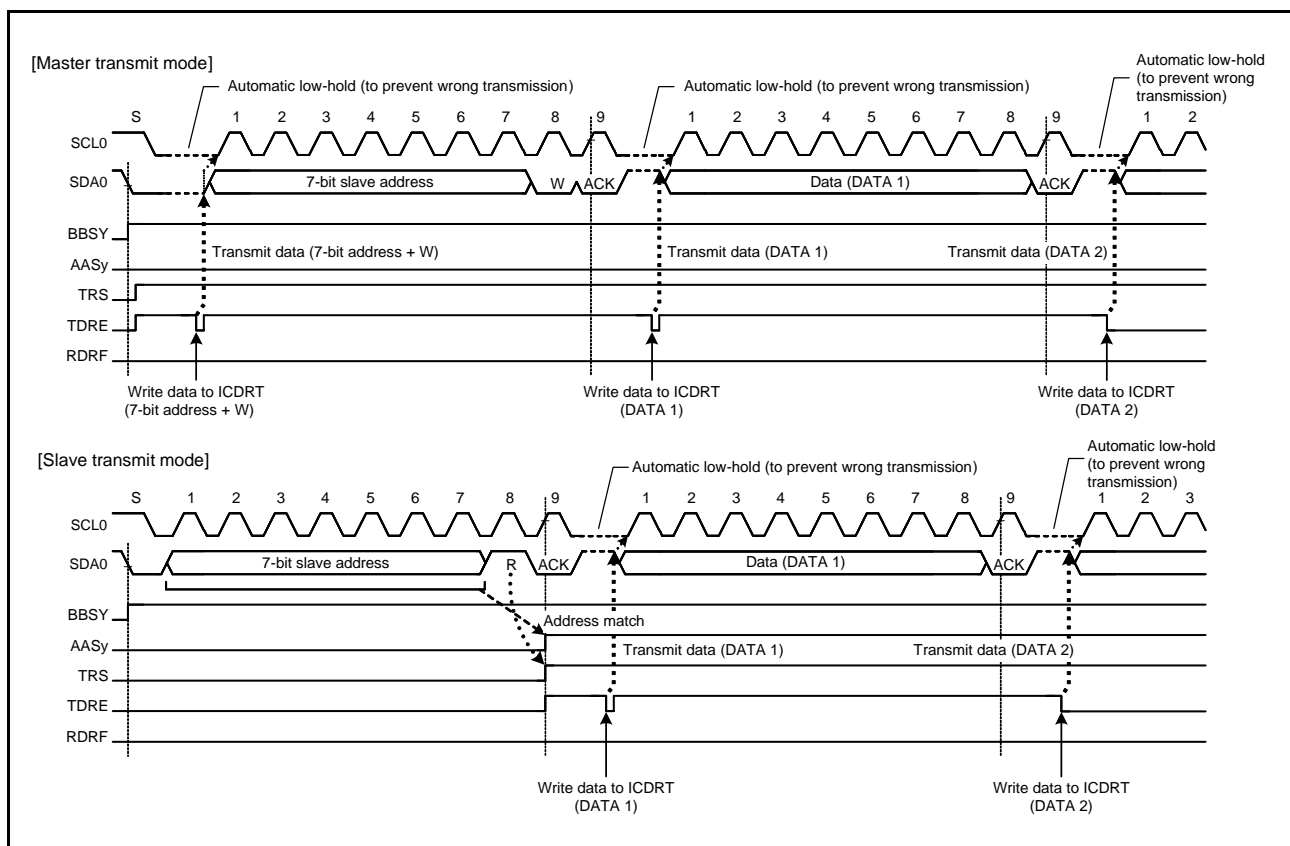


Figure 29.30 Automatic Low-Hold Operation in Transmit Mode

### 29.8.2 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 NACKE 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 SDA0 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 set the NACKF flag to 0. In master transmit mode, after setting the NACKF flag to 0, issue a restart condition, or issue a stop condition and then issue a start condition again.

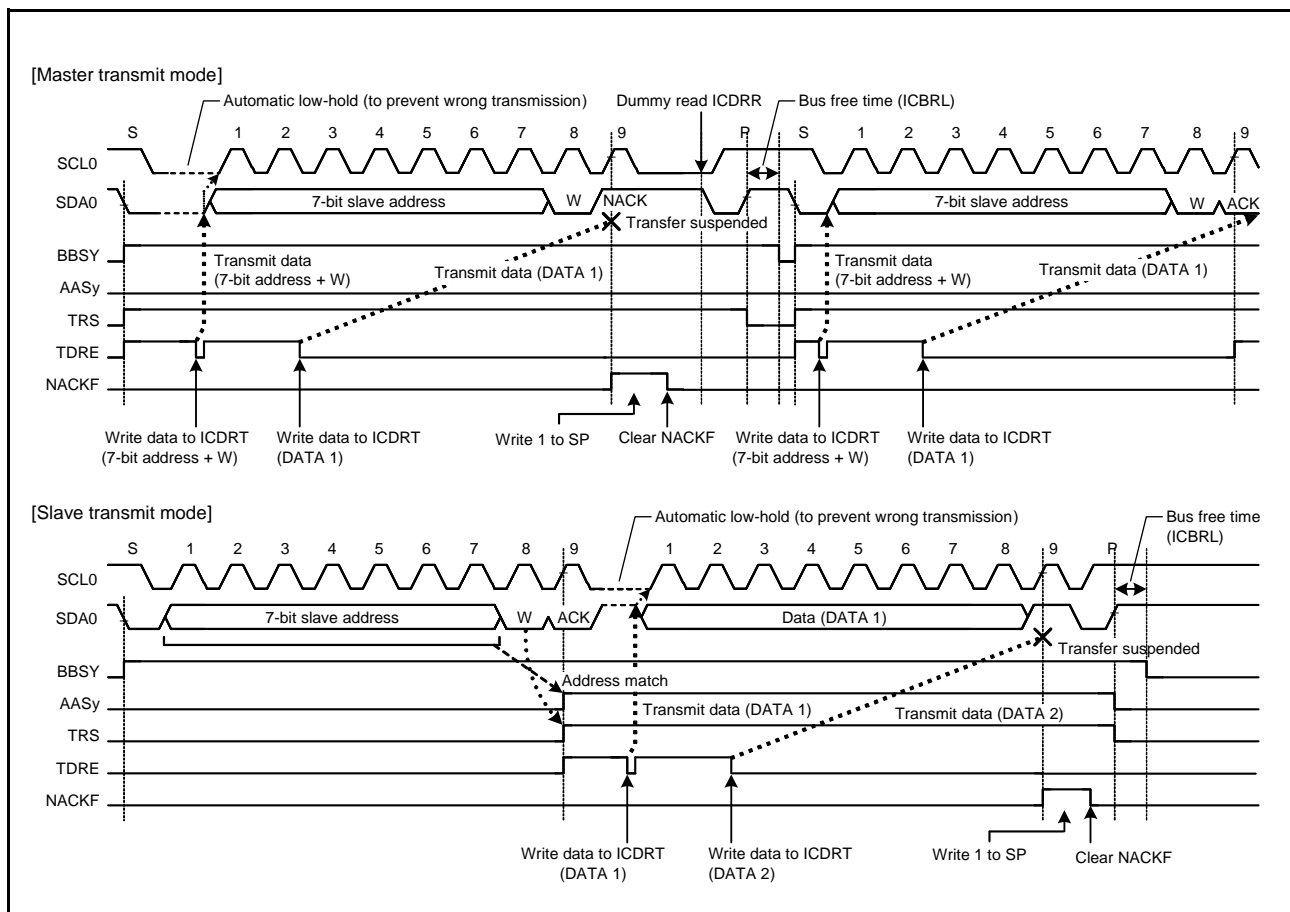


Figure 29.31 Suspension of Data Transfer When NACK is Received (NACKE = 1)

### 29.8.3 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 SCL0 line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is issued. This function does not disturb other communication because the RIIC does not hold the SCL0 line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCL0 line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

(1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the WAIT bit in ICMR3 is set to 1, the RIIC performs 1-byte receive operation using the WAIT bit function. Furthermore, when the ICMR3.RDRFS bit is 0, the RIIC automatically sends the 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 SCL0 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 bitwise 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) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the RDRFS bit in ICMR3 is set to 1, the RIIC performs 1-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 SCL clock cycle, and the SCL0 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.

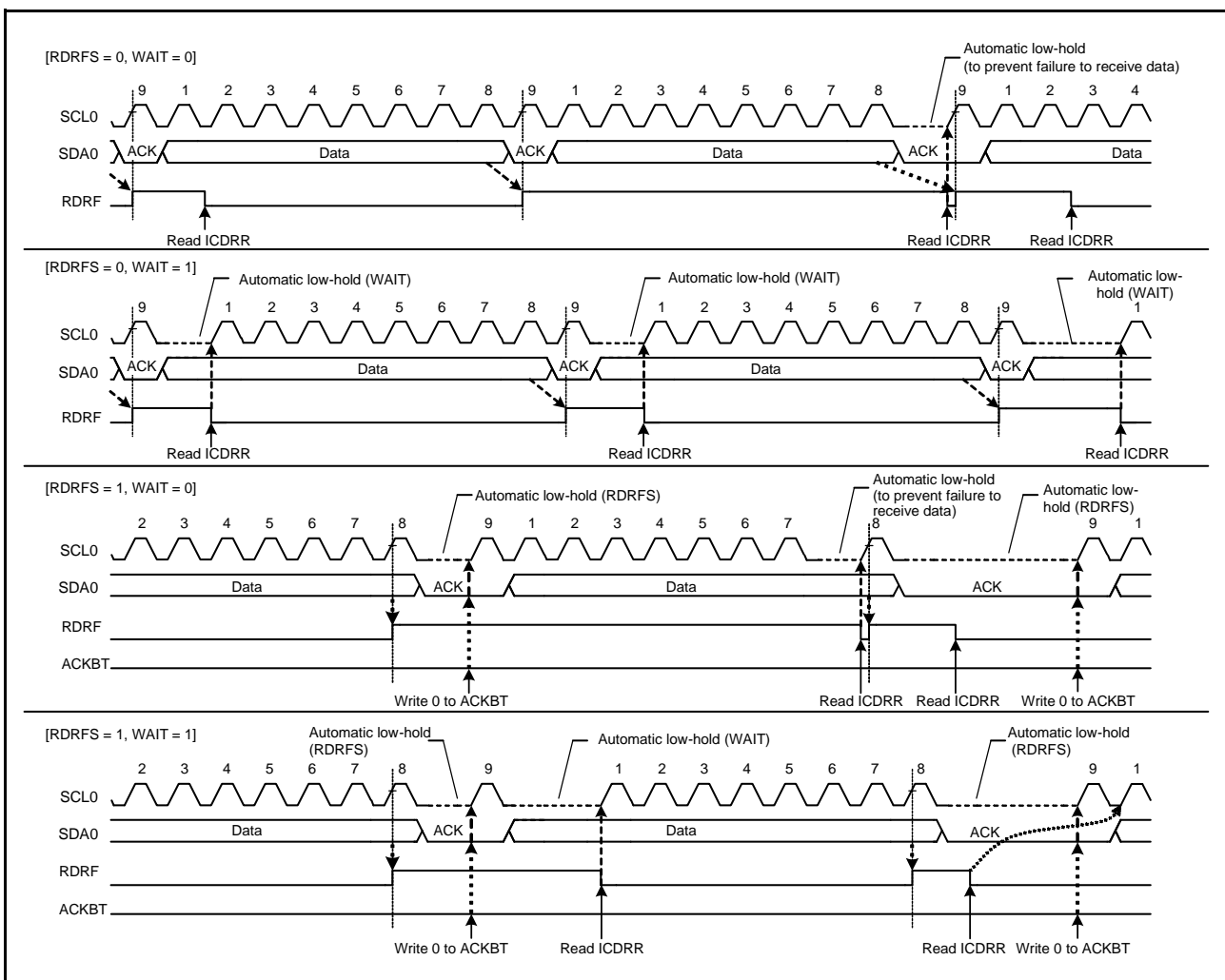


Figure 29.32 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

## 29.9 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.

### 29.9.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA0 line low to issue a start condition. However, if the SDA0 line has already been driven low by another master device issuing a start condition, the RIIC causes arbitration to be lost, so priority is given to transfer by the other master device. Similarly, if the ICCR2.ST bit is set to 1 while the ICCR2.BBSY flag is 1 (bus busy state), arbitration is lost, so priority is given to transfer by the other master device. No start condition is issued in this case.

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 SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low level is detected on the SDA0 line, the RIIC loses in arbitration.

After a loss in arbitration of mastership, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the 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 SDA0 line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 was set to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA0 line in master transmit mode (MST and TRS bits = 11b in ICCR2)

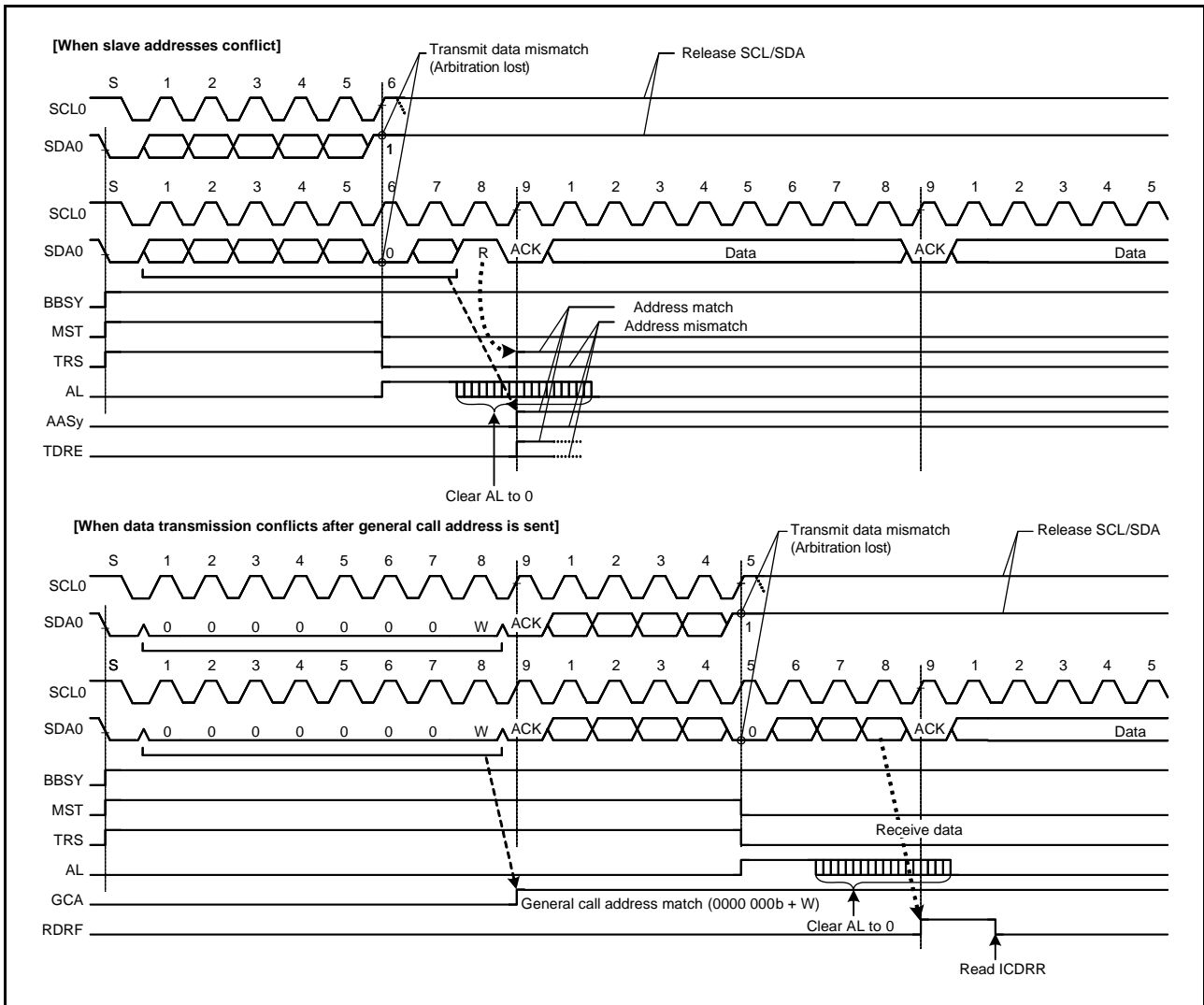


Figure 29.33 Examples of Master Arbitration-Lost Detection (MALE = 1)

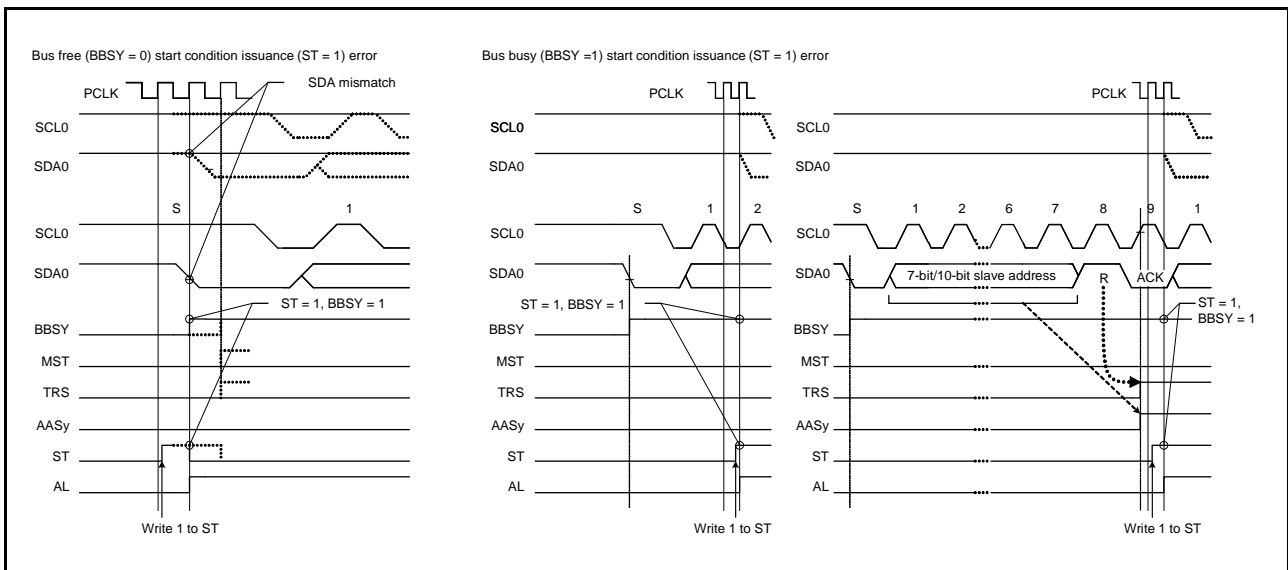
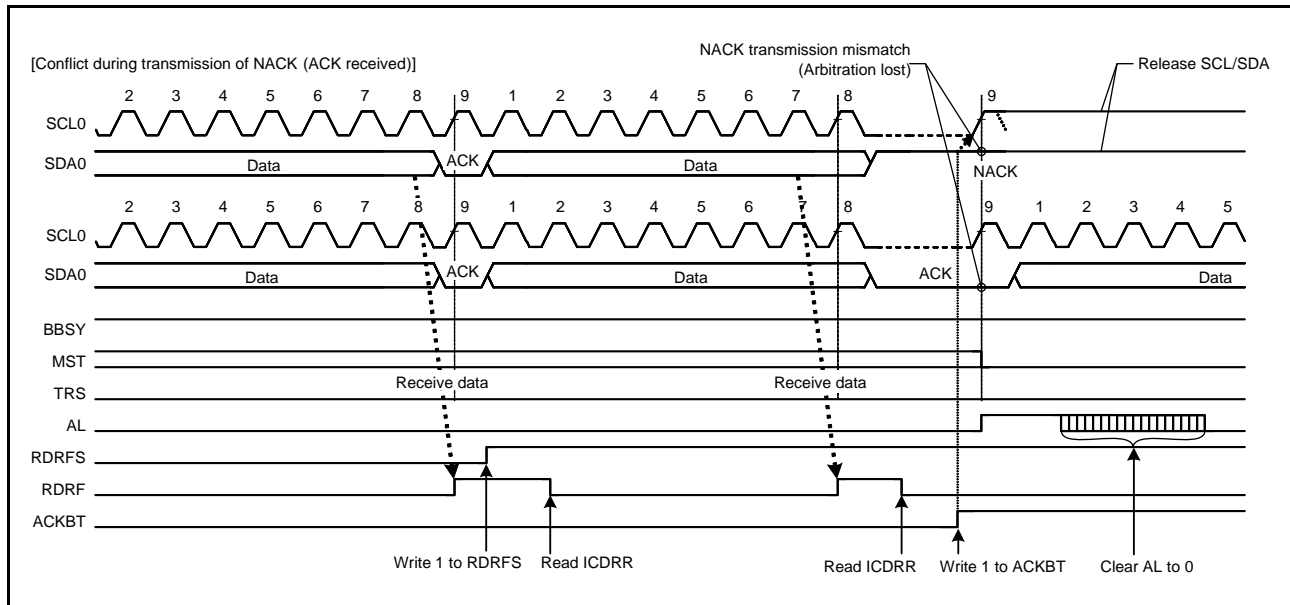


Figure 29.34 Arbitration-Lost When a Start Condition is Issued (MALE = 1)

### 29.9.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA0 line (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low level is detected on the SDA0 line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. Figure 29.35 shows an example of arbitration-lost detection during transmission of NACK.



**Figure 29.35 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)**

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and 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 SDA0 line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3)

### 29.9.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state and the low level is detected on the SDA0 line in slave transmit mode). This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

When it loses slave arbitration, the RIIC is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminate subsequent redundant processing (processing for the transmission of FFh).

The RIIC detects slave arbitration-lost when the following condition is met with the 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 SDA0 line in slave transmit mode (MST and TRS bits = 01b in ICCR2)

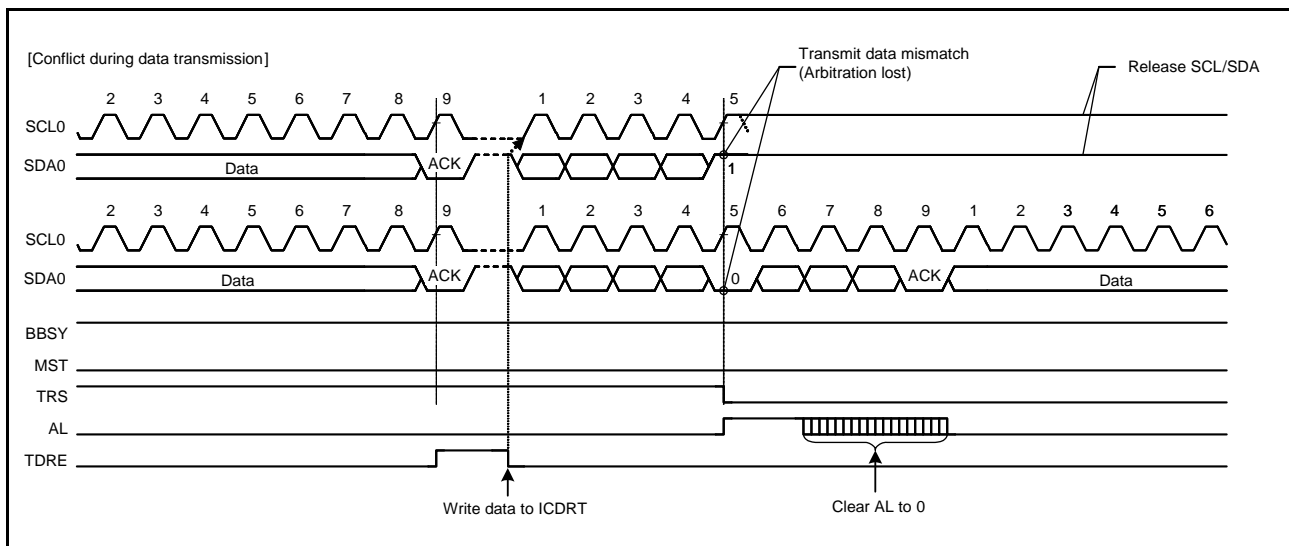


Figure 29.36 Example of Slave Arbitration-Lost Detection (SALE = 1)

## 29.10 Start Condition/Restart Condition/Stop Condition Issuing Function

### 29.10.1 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 state). 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 SDA0 line low (high level to low level).
- (2) Ensure the time set in ICBRH and the start condition hold time.
- (3) Drive the SCL0 line low (high level to low level).
- (4) Detect low level of the SCL0 line and ensure the low-level period of SCL0 line set in ICBRL.

### 29.10.2 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 state) 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 SDA0 line.
- (2) Ensure the low-level period of SCL0 line set in ICBRL.
- (3) Release the SCL0 line (low level to high level).
- (4) Detect a high level of the SCL0 line and ensure the time set in ICBRL and the restart condition setup time.
- (5) Drive the SDA0 line low (high level to low level).
- (6) Ensure the time set in ICBRH and the restart condition hold time.
- (7) Drive the SCL0 line low (high level to low level).
- (8) Detect a low level of the SCL0 line and ensure the low-level period of SCL0 line set in ICBRL.

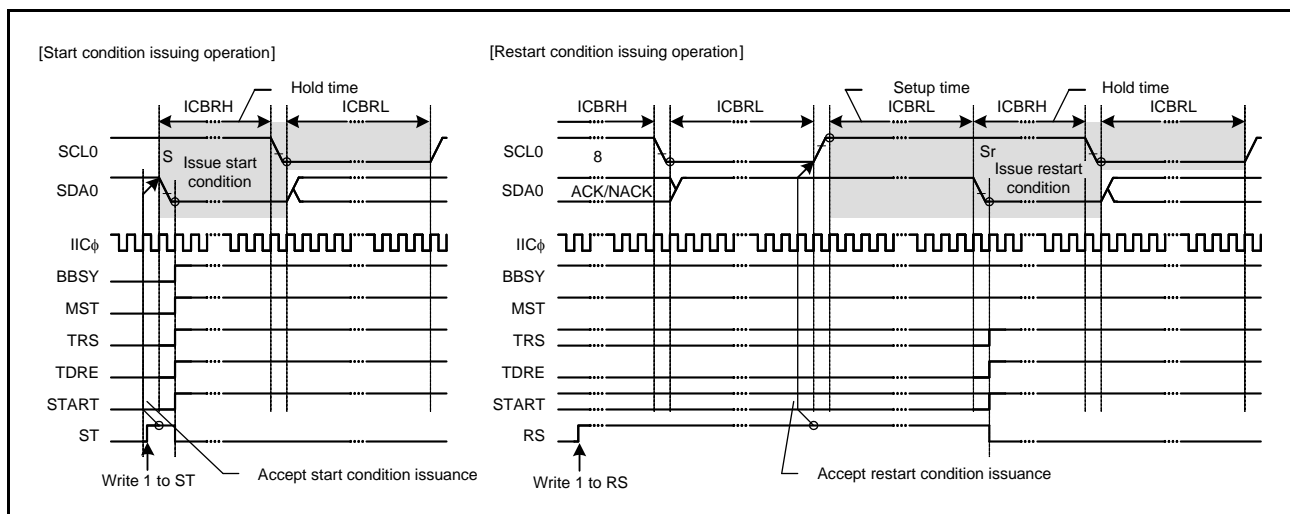


Figure 29.37 Start Condition/Restart Condition Issue Timing (ST and RS Bits)



### 29.10.3 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 state) and the MST bit in ICCR2 is 1 (master mode).

A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDA0 line low (high level to low level).
- Ensure the low-level period of SCL0 line set in ICBRL.
- Release the SCL0 line (low level to high level).
- Detect a high level of the SCL0 line and ensure the time set in ICBRH and the stop condition setup time.
- Release the SDA0 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).

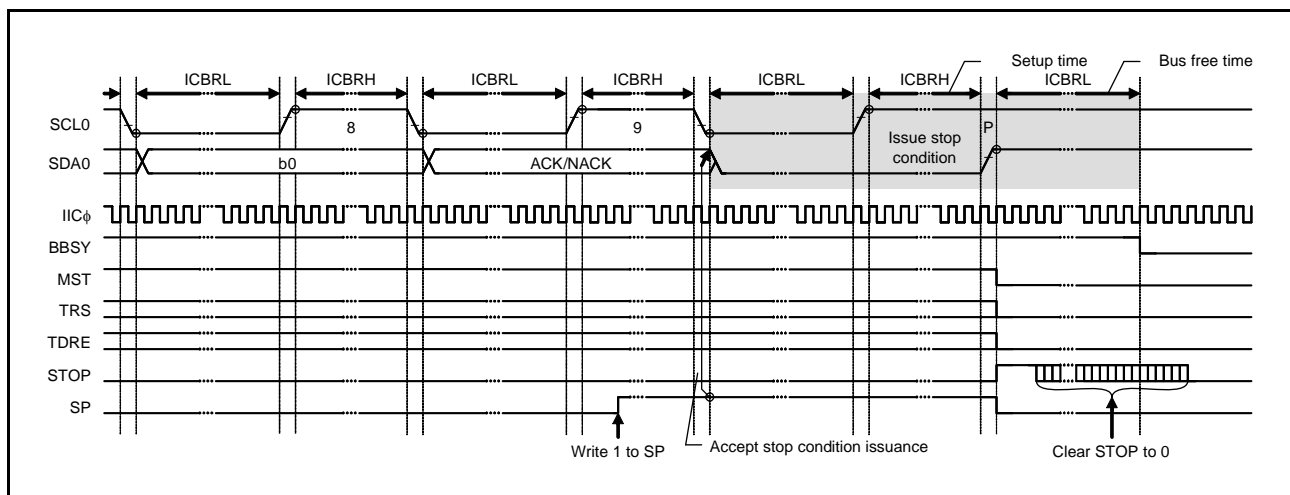


Figure 29.38 Stop Condition Issue Timing (SP Bit)

## 29.11 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 SCL0 line and/or SDA0 line.

As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL0 line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, the RIIC reset function, and internal reset function.

By checking 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 SCL0 or SDA0 lines.

### 29.11.1 Timeout Function

The RIIC includes a timeout function for detecting when the SCL0 line has been stuck longer than the predetermined time. The RIIC can detect an abnormal bus state by monitoring that the SCL0 line is stuck low or high for a predetermined time.

The timeout function monitors the SCL0 line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL0 line changes (rising or falling), but continues to count unless the SCL0 line changes. If the internal counter overflows due to no SCL0 line change, the RIIC can detect the timeout and report the bus hung state.

This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state that the SCL0 line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1).
- The RIIC's own slave address is detected (ICSR1 register is not 00h) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0).
- The bus is free (ICCR2.BBSY flag is 0) while generation of a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IIC $\phi$ ) 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 SCL0 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 set to 0, the internal counter does not work.

Note: When using the timeout detection function, refer to section 29.2.4, I<sup>2</sup>C Bus Mode Register 2 (ICMR2), section 29.2.18, Timeout Internal Counter (TMOCNT), and section 29.3.2, Initial Settings.

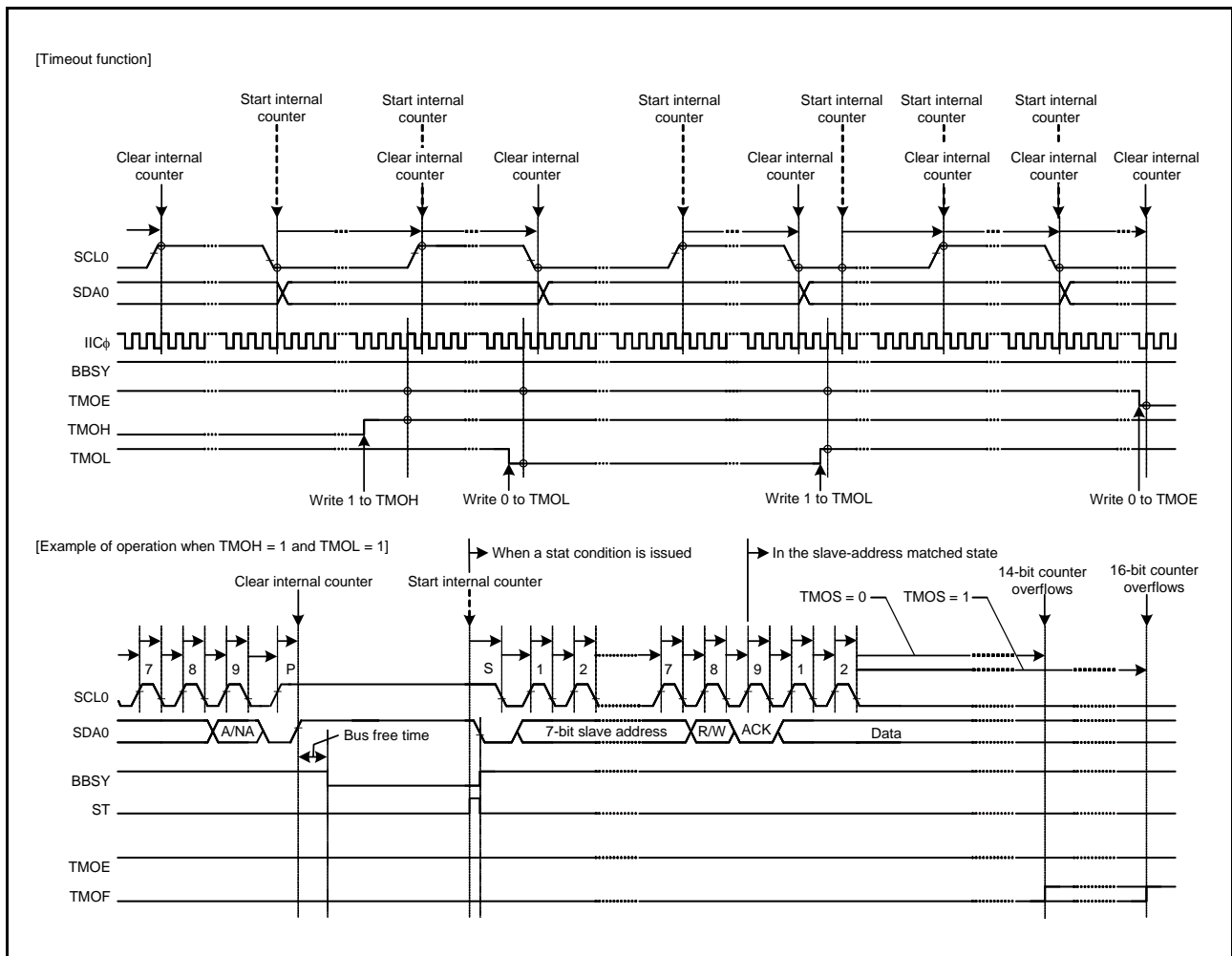


Figure 29.39 Timeout Function (ICMR1.CKS[2:0] = 000b)

### 29.11.2 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 SDA0 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 SDA0 line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from the RIIC with single cycles of the SCL clock as the unit if the RIIC cannot issue a stop condition because the slave device is holding the SDA0 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 set 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 SDA0 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 can be used to output extra cycles of SCL one by one to make the slave device release the SDA0 line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDA0 line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming release of the SDA0 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 set 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 SDA0 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 SCL0 line low

Figure 29.40 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

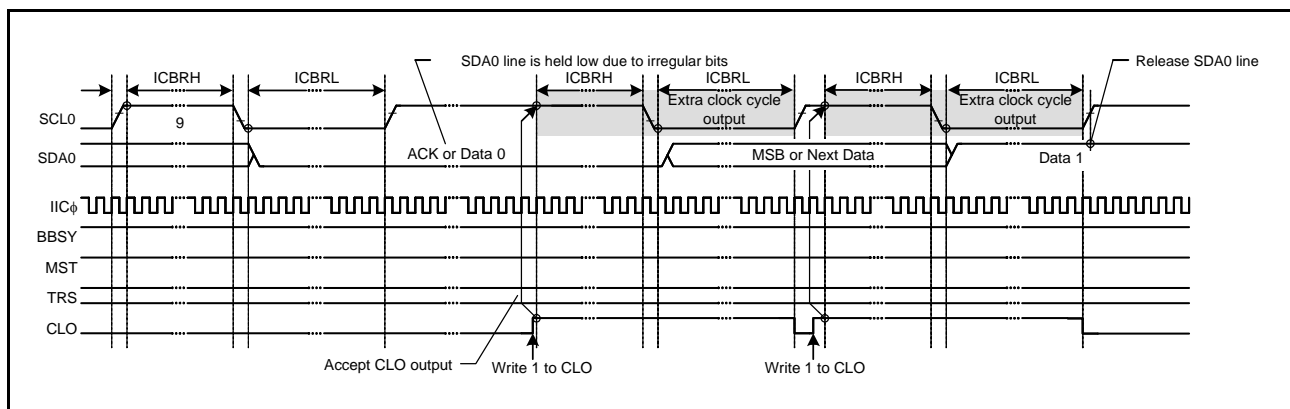


Figure 29.40 Extra SCL Clock Cycle Output Function (CLO Bit)

### 29.11.3 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 set 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 SCL0 and SDA0 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, refer to section 29.14, Resets and Register and Function States When Issuing Each Condition.

## 29.12 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, 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.

### 29.12.1 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 SCL0 pin and SDA0 pin and make the SCL0/SDA0 pin outputs high-impedance, which releases the bus.

#### (2) Measuring timeout of master device

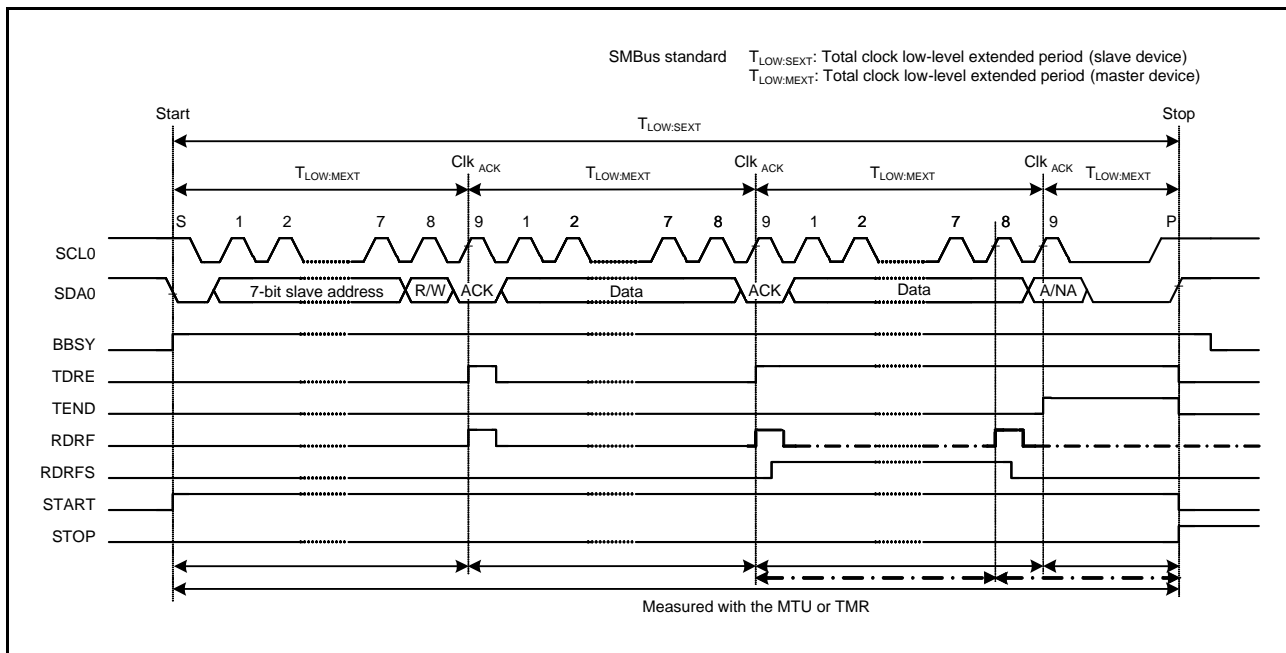
The following periods (timeout interval:  $T_{\text{LOW:MEXT}}$ ) must be measured for master devices in SMBus communication.

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition

To measure timeout for master devices, measure these periods with the MTU or TMR timer using a start condition detection interrupt (STI), stop condition detection interrupt (SPI), and transmit end interrupt (TEI) or receive data full interrupt (RXI) of the RIIC. The measured timeout period must be within the total clock low-level extended period (master device)  $T_{\text{LOW:MEXT}}$ : 10 ms (max.) of the SMBus standard, and the total of all  $T_{\text{LOW:MEXT}}$  from start condition to stop condition must be within  $T_{\text{LOW:SEXT}}$ : 25 ms (max.).

For the ACK receive timing (rising edge of the ninth SCL clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). 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 SCL clock cycle.

If the period measured with the MTU or TMR exceeds the total clock low-level extended period (master device)  $T_{\text{LOW:MEXT}}$ : 10 ms (max.) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout  $T_{\text{TIMEOUT}}$ : 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 29.41 SMBus Timeout Measurement**

### 29.12.2 Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of the RIIC. For the CRC generating polynomials of the CRC calculator, refer to section 32, CRC Calculator (CRC).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the RDRFS bit in ICMR3 to 1 before the rising edge of the eighth SCL clock cycle during reception of the final byte, and hold the SCL0 line low at the falling edge of the eighth clock cycle.

### 29.12.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address or to request its own slave address from the SMBus host.

For a product of this MCU to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so the RIIC has a function for detecting the host address. To detect the host address as a slave address, set the SMBS bit in ICMR3 and the HOAE bit in IC SER to 1. Operation after the host address has been detected is the same as normal slave operation.

## 29.13 Interrupt Sources

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 data full, transmit data empty, and transmit end.

Table 29.7 lists details of the several interrupt requests. The receive data full and transmit data empty are both capable of activating data transfer by the DTC.

**Table 29.7 Interrupt Sources**

Symbol	Interrupt Source	Interrupt Flag	DTC Activation	Priority	Interrupt Condition
EEI	Transfer error/ event generation	AL	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
RXI*2	Receive data full	RDRF	Possible		RDRF = 1 • RIE = 1
TXI*1	Transmit data empty	TDRE	Possible		TDRE = 1 • TIE = 1
TEI*3	Transmit end	TEND	Not possible	Low	TEND = 1 • TEIE = 1

Note: There is a delay time between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Since TXI is an edge-detected interrupt, it does not require clearing. Furthermore, the TDRE flag in ICSR2 (a condition for TXI) is automatically set to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).

Note 2. Since RXI is an edge-detected interrupt, it does not require clearing. Furthermore, the RDRF flag in ICSR2 (a condition for RXI) is automatically set to 0 when data are read from ICDRR.

Note 3. When using the TEI interrupt, clear the TEND flag in ICSR2 in the TEI interrupt handling.  
Note that the TEND flag in ICSR2 is automatically set to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).

Clear or mask the each flag during interrupt handling.

### 29.13.1 Buffer Operation for TXI and RXI Interrupts

If the conditions for generating a TXI and RXI interrupt are satisfied while the corresponding IR flag is 1, the interrupt request is not output for the ICU but retained internally (the capacity for internal retention is one request per source).

An interrupt request that was being retained within the ICU is output when the value of the ICU.IRn.IR flag becomes 0.

Internally retained interrupt requests are automatically cleared under normal conditions of usage.

Internally retained interrupt requests can also be cleared by writing 0 to the interrupt enable bit within the given peripheral module.



## 29.14 Resets and Register and Function States When Issuing Each Condition

The RIIC has reset, RIIC reset, and internal reset functions. Table 29.8 lists the resets and register and function states when issuing each condition.

**Table 29.8 Resets and Register and Function States When Issuing Each Condition**

		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	Retained	Retained	
	SCLO, SDAO		At a reset	At a reset			
	Others			Retained			
ICCR2	BBSY	At a reset	At a reset	Retained	Retained	Retained	
	ST			At a reset			At a reset
	Others						At a reset
ICMR1	BC[2:0]	At a reset	At a reset	At a reset	At a reset	Retained	
	Others			Retained			Retained
ICMR2		At a reset	At a reset	Retained	Retained	Retained	
ICMR3		At a reset	At a reset	Retained	Retained	Retained	
ICFER		At a reset	At a reset	Retained	Retained	Retained	
ICSER		At a reset	At a reset	Retained	Retained	Retained	
ICIER		At a reset	At a reset	Retained	Retained	Retained	
ICSR1		At a reset	At a reset	At a reset	Retained	At a reset	
ICSR2	TDRE, TEND	At a reset	At a reset	At a reset	Retained	At a reset	
	START				Retained		
	STOP				Retained	Retained	
	Others				Retained	Retained	
SARL0, SARL1, SARL2 SARU0, SARU1, SARU2		At a reset	At a reset	Retained	Retained	Retained	
ICBRH, ICBRL		At a reset	At a reset	Retained	Retained	Retained	
ICDRT		At a reset	At a reset	Retained	Retained	Retained	
ICDRR		At a reset	At a reset	Retained	Retained	Retained	
ICDRS		At a reset	At a reset	At a reset	Retained	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	

## 29.15 Event Link Output

The RIIC0 handles event output for the event link controller (ELC) corresponding to the following sources.

(1) Transfer error event

When a transfer error event occurs, the corresponding event signal can be output for another module via the ELC.

(2) Receive data full

When a receive data register becomes full, the corresponding event signal can be output for another module via the ELC.

(3) Transmit data empty

When a transmit data register becomes empty, the corresponding event signal can be output for another module via the ELC.

(4) Transmit end

On completion of transfer, the corresponding event signal can be output for another module via the ELC.

### 29.15.1 Interrupt Handling and Event Linking

The RIIC module produces four kinds of interrupt: transfer error (arbitration-lost detection, detection of NACK, detection of timeout, or detection of a stop condition) event, receive data full, transmit data empty, and transmit end interrupts detection of a start condition. Each of these has an enable bit to control enabling and disabling of the interrupt signal. An interrupt request signal is output for the CPU when an interrupt source condition is satisfied while the setting of the corresponding enable bit is enabled.

The corresponding event link output signals are sent to other modules as event signals via the ELC when the interrupt source conditions are satisfied, regardless of the settings of the interrupt enable bits.

For details on interrupt sources, see Table 29.7.

## 29.16 Usage Notes

### 29.16.1 Setting Module Stop Function

Module stop state can be entered or released using module stop control register B (MSTPCRB). The initial setting is for operation of the RIIC to be stopped. RIIC register access is enabled by releasing the module stop state.

For details on module stop control register B, refer to section 11, Low Power Consumption.

### 29.16.2 Notes on Starting Transfer

If the IR flag corresponding to the RIIC interrupt is 1 when transfer is started (ICCR1.ICE bit = 1), follow the procedure below to clear interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally retained after transfer starts, and this can lead to unanticipated behavior of the IR flag.

1. Confirm that the ICCR1.ICE bit is 0.
2. Set the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side to 0.
3. Read the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side and confirm that its value is 0.
4. Set the IR flag to 0.

## 30. Serial Sound Interface (SSI)

This MCU integrates one channel of the serial sound interface (SSI) compliant to the I<sup>2</sup>S bus standard. The SSI supports I<sup>2</sup>S bus compatibility and MSB-first and left-aligned/right-aligned formats, so it can be used to send or receive audio data with various devices.

### 30.1 Overview

**Table 30.1 SSI Specifications**

Item	Specifications
Number of channels	One channel (SSI0)
Operating mode	Non-compressed mode
Transfer formats	<ul style="list-style-type: none"> <li>I<sup>2</sup>S format supported</li> <li>MSB-first supported</li> <li>Left-aligned/right-aligned formats selectable</li> </ul>
Function	<ul style="list-style-type: none"> <li>Serves as both a transmitter and a receiver</li> <li>Channel 0 supports full-duplex communications.</li> <li>Capable of various audio formats</li> <li>SSISCK (serial bit clock) can be selected from among 16, 32, 48, and 64 fs (fs: Sampling rate)</li> <li>The master clock can be selected from either of the following:               <ul style="list-style-type: none"> <li>Master clock pin for audio (AUDIO_MCLK): 1 to 25 MHz</li> <li>Main clock</li> </ul> </li> <li>Includes 8-stage FIFO buffers in transmitter and receiver</li> <li>Capable of selecting whether to stop word select (SSIWS) or not when data transfer is stopped</li> </ul>
Interrupt sources	Three sources <ul style="list-style-type: none"> <li>Communication error               <ul style="list-style-type: none"> <li>Transmit underflow, transmit overflow, receive underflow, receive overflow, and idle</li> </ul> </li> <li>Receive data full</li> <li>Transmit data empty</li> </ul>
Low power consumption function	Module stop state can be set.

Figure 30.1 shows a block diagram of SSI (SSIO).

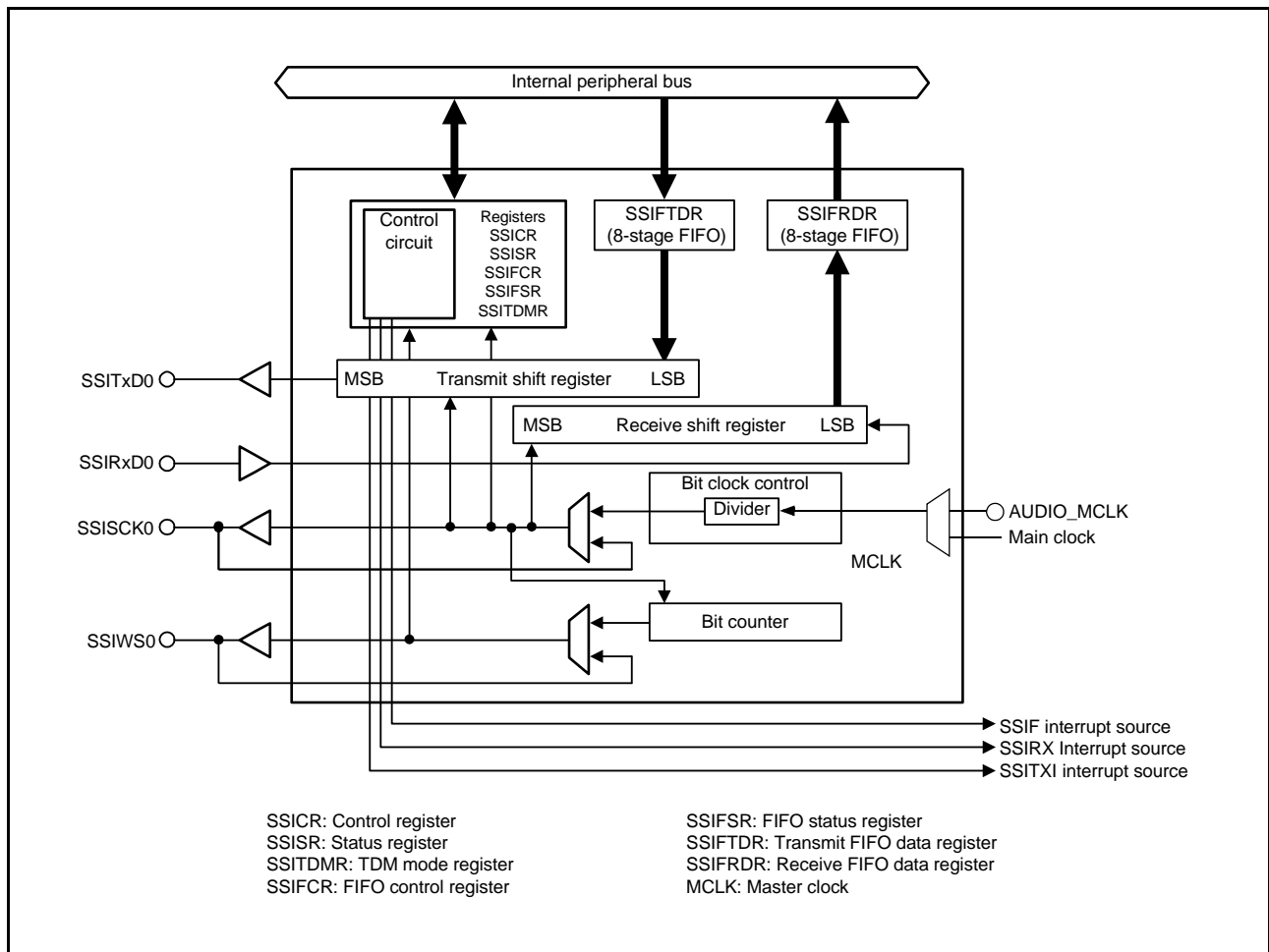


Figure 30.1 Block Diagram of SSI (SSIO)

Table 30.2 lists the I/O pins of the SSI.

Table 30.2 SSI I/O Pins

Channel	Pin Name	I/O	Description
SSIO	SSISCK0	I/O	Serial bit clock pin
	SSIWS0	I/O	Word selection pin
	SSITXD0	Output	Serial data output pin
	SSIRXD0	Input	Serial data input pin
	AUDIO_MCLK	Input	Master clock for audio pin (input master clock)

## 30.2 Register Descriptions

### 30.2.1 Control Register (SSICR)

Address(es): SSI0.SSICR 0008 A500h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IEN	—	CHNL[1:0]	DWL[2:0]			SWL[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
SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]			MUEN	—	TEN	REN	
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	REN	Receive Enable	0: Disables receive operation. 1: Enables receive operation.	R/W																																																																						
b1	TEN	Transmit Enable	0: Disables transmit operation. 1: Enables transmit operation.	R/W																																																																						
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W																																																																						
b3	MUEN	Mute Enable*1	0: Not muted. 1: Muted.	R/W																																																																						
b7 to b4	CKDV[3:0]	Serial Bit Clock Frequency Setting*3	<table border="0"> <tr> <td>b7</td> <td>b6</td> <td>b5</td> <td>b4</td> <td></td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0: MCLK</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>1: MCLK/2</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>0: MCLK/4</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>1: MCLK/8</td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>0</td><td>0: MCLK/16</td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>1</td><td>1: MCLK/32</td> </tr> <tr> <td>0</td><td>1</td><td>1</td><td>0</td><td>0: MCLK/64</td> </tr> <tr> <td>0</td><td>1</td><td>1</td><td>1</td><td>1: MCLK/128</td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>0</td><td>0: MCLK/6</td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>1</td><td>1: MCLK/12</td> </tr> <tr> <td>1</td><td>0</td><td>1</td><td>0</td><td>0: MCLK/24</td> </tr> <tr> <td>1</td><td>0</td><td>1</td><td>1</td><td>1: MCLK/48</td> </tr> <tr> <td>1</td><td>1</td><td>0</td><td>0</td><td>0: MCLK/96</td> </tr> </table> Settings other than above are prohibited.	b7	b6	b5	b4		0	0	0	0	0: MCLK	0	0	0	1	1: MCLK/2	0	0	1	0	0: MCLK/4	0	0	1	1	1: MCLK/8	0	1	0	0	0: MCLK/16	0	1	0	1	1: MCLK/32	0	1	1	0	0: MCLK/64	0	1	1	1	1: MCLK/128	1	0	0	0	0: MCLK/6	1	0	0	1	1: MCLK/12	1	0	1	0	0: MCLK/24	1	0	1	1	1: MCLK/48	1	1	0	0	0: MCLK/96	R/W
b7	b6	b5	b4																																																																							
0	0	0	0	0: MCLK																																																																						
0	0	0	1	1: MCLK/2																																																																						
0	0	1	0	0: MCLK/4																																																																						
0	0	1	1	1: MCLK/8																																																																						
0	1	0	0	0: MCLK/16																																																																						
0	1	0	1	1: MCLK/32																																																																						
0	1	1	0	0: MCLK/64																																																																						
0	1	1	1	1: MCLK/128																																																																						
1	0	0	0	0: MCLK/6																																																																						
1	0	0	1	1: MCLK/12																																																																						
1	0	1	0	0: MCLK/24																																																																						
1	0	1	1	1: MCLK/48																																																																						
1	1	0	0	0: MCLK/96																																																																						
b8	DEL	Serial Data Delay*3	0: I <sup>2</sup> S format compatibility One clock cycle delay between SSIWS and SSIDATA 1: MSB-first left-aligned/right-aligned format compatibility No delay between SSIWS and SSIDATA	R/W																																																																						
b9	PDTA	Parallel Data Allocation*3	When data word length is 8 or 16 bits: 0: The lower bits of parallel data (SSIFTDR, SSIFRDR) are transferred prior to the upper bits. 1: The upper bits of parallel data (SSIFTDR, SSIFRDR) are transferred prior to the lower bits.  When data word length is 18, 20, 22, or 24 bits: 0: Parallel data (SSIFTDR, SSIFRDR) is left-aligned. 1: Parallel data (SSIFTDR, SSIFRDR) is right-aligned.	R/W																																																																						
b10	SDTA	Serial Data Alignment*3	0: Transmitting and receiving in the order of serial data and padding bits 1: Transmitting and receiving in the order of padding bits and serial data	R/W																																																																						
b11	SPDP	Serial Padding Polarity*3	0: Padding data is 0. 1: Padding data is 1.	R/W																																																																						

Bit	Symbol	Bit Name	Description	R/W															
b12	SWSP	Word Select Polarity	0: SSIWS is low for the 1st system word, high for the 2nd system word. 1: SSIWS is high for the 1st system word, low for the 2nd system word.	R/W															
b13	SCKP	Serial Bit Clock Polarity*3	0: SSIWS and SSIDATA change at the SSISCK falling edge (sampled at the SCK rising edge). 1: SSIWS and SSIDATA change at the SSISCK rising edge (sampled at the SCK falling edge).	R/W															
<table border="1"> <thead> <tr> <th></th> <th>SCKP bit = 0</th> <th>SCKP bit = 1</th> </tr> </thead> <tbody> <tr> <td>SSIDATA input sampling timing for reception</td> <td>SSISCK rising edge</td> <td>SSISCK falling edge</td> </tr> <tr> <td>SSIDATA output changing timing for transmission</td> <td>SSISCK falling edge</td> <td>SSISCK rising edge</td> </tr> <tr> <td>SSIWS input sampling timing in slave mode (SWSD bit = 0)</td> <td>SSISCK rising edge</td> <td>SSISCK falling edge</td> </tr> <tr> <td>SSIWS output changing timing in master mode (SWSD bit = 1)</td> <td>SSISCK falling edge</td> <td>SSISCK rising edge</td> </tr> </tbody> </table>						SCKP bit = 0	SCKP bit = 1	SSIDATA input sampling timing for reception	SSISCK rising edge	SSISCK falling edge	SSIDATA output changing timing for transmission	SSISCK falling edge	SSISCK rising edge	SSIWS input sampling timing in slave mode (SWSD bit = 0)	SSISCK rising edge	SSISCK falling edge	SSIWS output changing timing in master mode (SWSD bit = 1)	SSISCK falling edge	SSISCK rising edge
	SCKP bit = 0	SCKP bit = 1																	
SSIDATA input sampling timing for reception	SSISCK rising edge	SSISCK falling edge																	
SSIDATA output changing timing for transmission	SSISCK falling edge	SSISCK rising edge																	
SSIWS input sampling timing in slave mode (SWSD bit = 0)	SSISCK rising edge	SSISCK falling edge																	
SSIWS output changing timing in master mode (SWSD bit = 1)	SSISCK falling edge	SSISCK rising edge																	
b14	SWSD	Word Select Direction*2, *3	0: SSIWS pin is input (slave mode). 1: SSIWS pin is output (master mode).	R/W															
b15	SCKD	Serial Bit Clock Direction*2, *3	0: SSISCK pin is input (slave mode). 1: SSISCK pin is output (master mode).	R/W															
b18 to b16	SWL[2:0]	System Word Length*3	Set the system word length to the bit clock frequency/2 fs. b18 b16 0 0 0: 8 bits (serial bit clock frequency = 16 fs) 0 0 1: 6 bits (serial bit clock frequency = 32 fs) 0 1 0: 24 bits (serial bit clock frequency = 48 fs) 0 1 1: 32 bits (serial bit clock frequency = 64 fs) Settings other than above are prohibited.	R/W															
b21 to b19	DWL[2:0]	Data Word Length*3	b21 b19 0 0 0: 8 bits 0 0 1: 16 bits 0 1 0: 18 bits 0 1 1: 20 bits 1 0 0: 22 bits 1 0 1: 24 bits Settings other than above are prohibited.	R/W															
b23, b22	CHNL[1:0]	Channels*3	b23 b22 0 0: One channel Settings other than above are prohibited.	R/W															
b24	—	Reserved	This bit is read as 0. The write value should be 0.	R/W															
b25	IEN	Idle Interrupt Enable	0: Disables an idle interrupt. 1: Enables an idle interrupt.	R/W															
b26	ROIEN	Receive FIFO Overflow Interrupt Enable	0: Disables a receive FIFO overflow interrupt. 1: Enables a receive FIFO overflow interrupt.	R/W															
b27	RUIEN	Receive FIFO Underflow Interrupt Enable	0: Disables a receive FIFO underflow interrupt. 1: Enables a receive FIFO underflow interrupt.	R/W															
b28	TOIEN	Transmit FIFO Overflow Interrupt Enable	0: Disables a transmit FIFO overflow interrupt. 1: Enables a transmit FIFO overflow interrupt.	R/W															
b29	TUIEN	Transmit FIFO Underflow Interrupt Enable	0: Disables a transmit FIFO underflow interrupt. 1: Enables a transmit FIFO underflow interrupt.	R/W															
b30	CKS	Audio Clock Select*3	0: AUDIO_MCLK input 1: Main clock	R/W															
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W															

Note 1. While this module is muted, 0 is transmitted regardless of the value of serial data, but data transfer is not stopped. Since the number of data in the transmit FIFO decreases, write dummy data to the SSIFTDR register to prevent the generation of a transmit underflow. When the MUEN bit is set to 1, the SSITXD0 and SSIDATA1 pins are immediately sets to 0 without synchronizing SSIWS.

Note 2. Set the SCKD and SWSD bits to the same value. Other settings are prohibited.

Note 3. Rewriting is allowed only in the idle state.

**REN Bit (Receive Enable)**

This bit enables or disables receive operation. Setting this bit to 1 starts receive operation.

**TEN Bit (Transmit Enable)**

This bit enables or disables transmit operation. Setting this bit to 1 starts transmit operation.

SSITXD0 of SSI0 is set as output while the I/O port function is selected, regardless of the TEN bit setting. SSIDATA1 of SSI1 is set as output when the TEN bit is 1 and input when the TEN bit is 0 while the I/O port function is selected for SSIDATA1.

**Table 30.3 SSITXD0, SSIRXD0, and SSIDATA1 Pin States**

Register Settings			SSI0		SSI1
MPC setting	TEN	REN	SSITXD0	SSIRXD0	SSIDATA1
SSI selected	0	0	Output	Input	Input
	0	1	Output	Input	Input
	1	0	Output	Input	Output
	1	1	Output	Input	—
SSI deselected	x	x	I/O port	I/O port	I/O port

x: Don't care

—: Settings prohibited.

I/O port: Depends on the settings of the I/O port and multi-function pin controller.

**CKDV[3:0] Bits (Serial Bit Clock Frequency Setting)**

These bits select the frequency of the serial bit clock in master mode. Since the input clock from the SSISCK pin is used in slave mode, the setting of these bits is ignored. The serial bit clock is used as the operating clock of the shift register.

Calculation Example:

When  $f_s$  (sampling rate) = the SSIWS frequency = 96 kHz and the system word length = 32 bits

The bit clock frequency =  $96 \text{ kHz} \times 32 \text{ bits} \times 2 = 6.144 \text{ MHz}$  is necessary, so set CKDV[3:0] = 0001b (MCLK/2) when MCLK = 12.288 MHz.

**PDTA Bit (Parallel Data Allocation)**

The setting of this bit specifies the allocation of data to be stored in the SSIFRDR register in receive mode and the SSIFTDR register in transmit mode.

During receive operation, the SSI stores the data received from the serial bus in SSIFRDR according to the PDTA bit setting.

During transmit operation, the SSI stores the data stored in SSIFTDR in the transmit shift register, and transmits the data to the serial bus according to the PDTA bit setting.



When PDTA = 0

DWL[2:0] Bits	SSIFTDR/SSIFRDR[31:0] Registers										
000b	<table border="1"> <tr> <td style="text-align: right;">31</td> <td style="text-align: center;">24 23</td> <td style="text-align: center;">16 15</td> <td style="text-align: center;">8 7</td> <td style="text-align: right;">0</td> </tr> <tr> <td style="text-align: center;">4th word</td> <td style="text-align: center;">3rd word</td> <td style="text-align: center;">2nd word</td> <td style="text-align: center;">1st word</td> <td></td> </tr> </table>	31	24 23	16 15	8 7	0	4th word	3rd word	2nd word	1st word	
31	24 23	16 15	8 7	0							
4th word	3rd word	2nd word	1st word								
001b	<table border="1"> <tr> <td style="text-align: right;">31</td> <td style="text-align: center;">16 15</td> <td style="text-align: right;">0</td> </tr> <tr> <td style="text-align: center;">2nd word</td> <td style="text-align: center;">1st word</td> <td></td> </tr> </table>	31	16 15	0	2nd word	1st word					
31	16 15	0									
2nd word	1st word										
010b	<table border="1"> <tr> <td style="text-align: right;">31</td> <td style="text-align: center;">14 13</td> <td style="text-align: right;">0</td> </tr> <tr> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Invalid</td> <td></td> </tr> </table>	31	14 13	0	Valid	Invalid					
31	14 13	0									
Valid	Invalid										
011b	<table border="1"> <tr> <td style="text-align: right;">31</td> <td style="text-align: center;">12 11</td> <td style="text-align: right;">0</td> </tr> <tr> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Invalid</td> <td></td> </tr> </table>	31	12 11	0	Valid	Invalid					
31	12 11	0									
Valid	Invalid										
100b	<table border="1"> <tr> <td style="text-align: right;">31</td> <td style="text-align: center;">10 9</td> <td style="text-align: right;">0</td> </tr> <tr> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Invalid</td> <td></td> </tr> </table>	31	10 9	0	Valid	Invalid					
31	10 9	0									
Valid	Invalid										
101b	<table border="1"> <tr> <td style="text-align: right;">31</td> <td style="text-align: center;">8 7</td> <td style="text-align: right;">0</td> </tr> <tr> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Invalid</td> <td></td> </tr> </table>	31	8 7	0	Valid	Invalid					
31	8 7	0									
Valid	Invalid										

When PDTA = 1

DWL[2:0] Bits	SSIFTDR/SSIFRDR[31:0] Registers										
000b	<table border="1"> <tr> <td style="text-align: right;">31</td> <td style="text-align: center;">24 23</td> <td style="text-align: center;">16 15</td> <td style="text-align: center;">8 7</td> <td style="text-align: right;">0</td> </tr> <tr> <td style="text-align: center;">1st word</td> <td style="text-align: center;">2nd word</td> <td style="text-align: center;">3rd word</td> <td style="text-align: center;">4th word</td> <td></td> </tr> </table>	31	24 23	16 15	8 7	0	1st word	2nd word	3rd word	4th word	
31	24 23	16 15	8 7	0							
1st word	2nd word	3rd word	4th word								
001b	<table border="1"> <tr> <td style="text-align: right;">31</td> <td style="text-align: center;">16 15</td> <td style="text-align: right;">0</td> </tr> <tr> <td style="text-align: center;">1st word</td> <td style="text-align: center;">2nd word</td> <td></td> </tr> </table>	31	16 15	0	1st word	2nd word					
31	16 15	0									
1st word	2nd word										
010b	<table border="1"> <tr> <td style="text-align: right;">31</td> <td style="text-align: center;">18 17</td> <td style="text-align: right;">0</td> </tr> <tr> <td style="text-align: center;">Invalid</td> <td style="text-align: center;">Valid</td> <td></td> </tr> </table>	31	18 17	0	Invalid	Valid					
31	18 17	0									
Invalid	Valid										
011b	<table border="1"> <tr> <td style="text-align: right;">31</td> <td style="text-align: center;">20 19</td> <td style="text-align: right;">0</td> </tr> <tr> <td style="text-align: center;">Invalid</td> <td style="text-align: center;">Valid</td> <td></td> </tr> </table>	31	20 19	0	Invalid	Valid					
31	20 19	0									
Invalid	Valid										
100b	<table border="1"> <tr> <td style="text-align: right;">31</td> <td style="text-align: center;">22 21</td> <td style="text-align: right;">0</td> </tr> <tr> <td style="text-align: center;">Invalid</td> <td style="text-align: center;">Valid</td> <td></td> </tr> </table>	31	22 21	0	Invalid	Valid					
31	22 21	0									
Invalid	Valid										
101b	<table border="1"> <tr> <td style="text-align: right;">31</td> <td style="text-align: center;">24 23</td> <td style="text-align: right;">0</td> </tr> <tr> <td style="text-align: center;">Invalid</td> <td style="text-align: center;">Valid</td> <td></td> </tr> </table>	31	24 23	0	Invalid	Valid					
31	24 23	0									
Invalid	Valid										

**CHNL[1:0] Bits (Channels)**

These bits select the number of channels to be decoded in each system word. Set these bits to 00b in this module.

### 30.2.2 Status Register (SSISR)

Address(es): SSI0.SSISR 0008 A504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	1	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
	—	—	—	—	—	—	—	—	—	TCHNO[1:0]	TSWNO	RCHNO[1:0]	RSWNO	IDST		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	IDST	Idle Status Flag	0: SSI communication is in progress. 1: SSI communication is idle.	R
b1	RSWNO	Receive System Word Number	Receive word number	R
b3, b2	RCHNO[1:0]	Receive Channel Number	These bits are read as 00b.	R
b4	TSWNO	Transmit System Word Number	Transmit word number	R
b6, b5	TCHNO[1:0]	Transmit Channel Number	These bits are read as 00b.	R
b24 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25	IIRQ	Idle Interrupt Status Flag	0: Not in idle state 1: In idle state	R
b26	ROIRQ	Receive Overflow Interrupt Status Flag	0: No receive overflow has occurred. 1: A receive overflow has occurred.	R/(W) *1
b27	RUIRQ	Receive Underflow Interrupt Status Flag	0: No receive underflow has occurred. 1: A receive underflow has occurred.	R/(W) *1
b28	TOIRQ	Transmit Overflow Interrupt Status Flag	0: No transmit overflow has occurred. 1: A transmit overflow has occurred.	R/(W) *1
b29	TUIRQ	Transmit Underflow Interrupt Status Flag	0: No transmit underflow has occurred. 1: A transmit underflow has occurred.	R/(W) *1
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit can be set to 0 by writing 0 after reading it as 1.

#### IDST Flag (Idle Status Flag)

This status flag indicates that the SSI is in idle state where communication is stopped.

This flag is set to 0 when communication starts after the SSICR.TEN bit or SSICR.REN bit is set to 1. Also, this flag is set to 1 if both the SSICR.TEN and SSICR.REN bits are set to 0 and system word communication is completed.

If the external device stops inputting the serial bit clock before communication is completed, this flag is not set to 1.

#### RSWNO Bit (Receive System Word Number)

The initial value of this bit is 1, and its value is inverted when the data is transferred from the receive shift register to the SSIFRDR register.

This bit is initialized to 1 when the REN bit value is changed from 0 to 1.

When the data word length specified by the SSICR.DWL[2:0] bits is 18 bits or more, this bit indicates which system word the data in the SSIFRDR register represents.

#### TSWNO Bit (Transmit System Word Number)

This status bit indicates the current word number.

The initial value of this is 1, and its value is inverted when the data is transferred from the SSIFTDR register to the transmit shift register.

This bit is initialized to 1 when the TEN bit value is changed from 0 to 1.

When the data word length specified by the SSICR.DWL[2:0] bits is 18 bits or more, this bit indicates the system word that is in the data transferred from the SSIFTDR register to the transmit shift register.

### **IIRQ Flag (Idle Interrupt Status Flag)**

This status flag indicates whether this module is in idle state.

This flag is set regardless of the value of the SSICR.IIEN bit to allow polling.

The interrupt can be masked by setting the SSICR.IIEN bit to 0, but the interrupt cannot be cleared by writing 0 to this flag.

If IIRQ flag = 1 and SSICR.IIEN bit = 1, an interrupt occurs.

### **ROIRQ Flag (Receive Overflow Interrupt Status Flag)**

This status flag indicates that receive data was supplied at a higher rate than was required. If a receive overflow occurs, stop reception and start from the beginning of the flowchart again.

This flag is set to 1 regardless of the setting of the SSICR.ROIEN bit. This flag can be set to 0 by writing 0 after reading it as 1.

If ROIRQ flag = 1 and SSICR.ROIEN bit = 1, an interrupt occurs.

If ROIRQ flag = 1, the data was transferred from the transmit shift register to the SSIFRDR register while the receive FIFO is full (SSIFSR.RDC[3:0] flags = 8h). This may lead to the loss of data.

Note: When an overflow occurs, the current data in the data buffer of this module is overwritten by the next incoming data from the SSI interface.

### **RUIRQ Flag (Receive Underflow Interrupt Status Flag)**

This status flag indicates that receive data was supplied at a lower rate than was required. If a receive underflow occurs, stop reception and start the flowchart again from the beginning.

This flag is set to 1 regardless of the setting of the SSICR.RUIEN bit. This flag can be set to 0 by writing 0 after reading it as 1.

If RUIRQ flag = 1 and SSICR.RUIEN bit = 1, an interrupt occurs.

If RUIRQ flag = 1, the SSIFRDR register was read while the receive FIFO is empty (SSIFSR.RDC[3:0] flags = 0h). This may cause invalid receive data to be stored.

### **TOIRQ Flag (Transmit Overflow Interrupt Status Flag)**

This status flag indicates that transmit data was supplied at a higher rate than was required. If a transmit overflow occurs, stop transmission and start from the beginning of the flowchart again.

This flag is set to 1 regardless of the setting of the SSICR.TOIEN bit. This flag can be set to 0 by writing 0 after reading it as 1.

If TOIRQ flag = 1 and SSICR.TOIEN bit = 1, an interrupt occurs.

If TOIRQ flag = 1, the SSIFTDR register had data written to it while the transmit FIFO is full (SSIFSR.TDC[3:0] flags = 8h). This may lead to the loss of data.

### **TUIRQ Flag (Transmit Underflow Interrupt Status Flag)**

This status flag indicates that transmit data was supplied at a lower rate than was required. If a transmit underflow occurs, stop transmission and start from the beginning of the flowchart again.

This flag is set to 1 regardless of the setting of the SSICR.TUIEN bit. This flag can be set to 0 by writing 0 after reading it as 1.

If TUIRQ flag = 1 and SSICR.TUIEN bit = 1, an interrupt occurs.

If TUIRQ flag = 1, the SSIFTDR register did not have data written to it before it was required for transmission. This may lead to the same data being transmitted once more.

Note: When a transmit underflow occurs, the last data input to SSIFTDR is transmitted until this module is in the idle state after transmission is stopped.

### 30.2.3 FIFO Control Register (SSIFCR)

Address(es): SSI0.SSIFCR 0008 A510h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
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
	—	—	—	—	—	—	—	—	TTRG[1:0]	RTRG[1:0]	TIE	RIE	TFRST	RFRST		
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	RFRST	Receive FIFO Data Register Reset*4	0: Clears the receive FIFO data reset. 1: Initiates the receive FIFO data reset.	R/W
b1	TFRST	Transmit FIFO Data Register Reset*4	0: Clears the transmit FIFO data reset. 1: Initiates the transmit FIFO data reset.	R/W
b2	RIE	Receive FIFO Data Full Interrupt Enable	0: Receive FIFO data full interrupt (RXI) request is disabled. 1: Receive FIFO data full interrupt (RXI) request is enabled.*1	R/W
b3	TIE	Transmit FIFO Data Empty Interrupt Enable	0: Transmit FIFO data empty interrupt (TXI) request is disabled. 1: Transmit FIFO data empty interrupt (TXI) request is enabled.*2	R/W
b5, b4	RTRG[1:0]	Receive FIFO Threshold Setting Trigger*4	b5 b4 0 0: 1 0 1: 2 1 0: 4 1 1: 6	R/W
b7, b6	TTRG[1:0]	Transmit FIFO Threshold Setting Trigger*4	b7 b6 0 0: 7 (1)*3 0 1: 6 (2)*3 1 0: 4 (4)*3 1 1: 2 (6)*3	R/W
b15 to b8	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b16	SSIRST	SSI Software Reset	0: Clears the SSI software reset. 1: Initiates the SSI software reset.	R/W
b30 to b17	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b31	AUCKE	Master Clock Enable*4	0: The master clock is disabled. 1: The master clock is enabled.	R/W

Note 1. The RXI request can be cleared by setting the SSIFSR.RDF flag to 0 (see the description of the SSIFSR.RDF flag for details) or RIE bit to 0.

Note 2. The TXI request can be cleared by setting the SSIFSR.TDE flag to 0 (see the description of the SSIFSR.TDE flag for details) or TIE bit to 0.

Note 3. The values in parenthesis are the number of empty stages in SSIFTDR at which the SSIFSR.TDE flag is set.

Note 4. Rewriting is allowed only in the idle state.

The SSIFCR register resets the number of the data bytes stored in the SSIFTDR and SSIFRDR registers, and specifies transmit FIFO and receive FIFO threshold values.

#### RFRST Bit (Receive FIFO Data Register Reset)

This bit invalidates the data in the SSIFRDR register to reset the FIFO to an empty state.

**TFRST Bit (Transmit FIFO Data Register Reset)**

This bit invalidates the data in the SSIFTDR register to reset the FIFO to an empty state.

**RIE Bit (Receive FIFO Data Full Interrupt Enable)**

This bit enables or disables generation of receive FIFO data full interrupt (RXI) requests when the SSIFSR.RDF flag is set to 1 during reception.

**TIE Bit (Transmit FIFO Data Empty Interrupt Enable)**

This bit enables or disables generation of transmit FIFO data empty interrupt (TXI) requests when the SSIFSR.TDE flag is set to 1 during transmit operation.

**RTRG[1:0] Bits (Receive FIFO Threshold Setting Trigger)**

These bits specify the receive FIFO threshold value. When the number of received data bytes stored in the SSIFRDR register (receive FIFO) has become equal to or greater than the value specified by the RTRG[1:0] bits, the SSIFSR.RDF flag is set to 1 and reading the received data is requested. If the SSIFCR.RIE bit is 1 at this time, a receive FIFO data full interrupt (RXI) request is generated.

**TTRG[1:0] Bits (Transmit FIFO Threshold Setting Trigger)**

These bits specify the transmit FIFO threshold value. When the number of transmit data bytes stored in the SSIFTDR register (transmit FIFO) has become equal to or less than the value specified by the TTRG[1:0], the SSIFSR.TDE flag is set to 1 and writing the transmit data is requested. If the SSIFCR.TIE bit is 1 at this time, a transmit FIFO data empty interrupt (TXI) request is generated.

**SSIRST Bit (SSI Software Reset)**

Writing 1 to this bit initializes the SSI internal status, registers other than the SSIFCR register, and bits other than this bit in the SSIFCR register. Since this bit is not automatically cleared to 0, confirm that 1 is written to it before writing 0. Do not write 0 to this bit and 1 to other bits at the same time. After modifying this bit, confirm that its value is modified before proceeding to the next processing.

### 30.2.4 FIFO Status Register (SSIFSR)

Address(es): SSI0.SSIFSR 0008 A514h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	TDC[3:0]			—	—	—	—	—	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RDC[3:0]			—	—	—	—	—	—	—	—	RDF
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	RDF	Receive Data Full Flag	0: Number of received data bytes in the SSIFRDR register is less than the set receive trigger number. 1: Number of received data bytes in the SSIFRDR register is equal to or greater than the set receive trigger number.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	RDC[3:0]	Receive Data Indicate Flag	Indicate the number of data units stored in the SSIFRDR register.	R
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	TDE	Transmit Data Empty Flag	0: Number of data bytes for transmission in the SSIFTDR register is greater than the set transmit trigger number. 1: Number of data bytes for transmission in the SSIFTDR register is equal to or less than the set transmit trigger number.*2	R/(W) *1
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	TDC[3:0]	Transmit Data Indicate Flag	Indicate the number of data units stored in the SSIFTDR register.	R
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit can be set to 0 by writing 0 after reading it as 1.

Note 2. Since the SSIFTDR register is an 8-stage FIFO register, the amount of data that can be written to it while TDE flag = 1 is "8 - transmit trigger number to be specified" bytes at maximum. Writing more data will be ignored. The number of data bytes in the SSIFTDR register is indicated in the TDC[3:0] flags.

The SSIFSR register consists of status flags indicating the operating status of the SSIFTDR register and SSIFRDR register.

#### RDF Flag (Receive Data Full Flag)

This flag indicates that, when the received data is transferred to the SSIFRDR register, the number of data bytes in the SSIFRDR register has become equal to or greater than the receive FIFO threshold value, and thus reading the received data from the SSIFRDR register has been enabled.

[Setting condition]

- The number of receive data bytes that is equal to or greater than the receive FIFO threshold value is stored in the SSIFRDR register.

[Clearing conditions]

- 0 is written to the RDF flag after the RDF flag is confirmed to be 1.
- Received data is read from the SSIFRDR register using DTC transfer (transfer of the last block in block transfer). Do not clear the RDF flag to 0 during DTC transfer.

Note: Since the SSIFRDR register is a 32-byte FIFO register, the maximum number of data bytes that can be read from it while the RDF flag is 1 is indicated in the RDC[3:0] flags. If reading data from the SSIFRDR register is

continued after all the data is read, undefined values will be read.

### **RDC[3:0] Flags (Receive Data Indicate Flag)**

These flags indicate the number of data bytes stored in the SSIFRDR register.

RDC[3:0] flags = 0h indicates no received data. RDC[3:0] flags = 8h indicates that 32 bytes of received data is stored in the SSIFRDR register.

### **TDE Flag (Transmit Data Empty Flag)**

This flag indicates that, when data is transferred from the SSIFTDR register to the SSITDR register, the number of data bytes in the SSIFTDR register has become less than the transmit FIFO threshold value, and thus writing transmit data to the SSIFTDR register has been enabled.

[Setting condition]

- The number of the transmit data bytes written to the SSIFTDR register is equal to or less than the transmit FIFO threshold value.

[Clearing conditions]

- 0 is written to the TDE flag after the TDE flag is confirmed to be 1.
- Transmit data is written to the SSIFTDR register using DTC transfer (transfer of the last block in block transfer). Do not clear the TDE flag to 0 during DTC transfer.

Note: Since the SSIFTDR register is a 32-byte FIFO register, the maximum number of bytes that can be written to it while the TDE flag is 1 is 8 - TDC[3:0]. If writing data to the SSIFTDR register is continued after all the data is written, writing will be invalid and an overflow occurs.

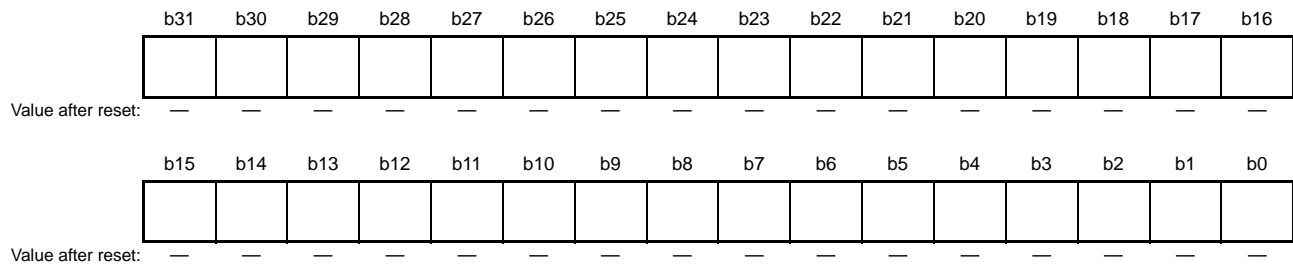
### **TDC[3:0] Flags (Transmit Data Indicate Flag)**

These flags indicate the number of data bytes stored in the SSIFTDR register.

TDC[3:0] flags = 0h indicates no data for transmission. TDC[3:0] flags = 8h indicates that 32 bytes of data for transmission is stored in the SSIFTDR register.

### 30.2.5 Transmit FIFO Data Register (SSIFTDR)

Address(es): SSI0.SSIFTDR 0008 A518h

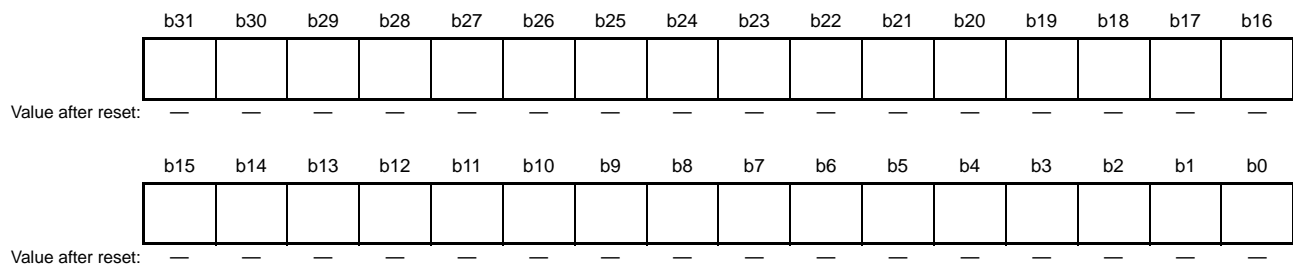


The SSIFTDR register is a write-only FIFO register consisting of eight stages of 32-bit registers for storing data to be serially transmitted. Write transmit data to the SSIFTDR register in 64-bit (two-stage FIFO) units regardless of the data word length setting. If transmit data ends on a 32-bit boundary, write 32-bit 0-data (0000 0000h) after the last transmit data is written, and stop transmission while 64-bit writing is completed. When the transmit shift register is empty, the SSI transfers the transmit data written to the SSIFTDR register to start serial transmission, which can be continued until the SSIFTDR register becomes empty.

Note that when the SSIFTDR register is full of data (32 bytes), the next data cannot be written to it. If writing is attempted, it will be ignored and an overflow occurs.

### 30.2.6 Receive FIFO Data Register (SSIFRDR)

Address(es): SSI0.SSIFRDR 0008 A51Ch



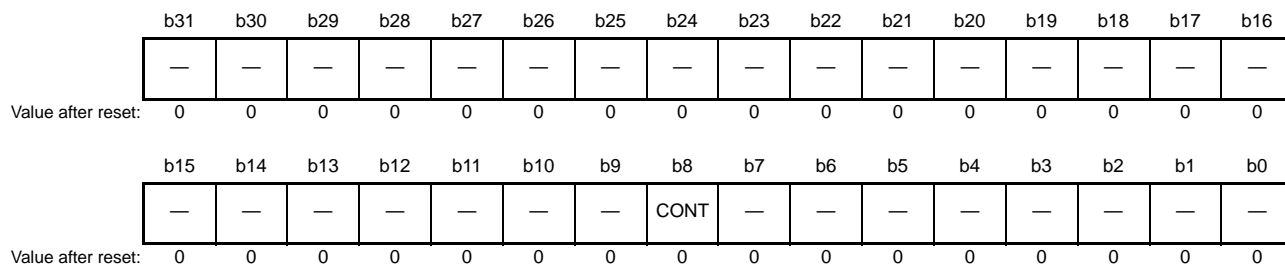
The SSIFRDR register is a read-only FIFO register consisting of eight stages of 32-bit registers for storing serially received data. Each time 4 bytes of serial data is received, the SSI stores the received serial data in the SSIFRDR register from the receive shift register according to the PDTA bit setting. Receive operation can be continued until a maximum 32 bytes of data have been stored to in the SSIFRDR register. The SSIFRDR register can be read but cannot be written to. Note that when the SSIFRDR register is read when it stores no received data, undefined values will be read and a receive underflow occurs.

After the SSIFRDR register becomes full of received data, the data received thereafter will be lost and a receive overflow occurs.



### 30.2.7 TDM Mode Register (SSITDMR)

Address(es): SSI0.SSITDMR 0008 A520h



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	CONT	WS Continue Mode*1	0: Disables WS continue mode. 1: Enables WS continue mode.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit can be set only in master mode (SSICR.SCKD bit = 1 and SSICR.SWSD bit = 1).

The SSITDMR register is a readable/writable 32-bit register that enables or disables WS continue mode.

## 30.3 Operation

### 30.3.1 Bus Format

This module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode. The bus format can be selected from one of the six modes shown in Table 30.4.

**Table 30.4 Bus Format**

	TEN	REN	SCKD	SWSD	MUEN	I IEN	TO IEN	TUI EN	ROI EN	RUI EN	CONT	SWSP	DEL	PDTA	SDTA	SPDP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
Non-compression slave receiver	0	1	0	0	Control bits						Configuration bits									
Non-compression slave transmitter	1	0	0	0																
Non-compression slave transceiver	1	1	0	0																
Non-compression master receiver	0	1	1	1																
Non-compression master transmitter	1	0	1	1																
Non-compression master transceiver	1	1	1	1																

### 30.3.2 Non-Compressed Mode

This SSI supports non-compressed mode only. It supports the I<sup>2</sup>S compatible format as well as MSB-first and left-aligned/right-aligned.

#### (1) Slave Receiver

This mode allows the module to receive serial data from another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

#### (2) Slave Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

#### (3) Slave Transceiver

This mode allows serial data transmission and reception between this module and another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

#### (4) Master Receiver

This mode allows the module to receive serial data from another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of this module. If the incoming data does not follow the configured format, operation is not guaranteed.

#### (5) Master Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of this module.

(6) Master Transceiver

This mode allows serial data transmission and reception between this module and another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of this module.

(7) Operating Settings Related to Word Length

All bits related to the SSICR register's word length are valid in non-compressed modes. There are many configurations this module supports, but some of the combinations are shown below for the I<sup>2</sup>S compatible format, MSB-first and left-aligned format, and MSB-first and right-aligned format.

- I<sup>2</sup>S Compatible Format

Figure 30.2 and Figure 30.3 show the I<sup>2</sup>S compatible format both without and with padding. Padding occurs when the data word length is smaller than the system word length.

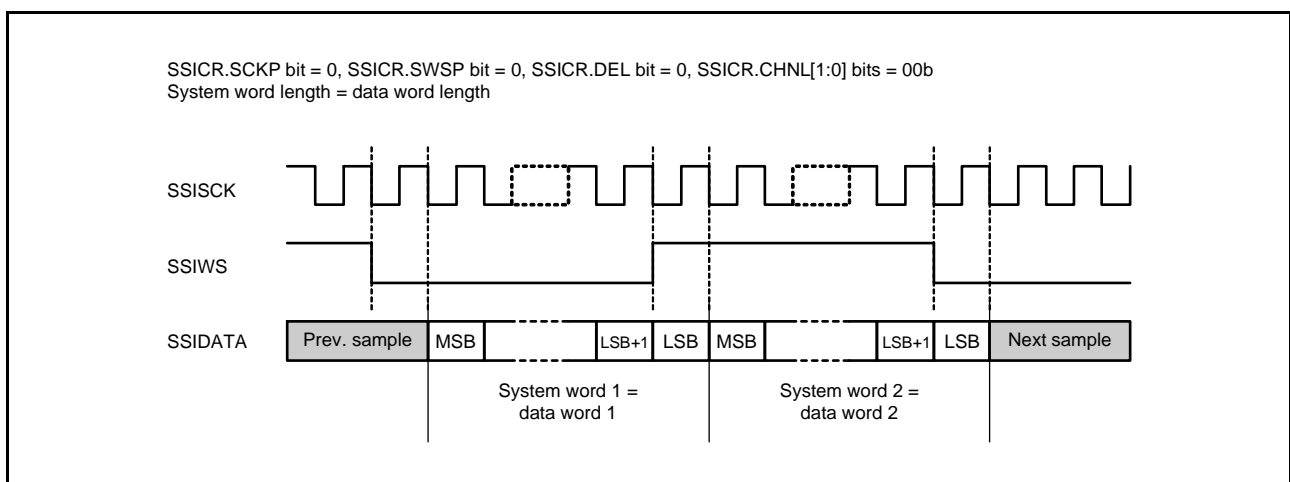


Figure 30.2 I<sup>2</sup>S Compatible Format (without Padding)

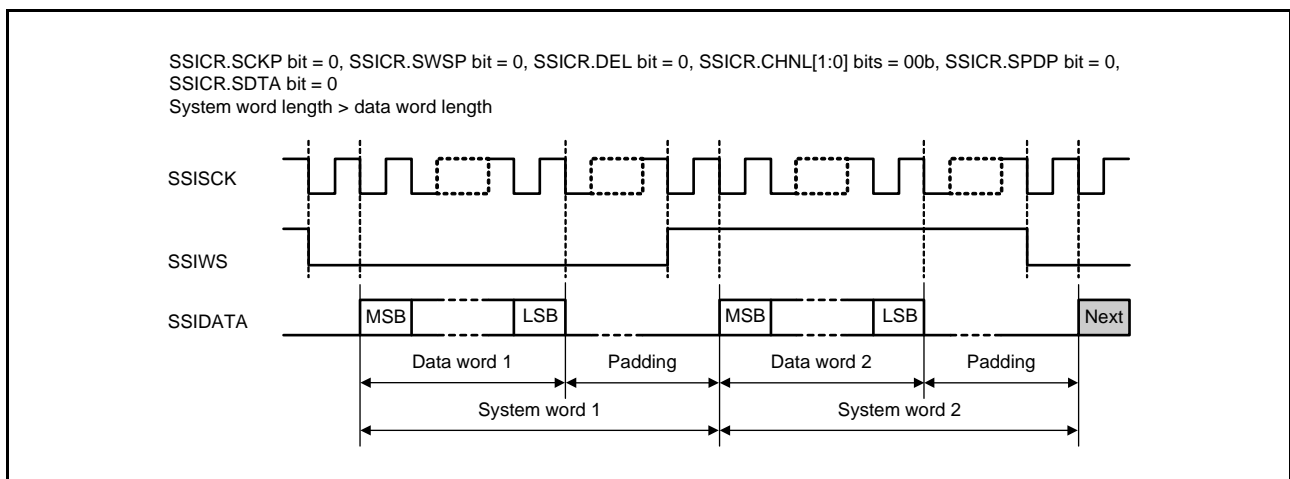
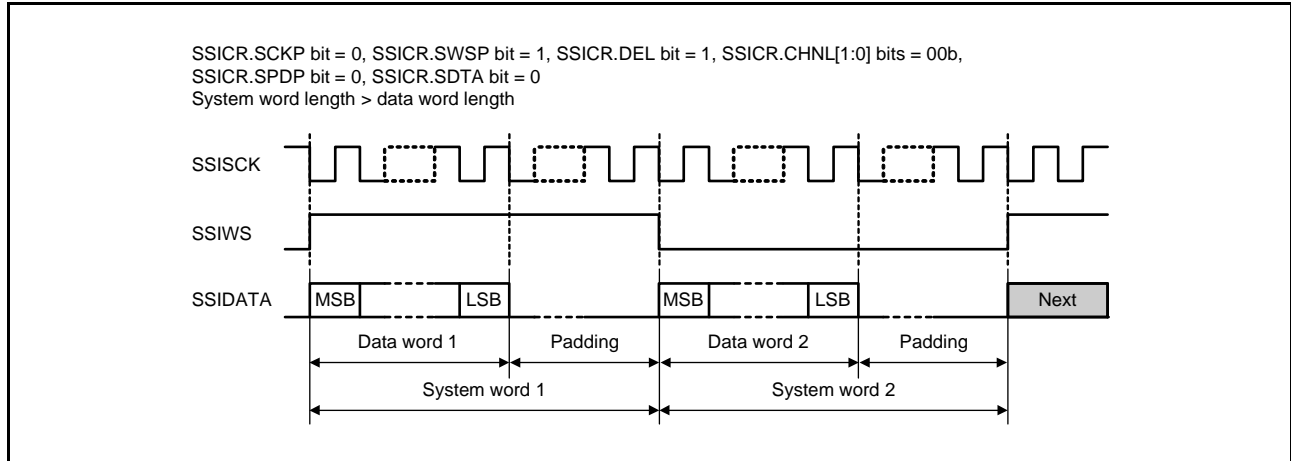


Figure 30.3 I<sup>2</sup>S Compatible Format (with Padding)

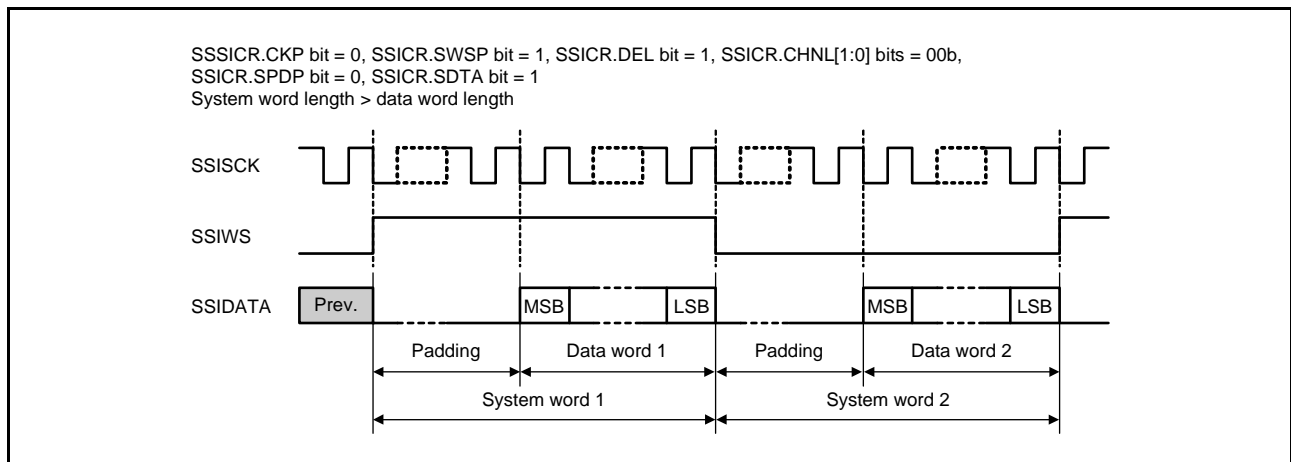
Figure 30.4 shows the MSB-first and left-aligned format and Figure 30.5 shows the MSB-first and right-aligned format.

- MSB-First and Left-Aligned Format



**Figure 30.4 MSB-First and Left-Aligned Format**  
(Transmitted and Received in the Order of Serial Data and Padding Bits)

- MSB-First and Right-Aligned Format



**Figure 30.5 MSB-First and Right-Aligned Format**  
(Transmitted and Received in the Order of Padding Bits and Serial Data)

Table 30.5 shows the number of padding bits for each of the valid setting.

**Table 30.5 Number of Padding Bits for Each Valid Setting**

Padding Bits per System Word			SSICR.DWL[2:0] Bits	000b	001b	010b	011b	100b	101b
SSICR.CHNL[1:0] Bits	Decoded Channels per System Word	SSICR.SWL[2:0] Bits	Data Word Length / System Word Length	8	16	18	20	22	24
00b	1	000b	8	0	—	—	—	—	—
		001b	16	8	0	—	—	—	—
		010b	24	16	8	6	4	2	0
		011b	32	24	16	14	12	10	8

(8) Operating Settings Other than Word Length Related Settings

Several more configuration bits in non-compressed mode are shown below. These bits are not mutually exclusive, but some combinations may not be useful for any other device.

These configuration bits are described below with reference to the basic format sample in Figure 30.6.

In Figure 30.6 to Figure 30.14, a system word length of 6 bits and a data word length of 4 bits are used for simplification of these figures.

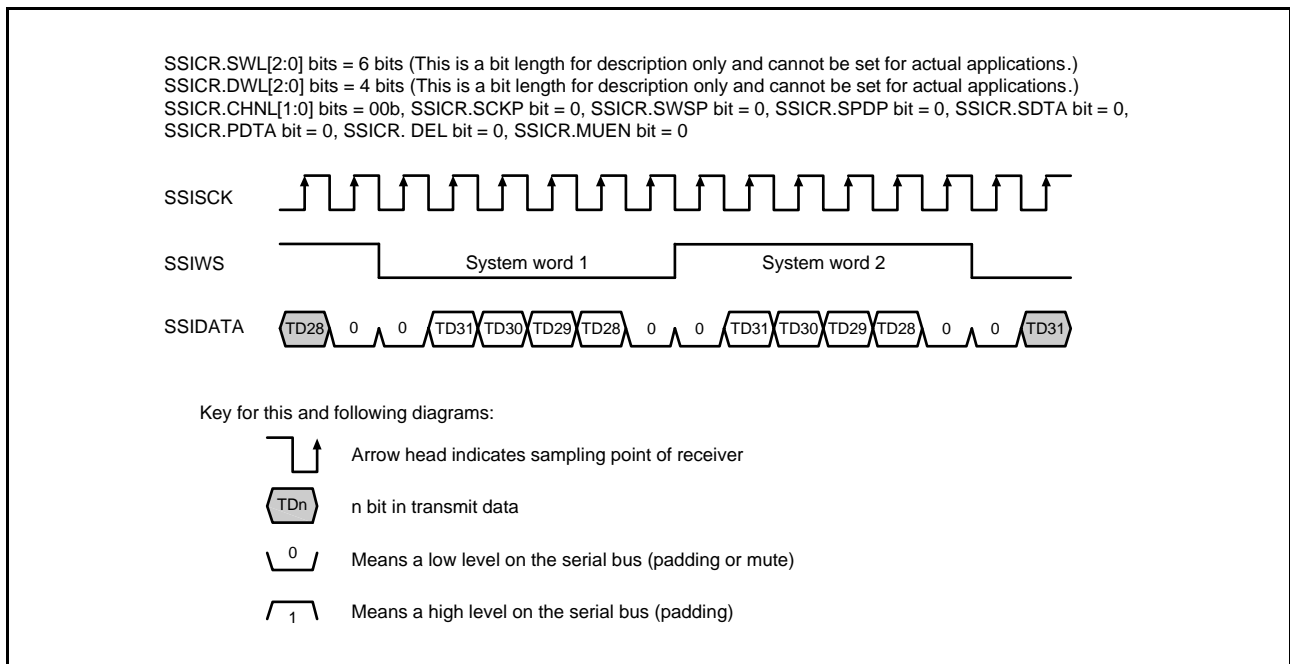


Figure 30.6 Basic Format Sample (Transmit Mode)

- Inverted Clock

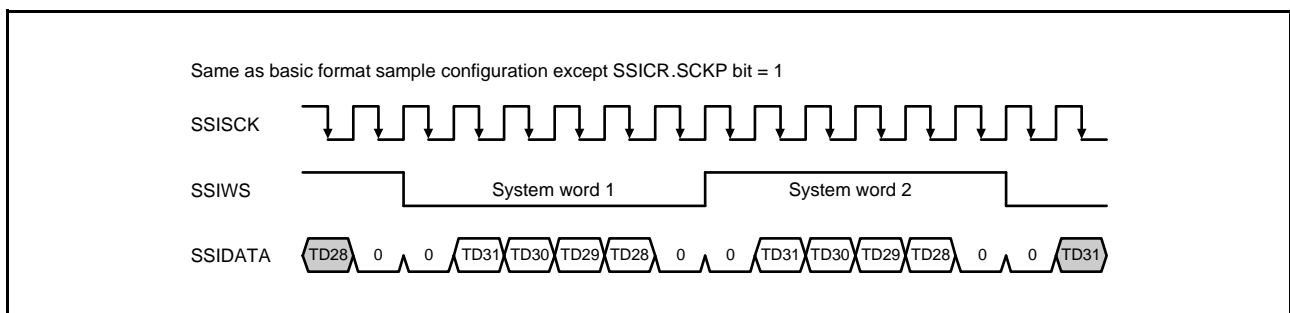


Figure 30.7 Inverted Clock

- Inverted Word Select

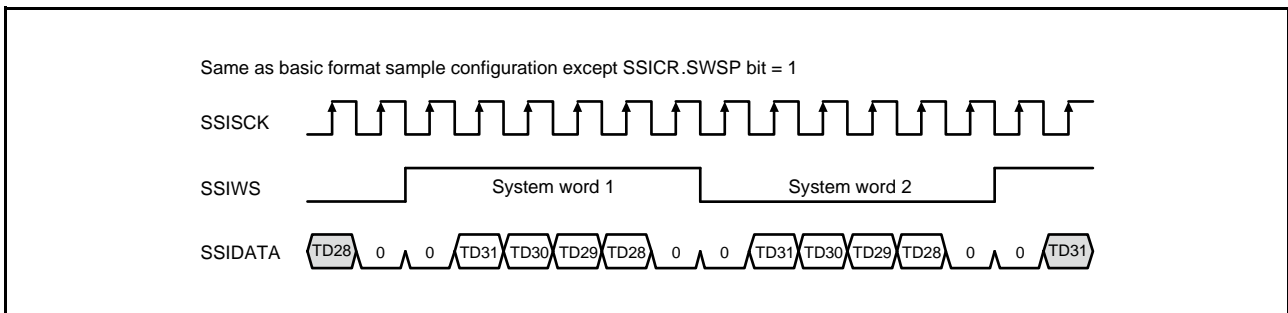


Figure 30.8 Inverted Word Select

- Inverted Padding Polarity

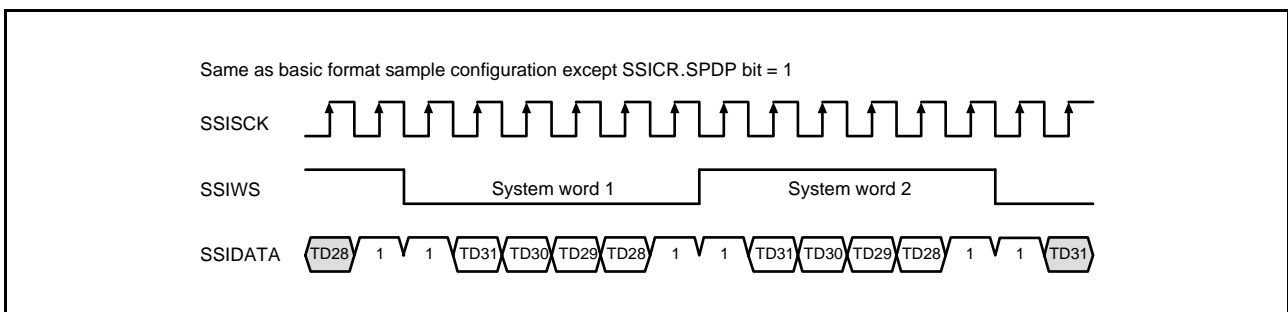


Figure 30.9 Inverted Padding Polarity

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

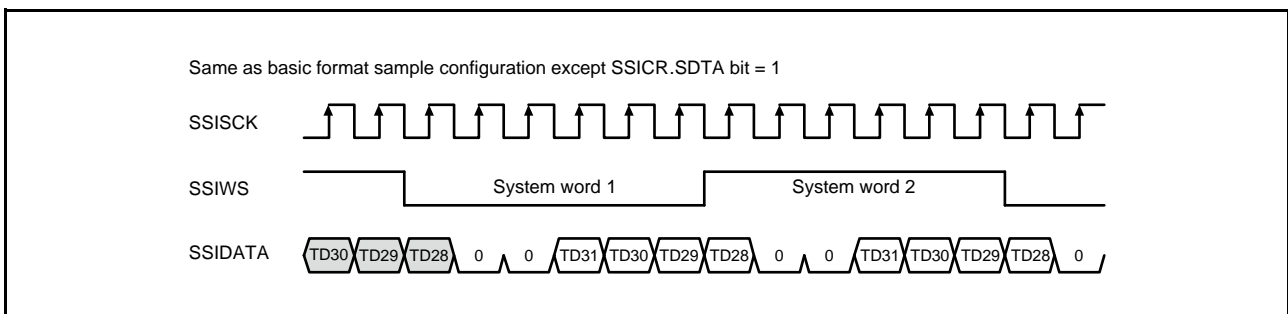


Figure 30.10 Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

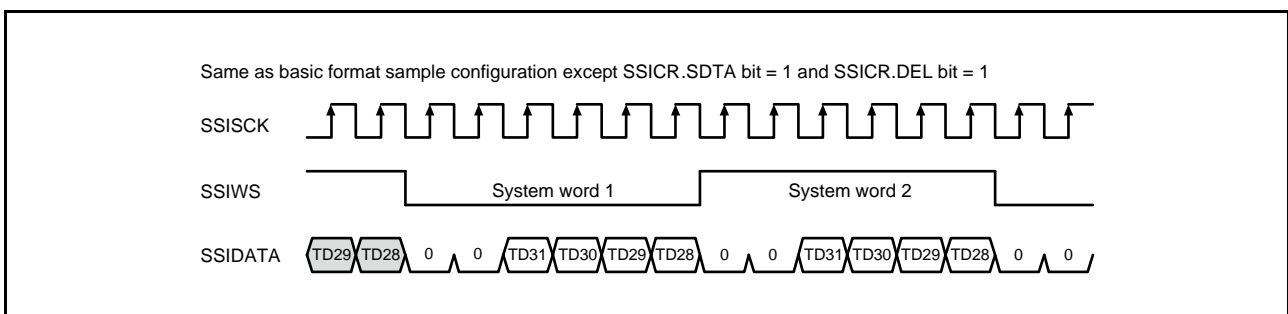


Figure 30.11 Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

- Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

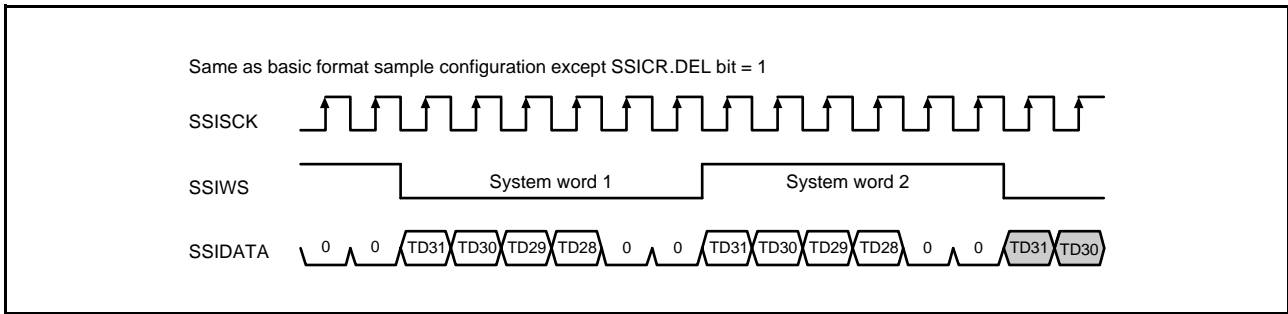


Figure 30.12 Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

- Parallel Right-Aligned with Delay

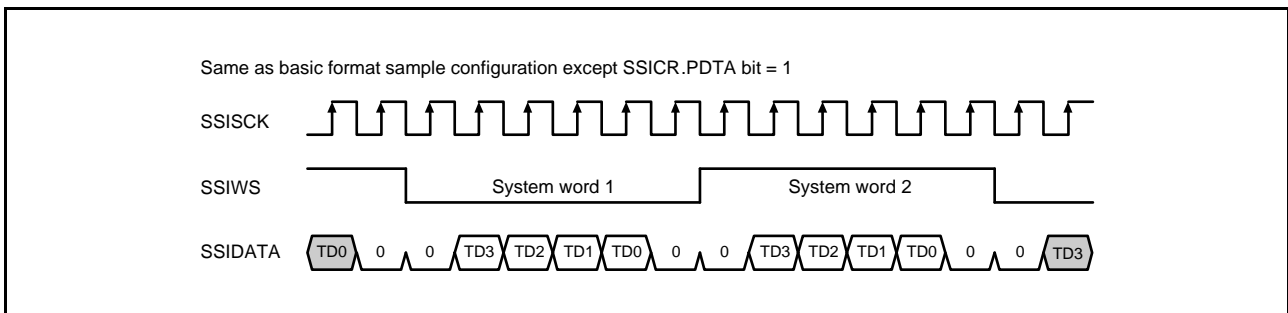


Figure 30.13 Parallel Right-Aligned with Delay

- Mute Enabled

When the SSICR.MUEN bit is set, the SSITXD0 and SSIDATA1 pins are set to 0 without synchronizing SSIWS.

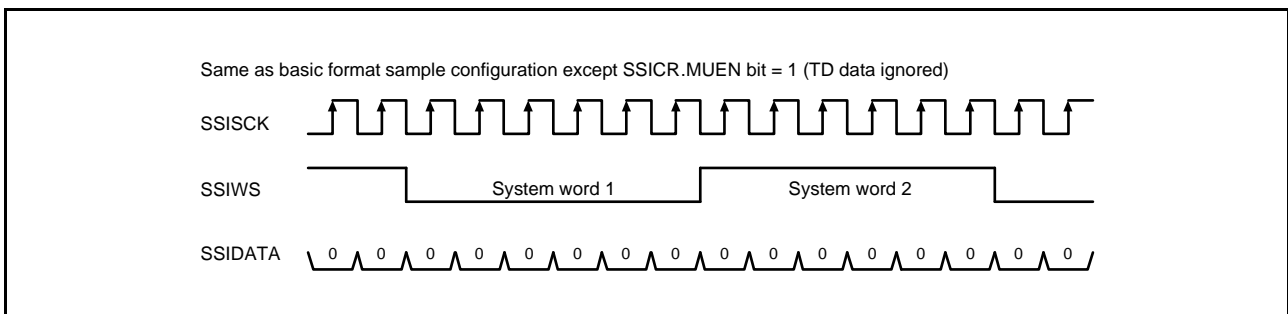
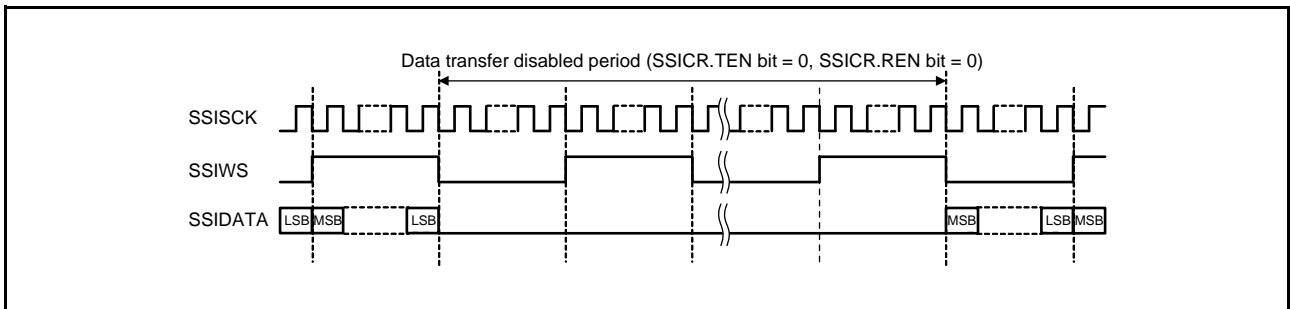


Figure 30.14 Mute Enabled

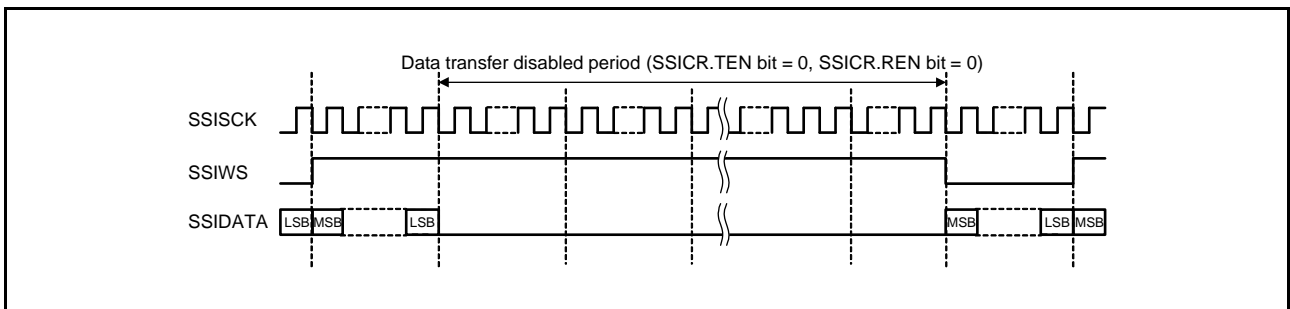
### 30.3.3 WS Continue Mode

In WS continue mode, the SSIWS signal continues to be output irrespective whether data transfer is enabled or disabled. This mode can be set using the SSITDMR.CONT bit. With this mode enabled, the SSIWS signal does not stop but continues operating even if the SSICR.TEN and REN bits are both set to 0 (transfer disabled). With this mode disabled, the SSIWS signal stops if the SSICR.TEN and REN bits are both set to 0.

Figure 30.15 and Figure 30.16 show the operations with WS continue mode enabled and disabled, respectively.



**Figure 30.15 WS Continue Mode Enabled**

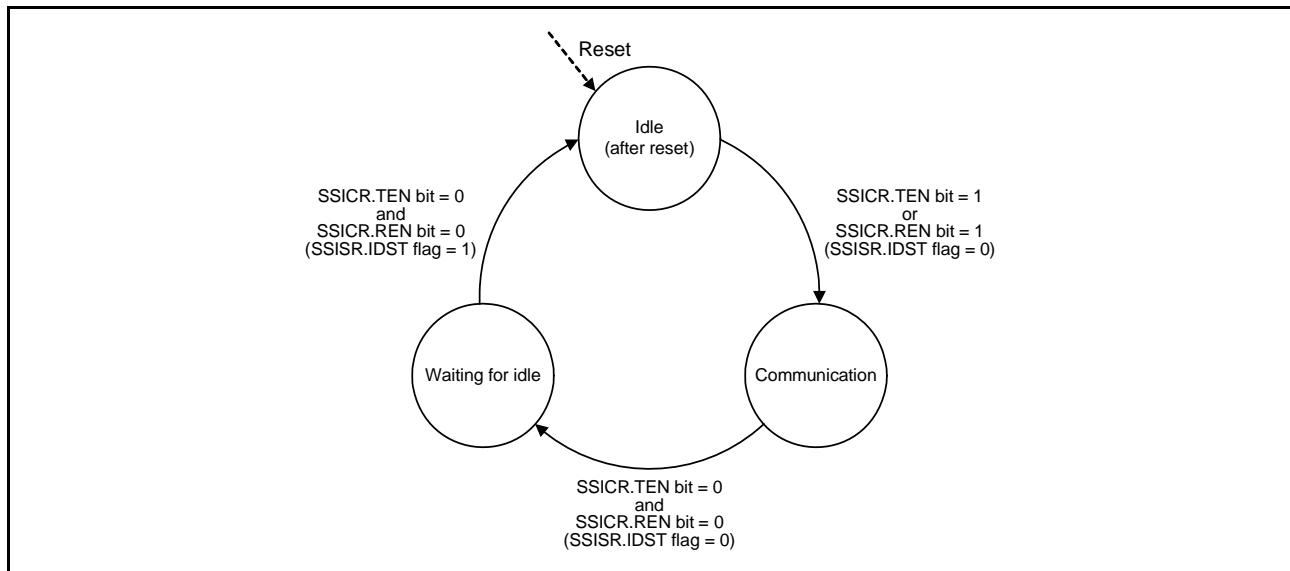


**Figure 30.16 WS Continue Mode Disabled**



### 30.3.4 Operating States

There are three states of operation: idle, communication, and waiting for idle. Figure 30.17 shows the operating state transitions.



**Figure 30.17** Operating State Transitions

#### (1) Idle State

This module enters this state when the MSTPCRD.MSTPD14 and MSTPD15 bits are set to 0 after a reset is released. All required configuration fields in the control register should be defined in this state. After the settings are made, the module enters communication state when the SSICR.TEN bit or SSICR.REN bit is set to 1.

#### (2) Communication State

Communication in this state depends on the selected operating state. For details, refer to section 30.3.5, Transmit Operation and section 30.3.6, Receive Operation.

#### (3) Waiting for Idle

This module enters this state when both the SSICR.TEN and SSICR.REN bits are set to 0 in communication state. If system word communication is completed in this state, the SSISR.IDST flag is set to 1 and this module enters the idle state.

### 30.3.5 Transmit Operation

Transmission can be controlled either by DTC transfer or interrupt.

DTC control is preferred to reduce the processor load. In transmission using the DTC, the processor will only receive interrupts if there is an underflow or overflow of data or if DTC transfer has been completed. In transmission using DTC transfer, set the number of DTC transfers to multiples of 2 to write transmit data to the SSIFTDR register in 64-bit (two-stage FIFO) units.

The alternative method is using the interrupts that this module generates to supply data as required. In transmission using interrupts, write transmit data in 64-bit units regardless of the data format. If transmit data ends on a 32-bit boundary, write 32-bit 0-data (0000 0000h) after the last transmit data is written, and complete writing on a 64-bit boundary.

When stopping transmission, stop writing to the SSIFTDR register while 64-bit writing is completed. After writing is stopped, wait until a transmit underflow occurs before setting the TEN bit to 0. During transmit underflow, the last data input to SSIFTDR is continuously transmitted until this module enters the idle state. After the TEN bit is set to 0, the clock\*<sup>1</sup> should continue to be supplied until the SSISR.IIRQ flag indicates that the module is in the idle state. If a transmit underflow error or transmit overflow error occurs during data transmission, transmit data to SSIFTDR may not be written in a 64-bit units. In that case, stop writing data, wait until a transmit underflow error occurs, and check the TSWNO when the transmit underflow has occurred. When the TSWNO bit is 1, write 32-bit 0-data (0000 0000h) to SSIFTDR and wait until an underflow occurs again. Once the TSWNO bit is confirmed to be 0, after the TEN bit is set to 0, the clock\*<sup>1</sup> should continue to be supplied until the SSISR.IIRQ flag indicates that the module is in the idle state.

Figure 30.18 shows transmission using the DTC, and Figure 30.19 shows transmission using interrupts.

Note 1. Input clock from the SSISCK pin when SSICR.SCKD bit = 0.  
Master clock when SSICR.SCKD bit = 1.

## (1) Transmission Using the DTC

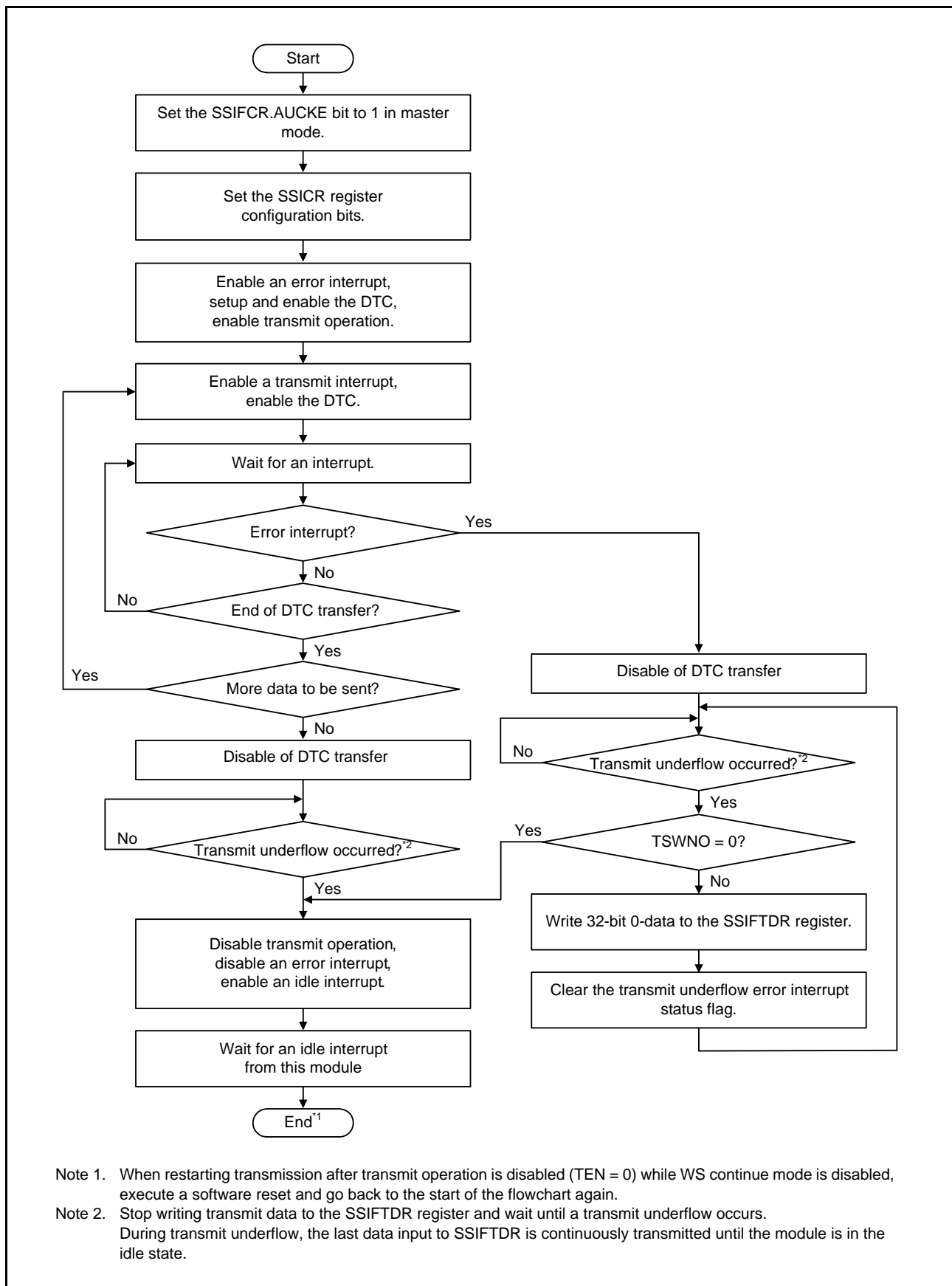


Figure 30.18 Transmission Using the DTC

## (2) Transmission Using Interrupts

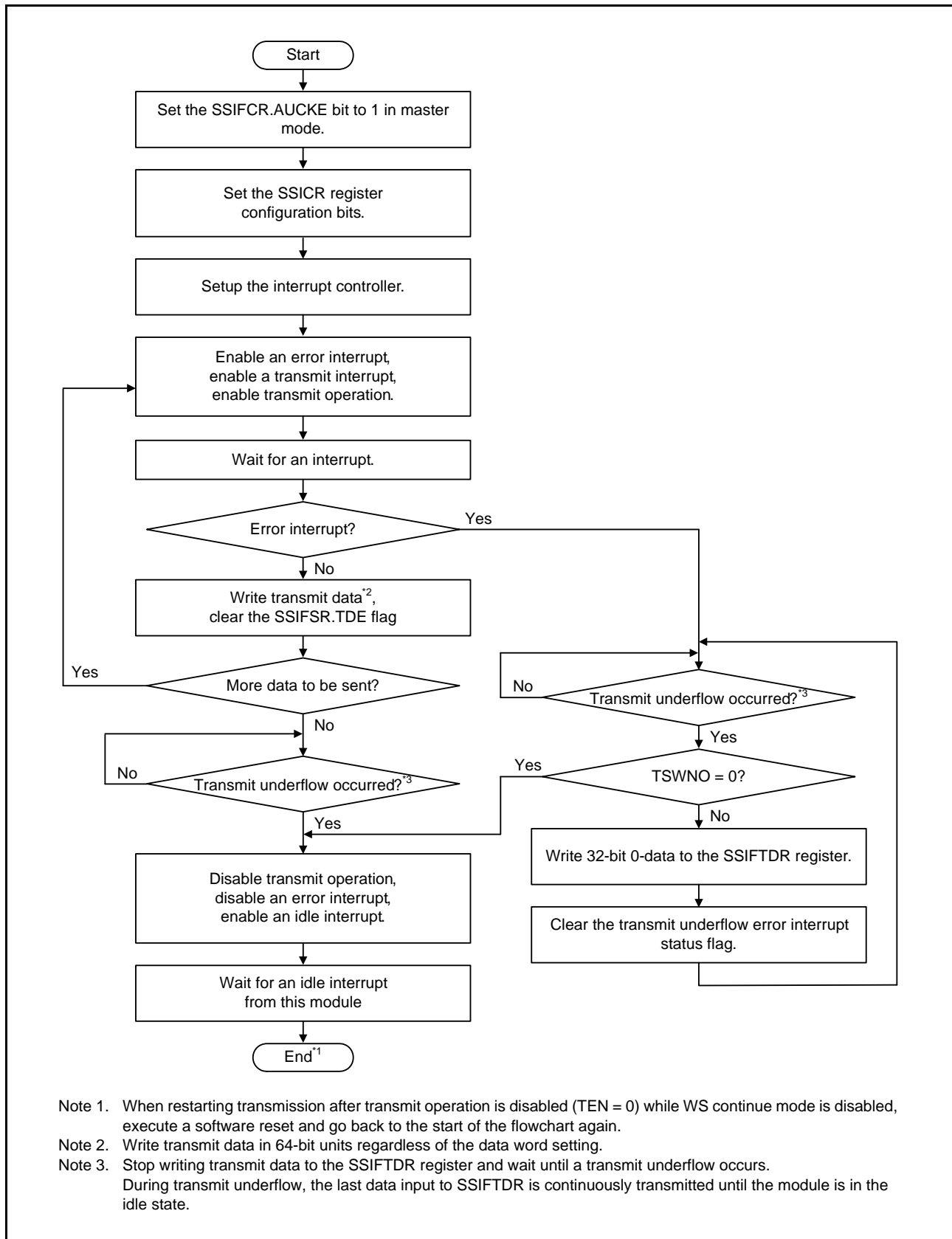


Figure 30.19 Transmission Using Interrupts

### 30.3.6 Receive Operation

Like transmission, reception can be controlled either by DTC transfer or interrupt.

Figure 30.20 and Figure 30.21 show the flow of operation.

When stopping reception, the clock\*<sup>1</sup> should continue to be supplied after the REN bit is set to 0 until the SSISR.IIRQ flag indicates that the module is in the idle state.

Note 1. Input clock from the SSISCK pin when SSICR.SCKD bit = 0.  
Master clock when SSICR.SCKD bit = 1.

#### (1) Reception Using the DTC

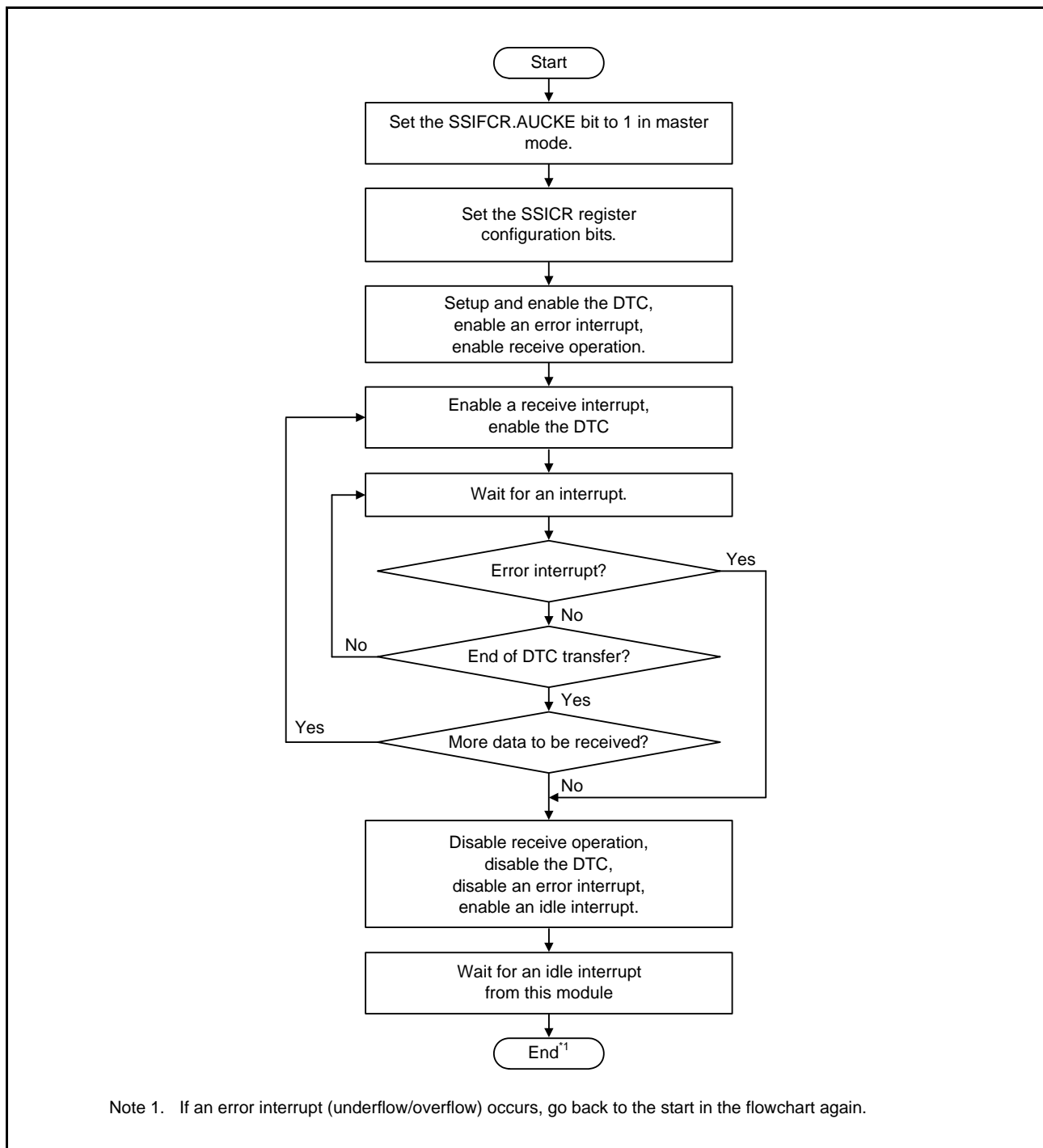


Figure 30.20 Reception Using the DTC

(2) Reception Using Interrupts

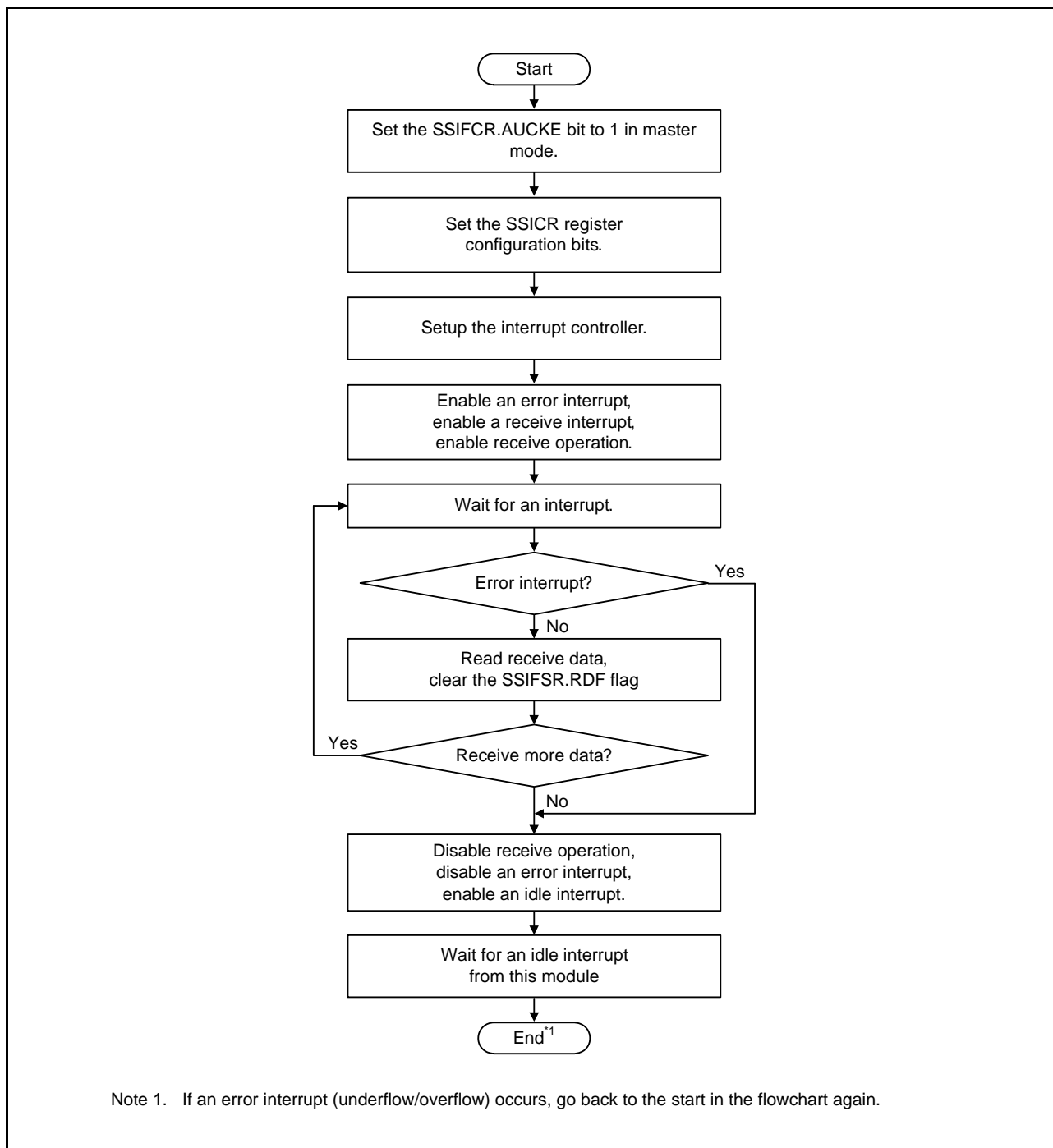


Figure 30.21 Reception Using Interrupts

### 30.3.7 Serial Bit Clock Control

The SSI controls and selects the serial bit clock, according to the SCKD and CKDV bit setting.

If the serial bit clock direction is set to input (SSICR.SCKD bit = 0), this module is in clock slave mode and the shift register uses the bit clock that was input to the SSISCK pin.

If the serial bit clock direction is set to output (SSICR.SCKD bit = 1), this module is in clock master mode, and the shift register uses the master clock or a divided master clock as the bit clock. The master clock is divided by the ratio specified by the SSICR.CKDV[3:0] bits for use as the bit clock by the shift register.

In either case the module pin, SSISCK, is the same as the bit clock.

## 30.4 Interrupt Sources

Table 30.6 lists the interrupt sources of the SSI. Each interrupt source can be enabled or disabled by the SSICR.TUIEN, TOIEN, RUIEN, ROIEN and IIEN bits, and the SSIFCR.TIE and RIE bits.

**Table 30.6 SSI Interrupt Sources**

Channel	Interrupt Source	Description	Interrupt Flag	DTC Activation
SSI0	SSIF0	Transmit underflow interrupt, Transmit overflow interrupt, Receive underflow interrupt, Receive overflow interrupt, Idle interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIIRQ SSISR.IIRQ	Not possible
	SSIRXIO	Receive data full interrupt	SSIFSR.RDF	Possible
	SSITXIO	Transmit data empty interrupt	SSIFSR.TDE	Possible

## 30.5 Usage Notes

### 30.5.1 Setting the Module Stop Function

Module stop state can be entered or released using the MSTPCRD register. The initial setting of the SSI is in the module stop state. SSI register access is enabled by releasing the module stop state.

For details on the MSTPCRD register, refer to section 11, Low Power Consumption.

### 30.5.2 Notes on Changing Transfer Modes

For mode transitions between the transmitter, receiver, and transceiver while WS continue mode is disabled (SSITDMR.CONT = 0), set the SSICR.TEN and SSICR.REN bits to 0 and make a transition to the idle state once. Set the SSICR.TEN and SSICR.REN bits again while the module is in the idle state and restart transfer.

### 30.5.3 Limits on WS Continue Mode

If WS continue mode setting is changed, the operation of the SSISCK and SSIWS signals immediately after switching are not guaranteed. If it affects the device to be connected, do not change the setting dynamically.



## 31. Serial Peripheral Interface (RSPI)

In this section, “PCLK” is used to refer to PCLKB.

### 31.1 Overview

This MCU includes one channel 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 31.1 lists the specifications of the RSPI, and Figure 31.1 shows a block diagram of the RSPI.

In this section, m as used with the RSPI command registers (SPCMDm) indicates 0 to 7.

**Table 31.1 RSPI Specifications (1/2)**

Item	Description
Number of channels	One channel
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 (4-wire method) or clock synchronous operation (3-wire method).</li> <li>Transmit-only operation is available.</li> <li>Communication mode: Full-duplex or transmit-only can be selected.</li> <li>Switching of the polarity of RSPCK</li> <li>Switching of the phase of RSPCK</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 division ratio ranges from divided by 2 to divided by 4096).</li> <li>In slave mode, the minimum PCLK clock divided by 8 can be input as RSPCK (the maximum frequency of RSPCK is that of PCLK divided by 8).</li> </ul> <p>Width at high level: 4 cycles of PCLK; width at low level: 4 cycles of PCLK</p>
Buffer configuration	<ul style="list-style-type: none"> <li>Double buffer configuration for the transmit/receive buffers</li> <li>128 bits for the transmit/receive buffers</li> </ul>
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 pins (SSLA0 to SSLA3) for each channel</li> <li>In single-master mode, SSLA0 to SSLA3 pins are output.</li> <li>In multi-master mode: <ul style="list-style-type: none"> <li>SSLA0 pin for input, and SSLA1 to SSLA3 pins for either output or unused.</li> </ul> </li> <li>In slave mode: <ul style="list-style-type: none"> <li>SSLA0 pin for input, and SSLA1 to SSLA3 pins for unused.</li> </ul> </li> <li>Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) <ul style="list-style-type: none"> <li>Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)</li> </ul> </li> <li>Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) <ul style="list-style-type: none"> <li>Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)</li> </ul> </li> <li>Controllable wait for next-access SSL output assertion (next-access delay) <ul style="list-style-type: none"> <li>Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)</li> </ul> </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: <ul style="list-style-type: none"> <li>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> </ul> </li> <li>A transfer can be initiated by writing to the transmit buffer.</li> <li>MOSI signal value specifiable in SSL negation</li> </ul>

**Table 31.1 RSPi Specifications (2/2)**

Item	Description
Interrupt sources	<ul style="list-style-type: none"><li>• Interrupt sources<ul style="list-style-type: none"><li>Receive buffer full interrupt</li><li>Transmit buffer empty interrupt</li><li>RSPi error interrupt (mode fault, overrun, parity error)</li><li>RSPi idle interrupt (RSPi idle)</li></ul></li></ul>
Others	<ul style="list-style-type: none"><li>• Function for switching between CMOS output and open-drain output</li><li>• Function for initializing the RSPi</li><li>• Loopback mode</li></ul>
Low power consumption function	Module stop state can be set.

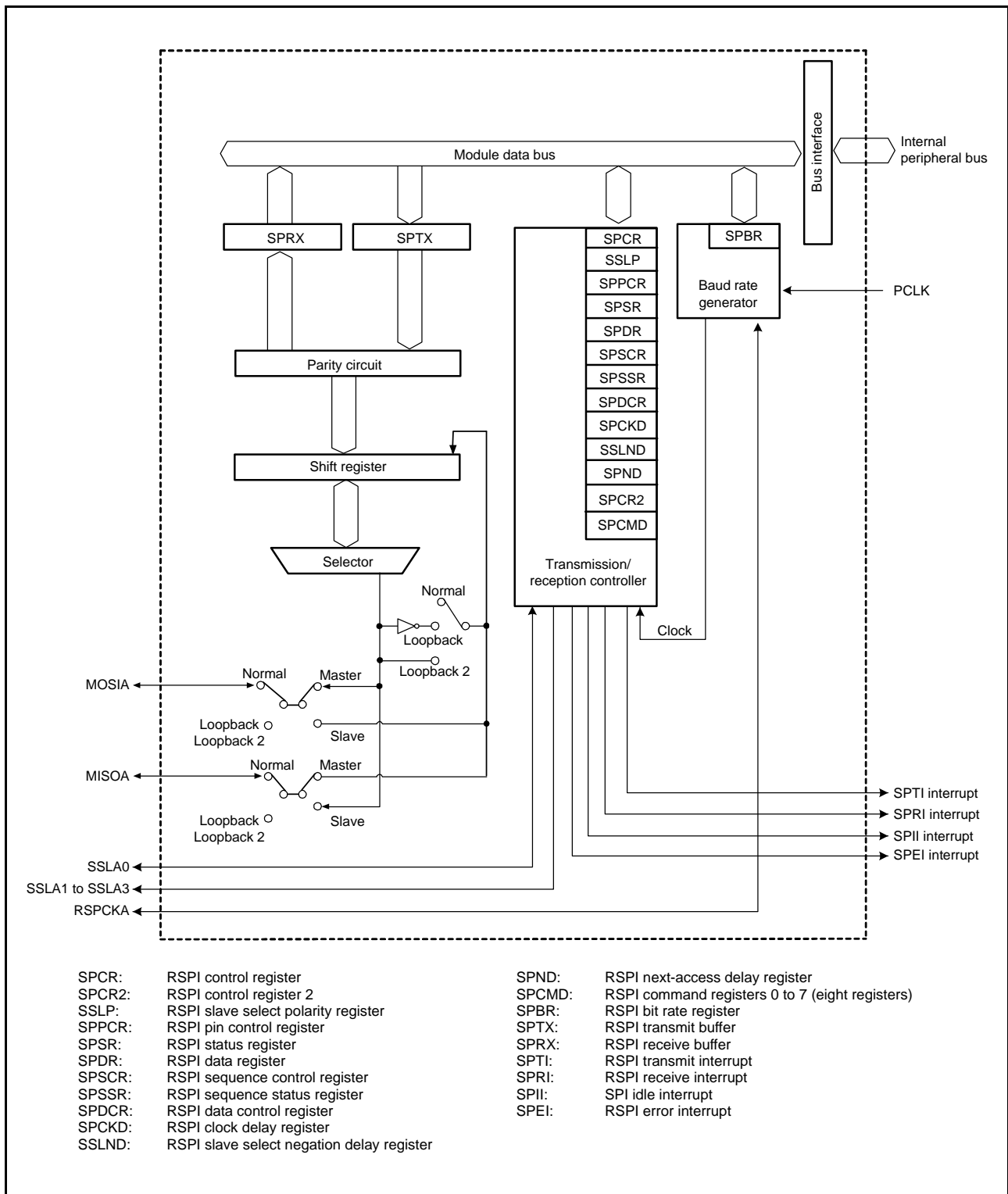


Figure 31.1 RSPI Block Diagram

Table 31.2 lists the I/O pins used in the RSPI.

The RSPI automatically switches the I/O direction of the SSLA0 pin. SSLA0 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 RSPCKA, MOSIA, and MISOA are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLA0 pin. Refer to section 31.3.2, Controlling RSPI Pins for details.

**Table 31.2 RSPI Pin Configuration**

Channel	Pin Name	I/O	Function
RSPI0	RSPCKA	I/O	Clock I/O
	MOSIA	I/O	Master transmit data I/O
	MISOA	I/O	Slave transmit data I/O
	SSLA0	I/O	Slave selection I/O
	SSLA1	Output	Slave selection output
	SSLA2	Output	Slave selection output
	SSLA3	Output	Slave selection output

## 31.2 Register Descriptions

### 31.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 0008 8380h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODF EN	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 (4-wire method) 1: Clock synchronous operation (3-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	Transmit Buffer Empty Interrupt Enable	0: Disables the generation of transmit buffer empty interrupt requests 1: Enables the generation of transmit buffer empty 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 Buffer Full Interrupt Enable	0: Disables the generation of RSPI receive buffer full interrupt requests 1: Enables the generation of RSPI receive buffer full interrupt requests	R/W

If the SPCR.MSTR, SPCR.MODFEN, or SPCR.TXMD bit is changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

#### SPMS Bit (RSPI Mode Select)

The SPMS bit selects SPI operation (4-wire method) or clock synchronous operation (3-wire method).

The SSLA0 to SSLA3 pins are not used in clock synchronous operation. The RSPCKA, MOSIA, and MISOA pins 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 should not be performed 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 (refer to section 31.3.6, 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 (refer to section 31.3.8, Error Detection). In addition, the RSPI determines the I/O direction of the SSLA0 to SSLA3 pins based on combinations of the MODFEN and MSTR bits (refer to section 31.3.2, 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 RSPCKA, MOSIA, MISOA, and SSLA0 to SSLA3.

**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 (refer to section 31.3.8, Error Detection).

**SPTIE Bit (Transmit Buffer Empty Interrupt Enable)**

The SPTIE bit enables or disables the generation of transmit buffer empty interrupt requests when the RSPI detects when the transmit buffer is empty.

A transmit buffer empty interrupt request when transmission starts is generated by setting the SPE and SPTIE bits to 1 at the same time or by setting the SPE bit to 1 after setting the SPTIE bit to 1.

Note that a transmit buffer interrupt is generated when the SPTIE bit is 1 even if the RSPI function is disabled (the SPTIE bit is changed to 0).

**SPE Bit (RSPI Function Enable)**

The SPE bit enables or disables the RSPI function.

When the SPSR.MODF flag is 1, the SPE bit cannot be set to 1. For details, refer to section 31.3.8, 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 31.3.9, Initializing RSPI. Furthermore, a transmit buffer empty 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 Buffer Full Interrupt Enable)**

If the RSPI has detected a receive buffer full write after completion of a serial transfer, the SPRIE bit enables or disables the generation of an RSPI receive buffer full interrupt request.

### 31.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPI0.SSLP 0008 8381h

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 should not be performed.

### 31.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 0008 8382h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b0	SPLP	RSPI Loopback	0: Normal mode 1: Loopback mode (data is inverted for transmission)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (data is not inverted for transmission)	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 MOSIA pin during MOSI idling corresponds to low 1: The level output on the MOSIA pin during MOSI idling corresponds to high	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 should not be performed.

#### 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 MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects (inverts) 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 MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA 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 MOSIA pin output value during the SSL negation period (including the SSL retention period during a burst transfer).

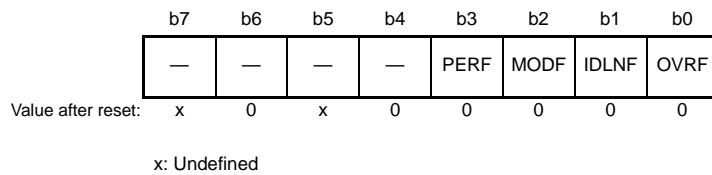
#### MOIFE Bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSIA 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 MOSIA pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSIA pin.



### 31.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 0008 8383h



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 the next serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer is full.

[Clearing condition]

- When SPSR is read while the OVRF flag is 1, and then 0 is written to the OVRF flag.

#### IDLNF Flag (RSPI Idle Flag)

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

Master mode

- Condition 1 and condition 2 are not satisfied in master mode under the [Clearing condition] below.

Slave mode

- The SPCR.SPE bit is 1 (RSPI function is enabled)

[Clearing condition]

Master mode

- The following 1 is satisfied (condition 1) or all of the following 2 to 4 are satisfied (condition 2).
  - The SPCR.SPE bit is 0 (RSPI is initialized)
  - The transmit buffer (SPTX) is empty (data for the next transfer is not set)
  - The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
  - 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 SSLAi 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 SSLAi 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 SSLAi signal is determined by the SSLP.SSLiP bit (SSLi signal polarity setting bit).

[Clearing condition]

- When SPSR is read while the MODF flag is 1, and then 0 is written to the MODF flag

**PERF Flag (Parity Error Flag)**

Indicates the occurrence of a parity error.

[Setting condition]

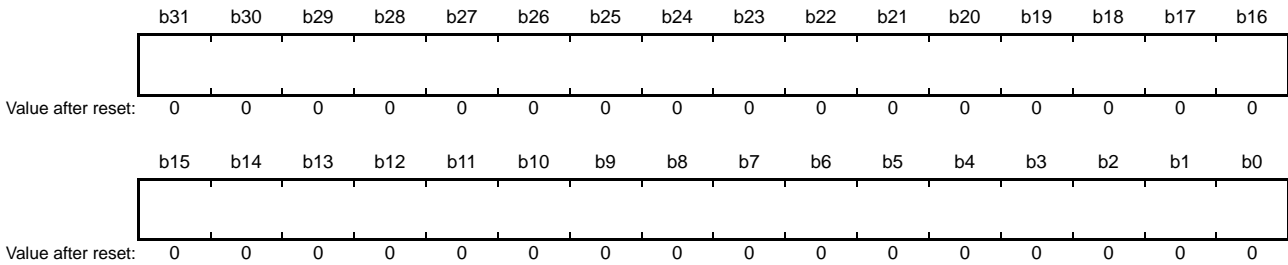
- When a serial transfer ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, the RSPI detects a parity error

[Clearing condition]

- When SPSR is read while the PERF flag is 1, and then 0 is written to the PERF flag

### 31.2.5 RSPI Data Register (SPDR)

Address(es): RSPI0.SPDR 0008 8384h



Address(es): RSPI0.SPDR 0008 8384h



SPDR is the interface with the buffers that hold data for transmission and reception by the RSPI.

When accessing in longwords (the SPLW bit is 1), access SPDR.

When accessing in words (the SPLW bit is 0), access the higher-order 16 bits (H) of SPDR.

The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR. Figure 31.2 shows the Configuration of SPDR.

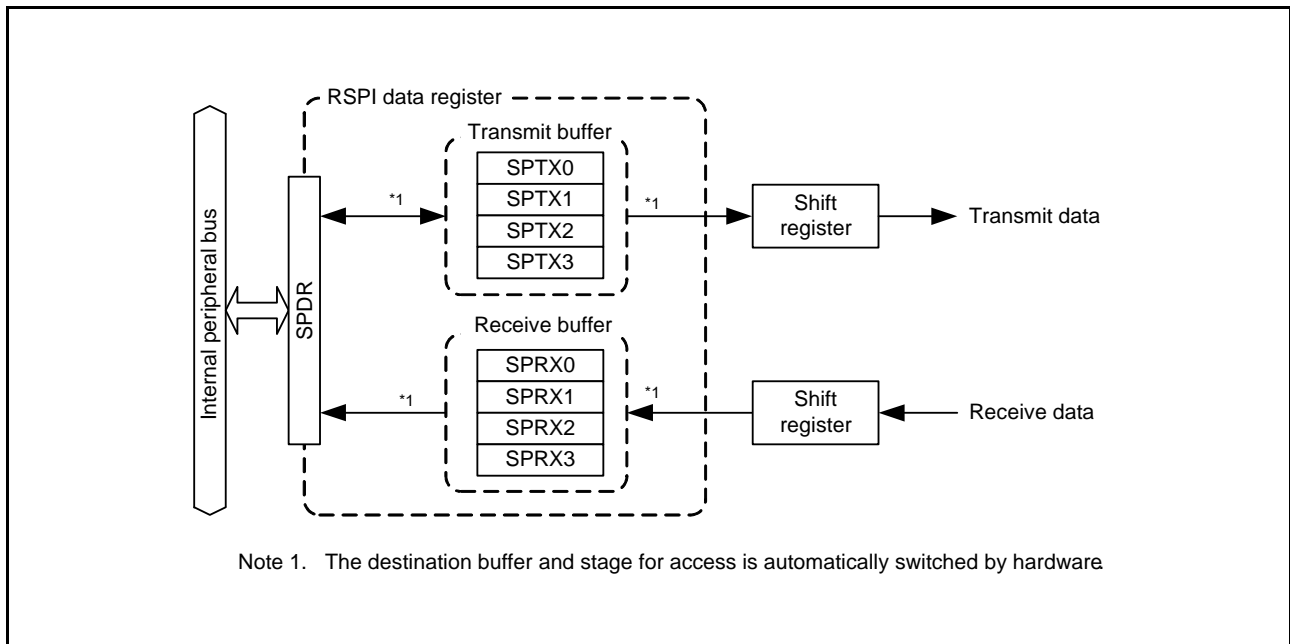


Figure 31.2 Configuration of SPDR

The transmit and receive buffers each have four stages. 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 (SPTX<sub>n</sub>) (n = 0 to 3) 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 SPTX<sub>n</sub> ( $n = 0$  to 3) are stored in the corresponding bits in SPRX<sub>n</sub>. For example, if the data length is 9 bits, received data are stored in the SPRX<sub>n</sub>[8:0] bits and the SPTX<sub>n</sub>[31:9] bits are stored in the SPRX<sub>n</sub>[31:9] bits.

### (1) Bus Interface

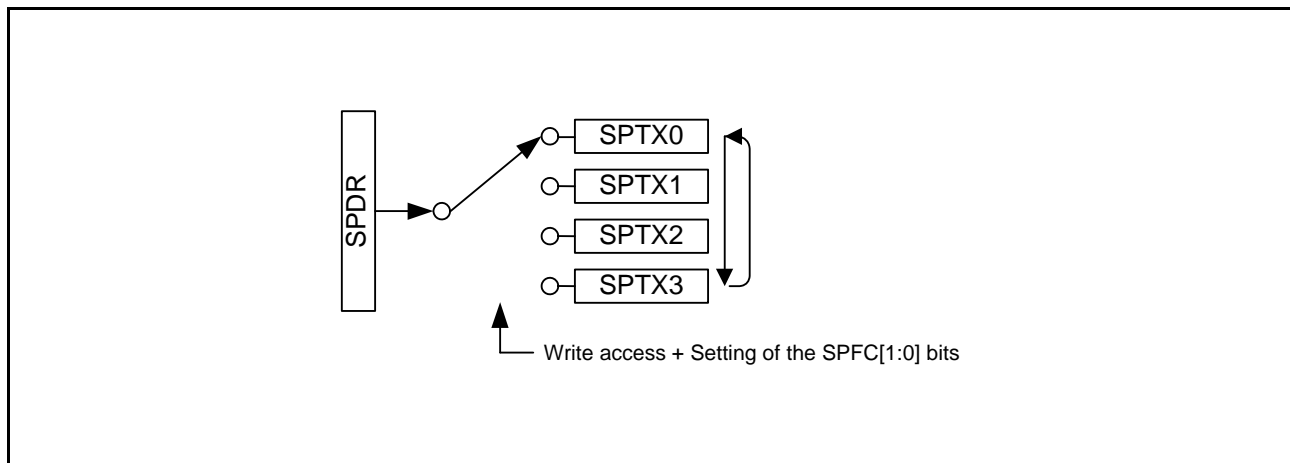
SPDR is the interface with 32-bit wide transmit and receive buffers, each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the 4-byte address space for SPDR. Furthermore, the unit of access for SPDR is selected by the RSPI longword access/word access specification bit in the RSPI data control register (SPDCR.SPLW). 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

Data written to SPDR are written to a transmit buffer (SPTX<sub>n</sub>). This is not influenced by the value of the SPDCR.SPRDTD bit unlike when reading from SPDR.

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 31.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to SPDR.



**Figure 31.3 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 SPTX3.
  - 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 transmit buffer (SPTX<sub>n</sub>) after generation of the transmit buffer empty interrupt, write the number of frames set by the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR). Even if the number of frames is written to the transmit buffer (SPTX<sub>n</sub>), the value of the buffer is not updated after completion of the writing and before generation of the next transmit buffer empty interrupt.

(b) Reading

SPDR can be read to read the value of a receive buffer (SPRX<sub>n</sub>) or a transmit buffer (SPTX<sub>n</sub>). The setting of the RSPI receive/transmit data select bit in the RSPI data control register (SPDCR.SPDRD) selects whether reading is of the receive or transmit buffer.

The sequence of reading the SPDR register is controlled by independent pointers, receive buffer read pointer and transmit buffer read pointer.

Figure 31.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from SPDR.

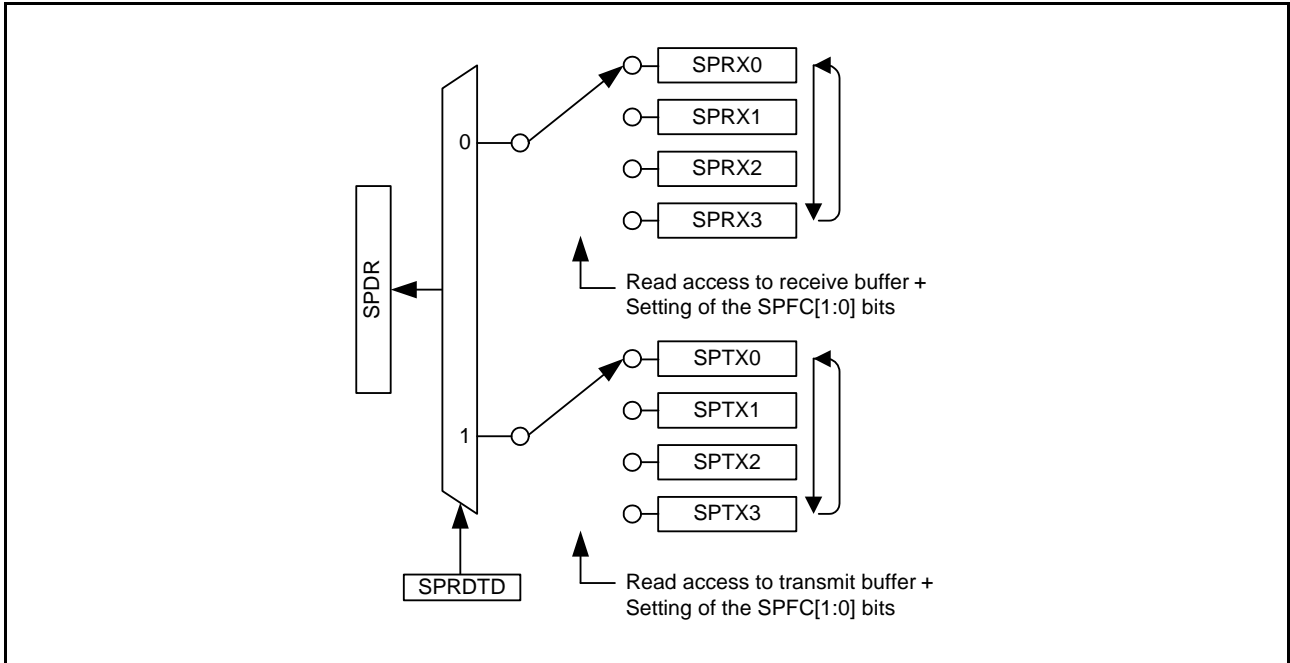


Figure 31.4 Configuration of SPDR (Reading)

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically.

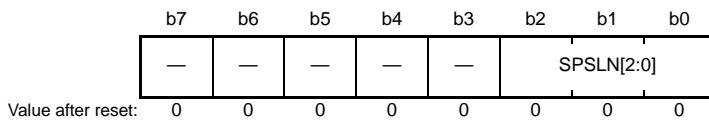
The sequence of switching the receive buffer read pointer is the same as that for the transmit buffer write pointer.

However, when 1 is written to the RSRI 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.

The transmit buffer read pointer is updated when writing to SPDR, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to SPDR is read. However, after generation of the transmit buffer empty interrupt, the values read from the transmit buffer are all 0 in the interval after completion of writing the number of frames of data specified in the number of frames specification bits (SPDCR.SPFC[1:0]) and before generation of the next buffer empty interrupt.

### 31.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 0008 8388h



Bit	Symbol	Bit Name	Description	R/W																																													
b2 to b0	SPSLN[2:0]	RSPI Sequence Length Specification	<table style="border: none; width: 100%;"> <tr> <td style="width: 10%;">b2</td> <td style="width: 10%;">b1</td> <td style="width: 10%;">b0</td> <td style="width: 10%;">Sequence Length</td> <td style="width: 50%;">Referenced SPCMD0 to SPCMD7 (No.)</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0→0→...</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>2</td> <td>0→1→0→...</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>3</td> <td>0→1→2→0→...</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>4</td> <td>0→1→2→3→0→...</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>5</td> <td>0→1→2→3→4→0→...</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>6</td> <td>0→1→2→3→4→5→0→...</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>7</td> <td>0→1→2→3→4→5→6→0→...</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>8</td> <td>0→1→2→3→4→5→6→7→0→...</td> </tr> </table> <p>The order in which the SPCMD0 to SPCMD7 registers are to be referenced is 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 references SPCMD0.</p>	b2	b1	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→...	R/W
b2	b1	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→...																																													
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.

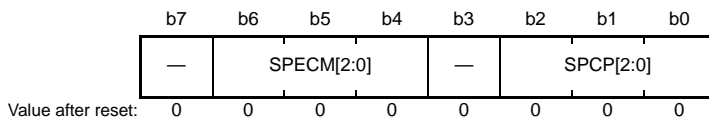
#### SPSSLN[2:0] Bits (RSPI Sequence Length Specification)

The SPSSLN[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 SPSSLN[2:0] bits.

In slave mode, SPCMD0 is referred.

### 31.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 0008 8389h



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, refer to section 31.3.10.1, 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 flags 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, refer to section 31.3.8, Error Detection. For the RSPI's sequence control, refer to section 31.3.10.1, Master Mode Operation.

### 31.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 0008 838Ah



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 should not be performed.

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).

$$\text{Bit rate} = \frac{f(\text{PCLK})}{2 \times (n + 1) \times 2^N}$$

Table 31.3 lists examples of the relationship among the SPBR settings, the BRDV[1:0] settings, and bit rates. Use the bit rate that meets electrical characteristics based on the AC specifications of the target device.

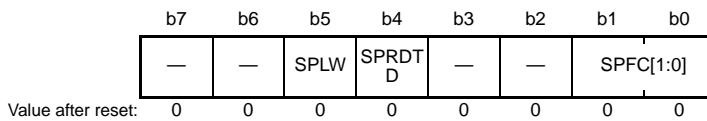
**Table 31.3 Relationship among SPBR Settings, BRDV[1:0] Settings, and Bit Rates**

SPBR (n)	BRDV[1:0] Bits (N)	Division Ratio	Bit Rate
			PCLK = 32 MHz
0	0	2	16.0 Mbps
1	0	4	8.00 Mbps
2	0	6	5.33 Mbps
3	0	8	4.00 Mbps
4	0	10	3.20 Mbps
5	0	12	2.67 Mbps
5	1	24	1.33 Mbps
5	2	48	667 kbps
5	3	96	333 kbps
255	3	4096	7.81 kbps



### 31.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 0008 838Bh



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 Select	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 SPSR.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 receive buffer full interrupt, and start of transmission or generation of transmit buffer empty interrupts. Table 31.4 lists 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 should not be performed.

**Table 31.4 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	000b	00b	1	1
1-2	000b	01b	2	2
1-3	000b	10b	3	3
1-4	000b	11b	4	4
2-1	001b	01b	2	2
2-2	001b	11b	4	4
3	010b	10b	3	3
4	011b	11b	4	4
5	100b	00b	5	1
6	101b	00b	6	1
7	110b	00b	7	1
8	111b	00b	8	1

**SPRDTD Bit (RSPI Receive/Transmit Data Select)**

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 transmit buffer empty interrupt.

For details, refer to section 31.2.5, RSPI Data Register (SPDR).

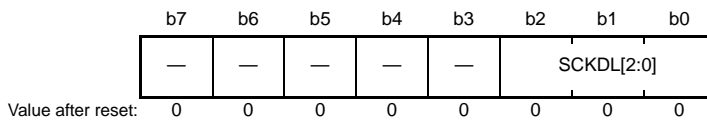
**SPLW Bit (RSPI Longword Access/Word Access Specification)**

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, operations should not be performed.

### 31.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 0008 838Ch



Bit	Symbol	Bit Name	Description	R/W																											
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	<table border="0"> <tr> <td>b2</td> <td>b0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0: 1 RSPCK</td> </tr> <tr> <td>0</td> <td>0</td> <td>1: 2 RSPCK</td> </tr> <tr> <td>0</td> <td>1</td> <td>0: 3 RSPCK</td> </tr> <tr> <td>0</td> <td>1</td> <td>1: 4 RSPCK</td> </tr> <tr> <td>1</td> <td>0</td> <td>0: 5 RSPCK</td> </tr> <tr> <td>1</td> <td>0</td> <td>1: 6 RSPCK</td> </tr> <tr> <td>1</td> <td>1</td> <td>0: 7 RSPCK</td> </tr> <tr> <td>1</td> <td>1</td> <td>1: 8 RSPCK</td> </tr> </table>	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
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																													
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 SSLAi 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 should not be performed.

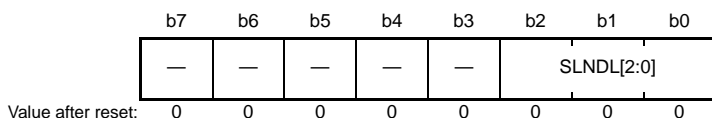
#### SCKDL[2:0] Bits (RSPCK Delay Setting)

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.

### 31.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 0008 838Dh



Bit	Symbol	Bit Name	Description	R/W																											
b2 to b0	SLNDL[2:0]	SSL Negation Delay Setting	<table border="0" style="font-size: small;"> <tr> <td>b2</td> <td>b0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0: 1 RSPCK</td> </tr> <tr> <td>0</td> <td>0</td> <td>1: 2 RSPCK</td> </tr> <tr> <td>0</td> <td>1</td> <td>0: 3 RSPCK</td> </tr> <tr> <td>0</td> <td>1</td> <td>1: 4 RSPCK</td> </tr> <tr> <td>1</td> <td>0</td> <td>0: 5 RSPCK</td> </tr> <tr> <td>1</td> <td>0</td> <td>1: 6 RSPCK</td> </tr> <tr> <td>1</td> <td>1</td> <td>0: 7 RSPCK</td> </tr> <tr> <td>1</td> <td>1</td> <td>1: 8 RSPCK</td> </tr> </table>	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
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																													
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W																											

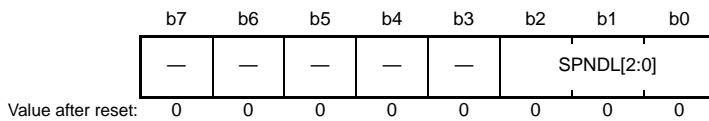
SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSLAi signal during a serial transfer by the RSPI in master mode. If the contents of SSLND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

#### 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.

### 31.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 0008 838Eh



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 SSLAi 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 should not be performed.

#### 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.

### 31.2.13 RSPI Control Register 2 (SPCR2)

Address(es): RSPI0.SPCR2 0008 838Fh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	PTE	SPIIE	SPOE	SPPE
0	0	0	0	0	0	0	0

Value after reset:

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-Diagnosis	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 or SPOE bit in SPCR2 is changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

#### 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 set to 0.

#### PTE Bit (Parity Self-Diagnosis)

The PTE bit enables the self-diagnosis function of the parity circuit in order to check whether the parity function is operating correctly.

## 31.2.14 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

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			
Value after reset:	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
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.SPSLN[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 should not be performed.

#### **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 (refer to section 31.2.8, 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 enables 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 SSLAi signal assertion when the RSPI performs serial transfers in master mode.

Setting the SSLA[2:0] bits controls the assertion for the SSLAi signal. When an SSLAi 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 SSL signals in the negated state (as the SSLA0 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 SSLAi 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.

Setting the SSLKP bit to 1 enables a burst transfer. For details, refer to section 31.3.10.1, Master Mode Operation (4) Burst Transfer.

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 SSLAi signal inactive until the RSPI enables the SSLAi 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 SSLAi 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 SSLAi 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.

### 31.3 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.

#### 31.3.1 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 31.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

**Table 31.5 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
RSPCKA signal	Input	Output	Output/Hi-Z	Input	Output
MOSIA signal	Input	Output	Output/Hi-Z	Input	Output
MISOA signal	Output/Hi-Z	Input	Input	Output	Input
SSLA0 signal	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSLA1 to SSLA3 signals	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity change 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				
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB				
Transfer data length	8 to 16, 20, 24, 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				
Receive buffer full detection	Supported*2				
Overrun error detection	Supported*2				
Parity error detection	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.

### 31.3.2 Controlling RSPI Pins

According to the MSTR, MODFEN, and SPMS bits in SPCR and the ODRn.Bi bit for I/O ports, the RSPI can switch pin states. Table 31.6 lists the relationship between pin states and bit settings. Setting the ODRn.Bi bit for an I/O port to 0 selects CMOS output; setting it to 1 selects open-drain output. The I/O port settings should follow this relationship.

**Table 31.6 Relationship between Pin States and Bit Settings**

Mode	Pin	Pin State*2	
		ODRn.Bi Bit for I/O Ports = 0	ODRn.Bi Bit for I/O Ports = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKA	CMOS output	Open-drain output
	SSLA0 to SSLA3	CMOS output	Open-drain output
	MOSIA	CMOS output	Open-drain output
	MISOA	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKA*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLA0	Input	Input
	SSLA1 to SSLA3*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIA*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISOA	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKA	Input	Input
	SSLA0	Input	Input
	SSLA1 to SSLA3*5	Hi-Z*1	Hi-Z*1
	MOSIA	Input	Input
	MISOA*4	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (Clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKA	CMOS output	Open-drain output
	SSLA0 to SSLA3*5	Hi-Z*1	Hi-Z*1
	MOSIA	CMOS output	Open-drain output
	MISOA	Input	Input
Slave mode (Clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKA	Input	Input
	SSLA0 to SSLA3*5	Hi-Z*1	Hi-Z*1
	MOSIA	Input	Input
	MISOA	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 SSLA0 is at the active level, the pin state is Hi-Z.

Note 4. When SSLA0 is at the non-active level or the SPCR.SPE bit is 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 31.7.

**Table 31.7 MOSI Signal Value Determination during SSL Negation Period**

MOIFE Bit	MOIFV Bit	MOSIA Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

### 31.3.3 RSPI System Configuration Examples

#### 31.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 31.5 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a master. In the single-master/single-slave configuration, the SSLA0 to SSLA3 output of this MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in a select state.\*1

This MCU (master) drives the RSPCKA and MOSIA. The SPI slave 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 SSLAi output of this MCU should be connected to the SSL input of the slave device.

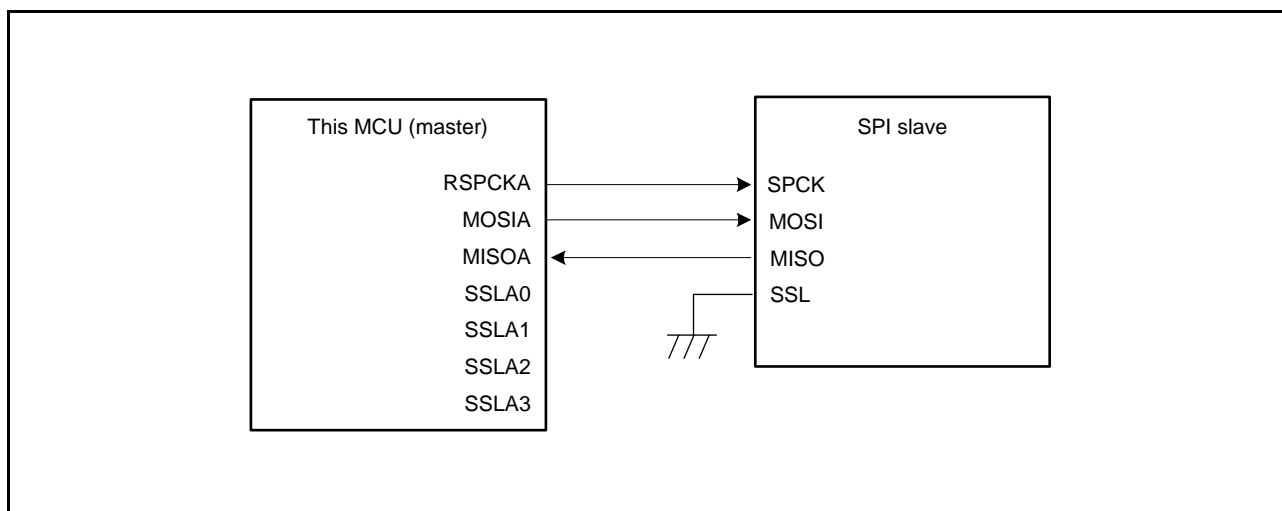


Figure 31.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)

### 31.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 31.6 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave, the SSLA0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI. This MCU (slave) drives the MISOA.\*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLA0 input of this MCU (slave) is fixed to the low level, this MCU (slave) is maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 31.7).

Note 1. When SSLA0 is at the non-active level, the pin state is Hi-Z.

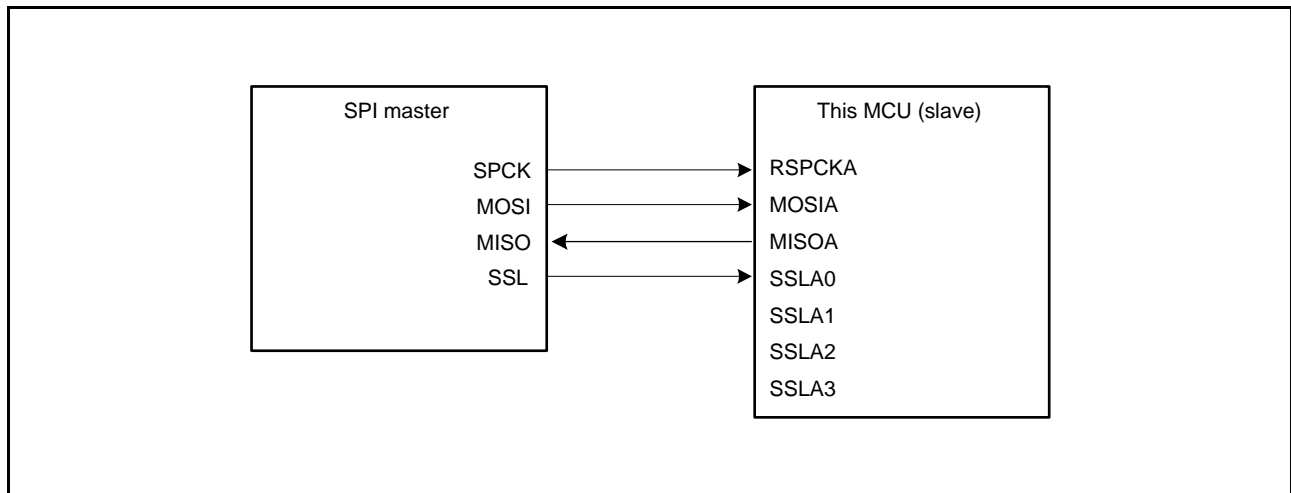


Figure 31.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)

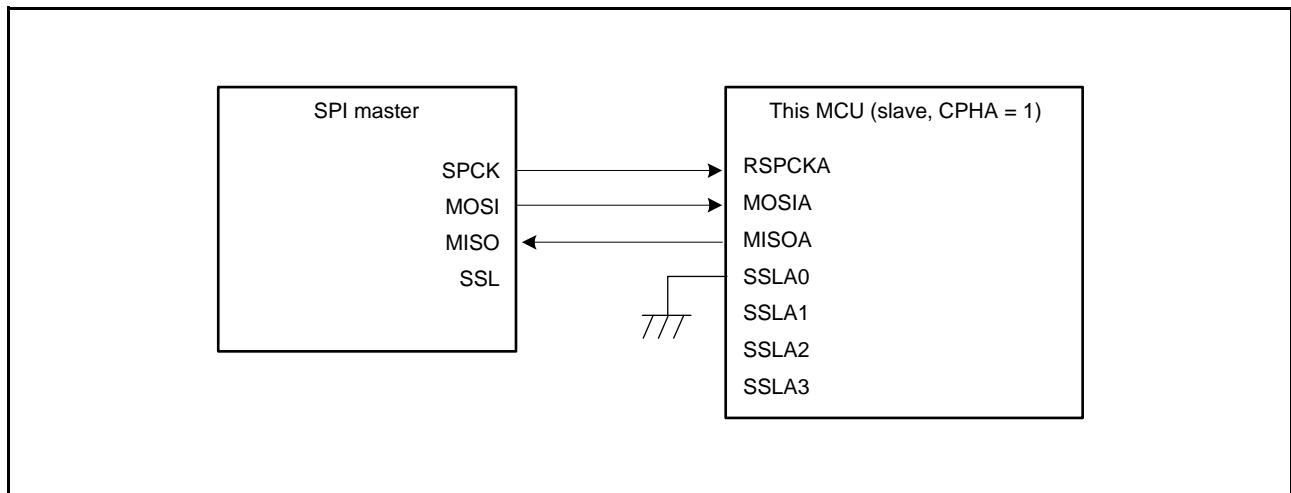


Figure 31.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)

### 31.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 31.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 31.8, the RSPI system is comprised of this MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKA and MOSIA outputs of this MCU (master) are connected to the RSPCK and MOSI inputs of SPI slave 0 to SPI slave 3. The MISO outputs of SPI slave 0 to SPI slave 3 are all connected to the MISOA input of this MCU (master). SSLA0 to SSLA3 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 3, respectively.

This MCU (master) drives RSPCK, MOSI, and SSLA0 to SSLA3. Of the SPI slave 0 to SPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

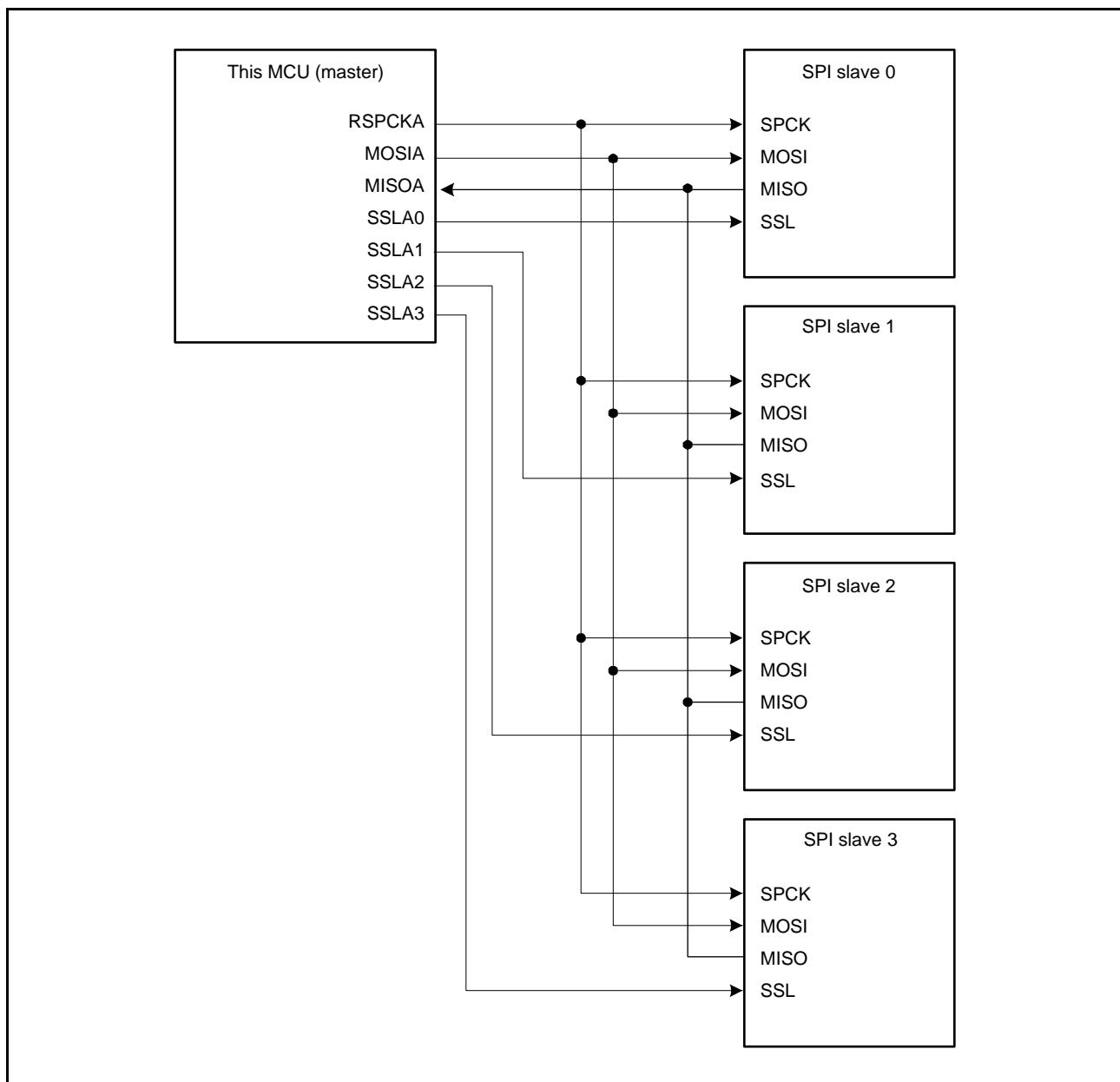


Figure 31.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)

### 31.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 31.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 31.9, the RSPI system is comprised of an SPI master and two MCUs (slave X and slave Y).

The SPCK and MOSI outputs of the SPI master are connected to the RSPCKA and MOSIA inputs of the MCUs (slave X and slave Y). The MISOA outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSLA0 inputs of the MCUs (slave X and slave Y), respectively.

The SPI master drives SPCK, MOSI, SSLX, and SSLY. Of the MCUs (slave X and slave Y), the slave that receives low-level input into the SSLA0 input drives MISOA.

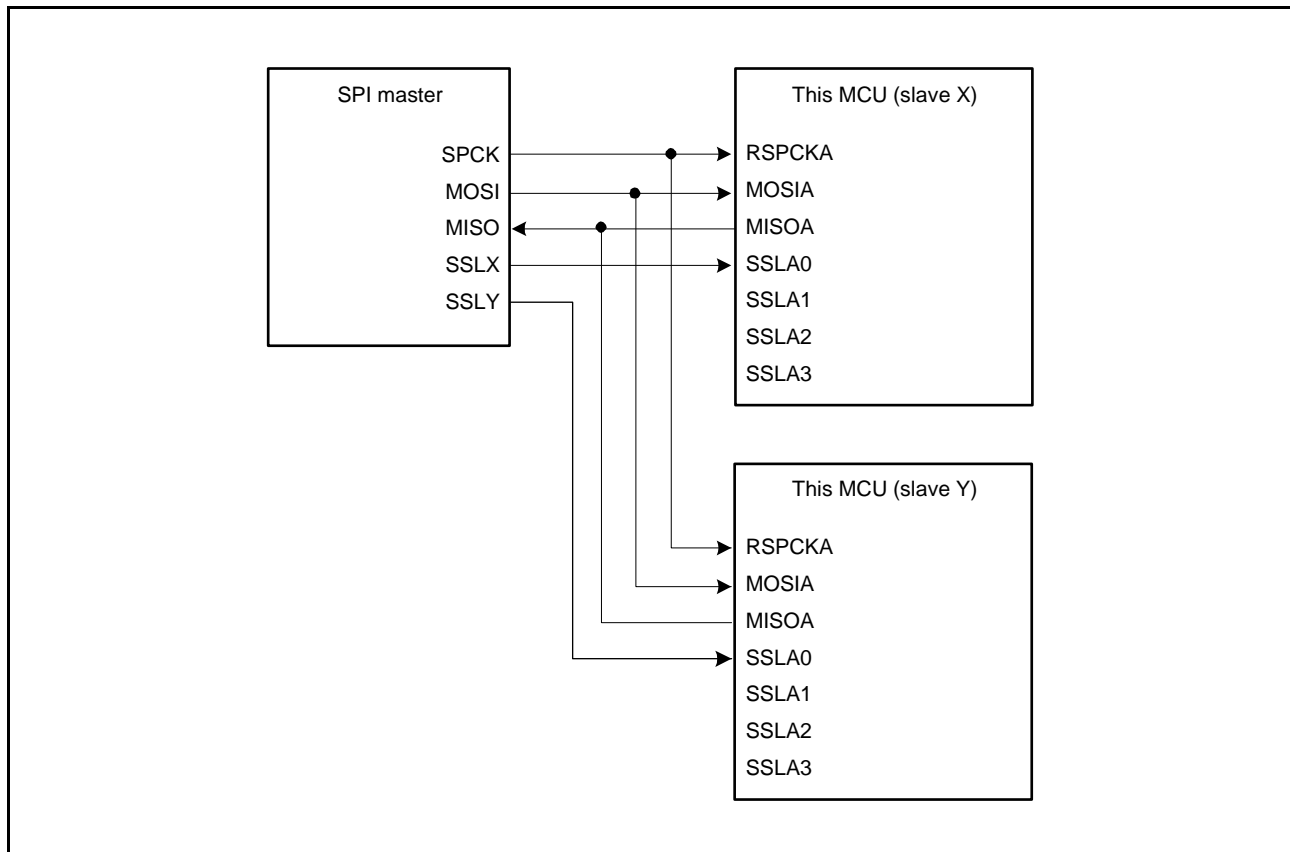


Figure 31.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)

### 31.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 31.10 shows a multi-master/multi-slave RSPi system configuration example when this MCU is used as a master. In the example of Figure 31.10, the RSPi system is comprised of two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCKA and MOSIA outputs of the MCUs (master X and master Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOA inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSLA0 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSLA0 input of this MCU (master X). The SSLA1 and SSLA2 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSLA0 input, and SSLA1 and SSLA2 outputs for slave connections, the SSLA3 output of this MCU is not required.

This MCU drives RSPCKA, MOSIA, SSLA1, and SSLA2 when the SSLA0 input level is high. When the SSLA0 input level is low, this MCU detects a mode fault error, sets RSPCKA, MOSIA, SSLA1, and SSLA2 to Hi-Z, and releases the RSPi bus right to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

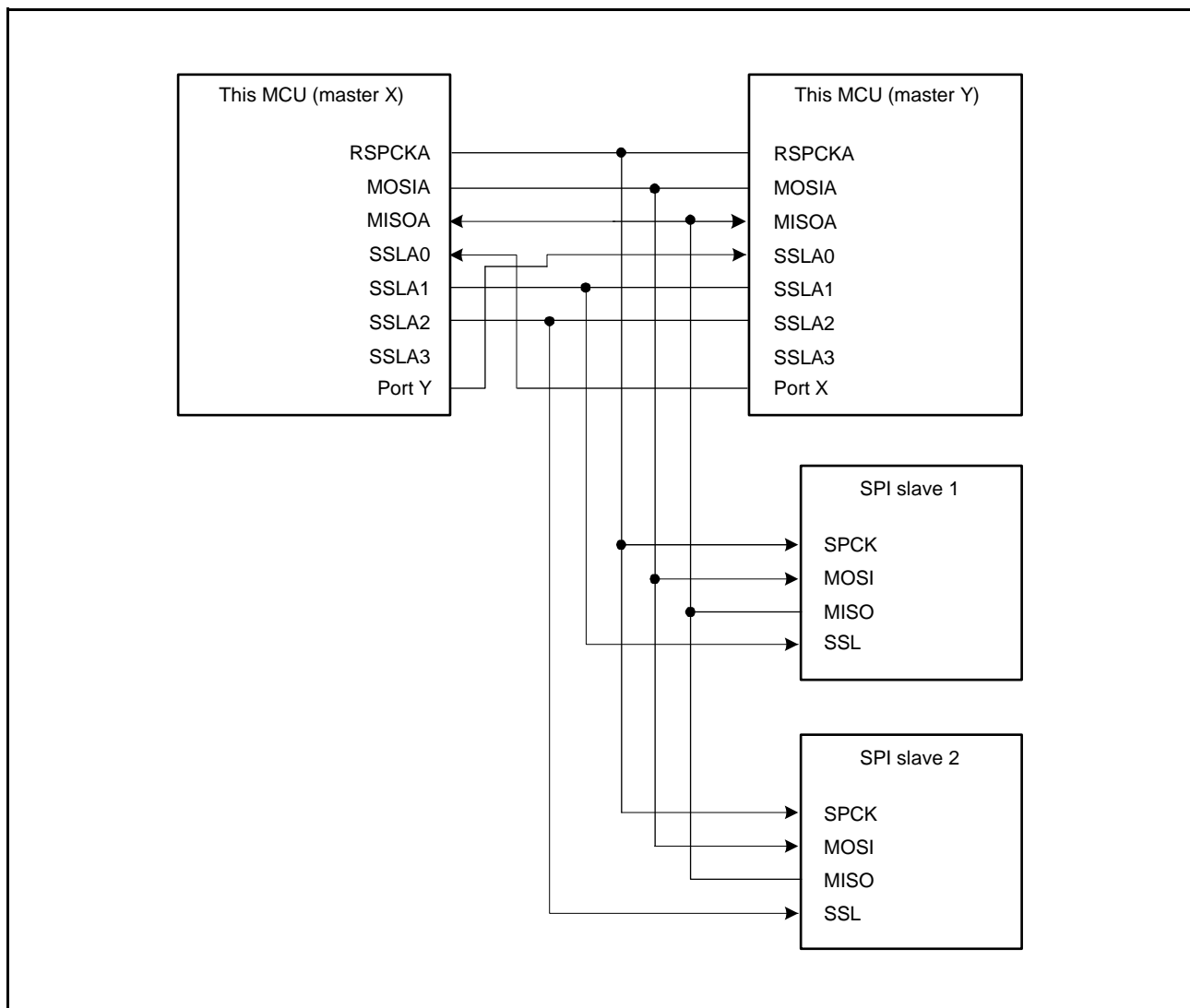


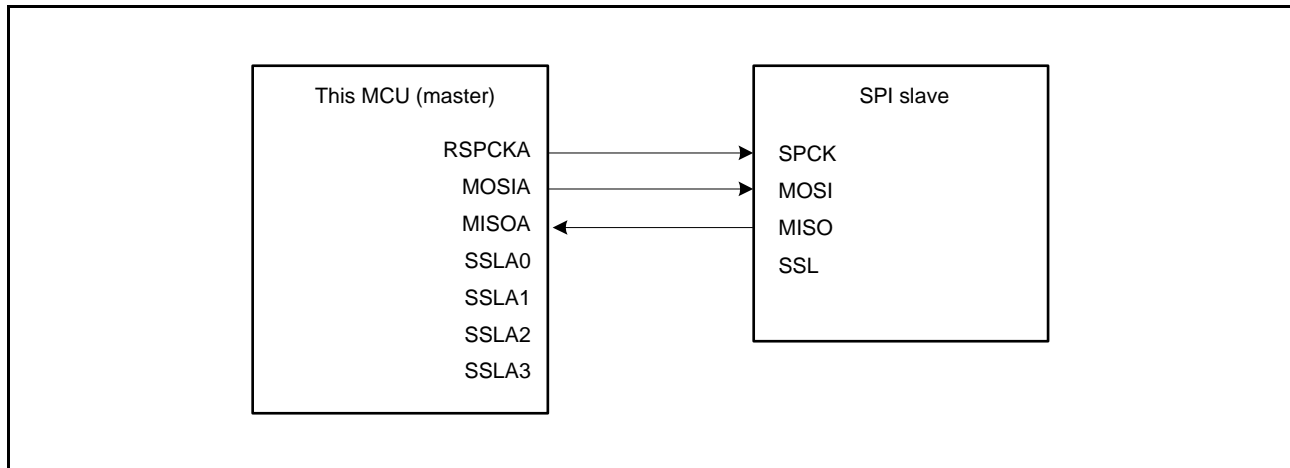
Figure 31.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)



### 31.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 31.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPi system configuration example when this MCU is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLA0 to SSLA3 of this MCU (master) are not used.

This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

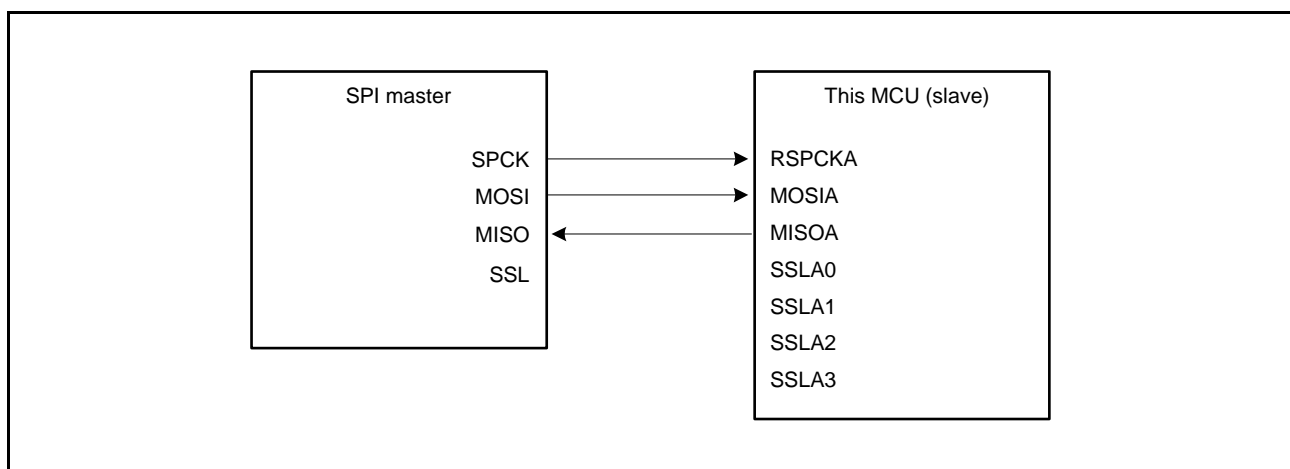


**Figure 31.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)**

### 31.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 31.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPi system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave (clock synchronous operation), this MCU (slave) drives the MISOA and the SPI master drives the SPCK and MOSI. In addition, SSLA0 to SSLA3 of this MCU (slave) are not used.

Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this MCU (slave) can execute serial transfer.



**Figure 31.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)**

### 31.3.4 Data Format

The RSPI's data format depends on the settings in RSPI command register  $m$  (SPCMD $m$ ) ( $m = 0$  to  $7$ ) and the parity enable bit in RSPI control register 2 (SPCR2.SPPE). Regardless of whether the MSB or LSB is first, the RSPI treats the range from the LSB bit in 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 RSPI command register  $m$  (SPCMD $m$ .SPB[3:0]).

#### (b) With Parity Enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register  $m$  (SPCMD $m$ .SPB[3:0]). In this case, however, the last bit is a parity bit.

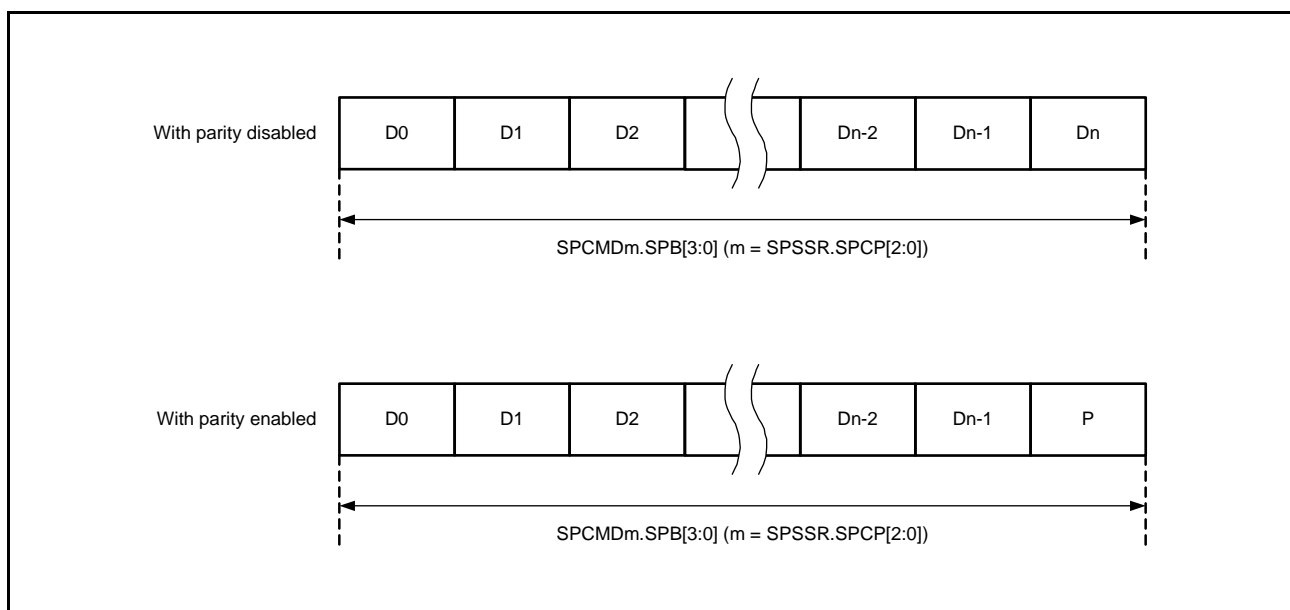


Figure 31.13 Outline of the Data Format (with Parity Disabled/Enabled)

### 31.3.4.1 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 31.14 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.

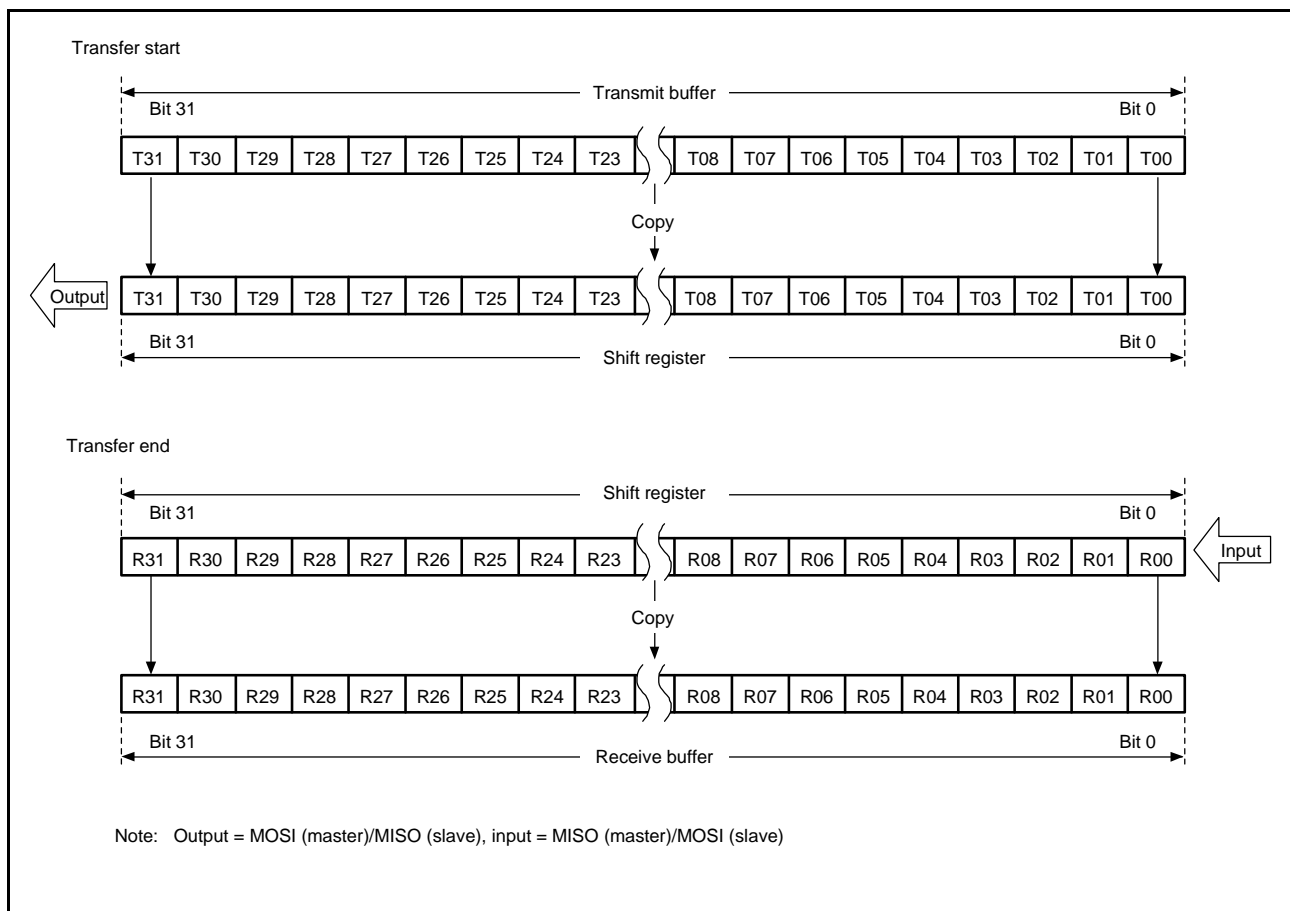


Figure 31.14 MSB First Transfer (32-Bit Data, Parity Disabled)

(2) MSB First Transfer (24-Bit Data)

Figure 31.15 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 8 bits of the transmit buffer are stored in the higher-order 8 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 8 bits of the receive buffer.

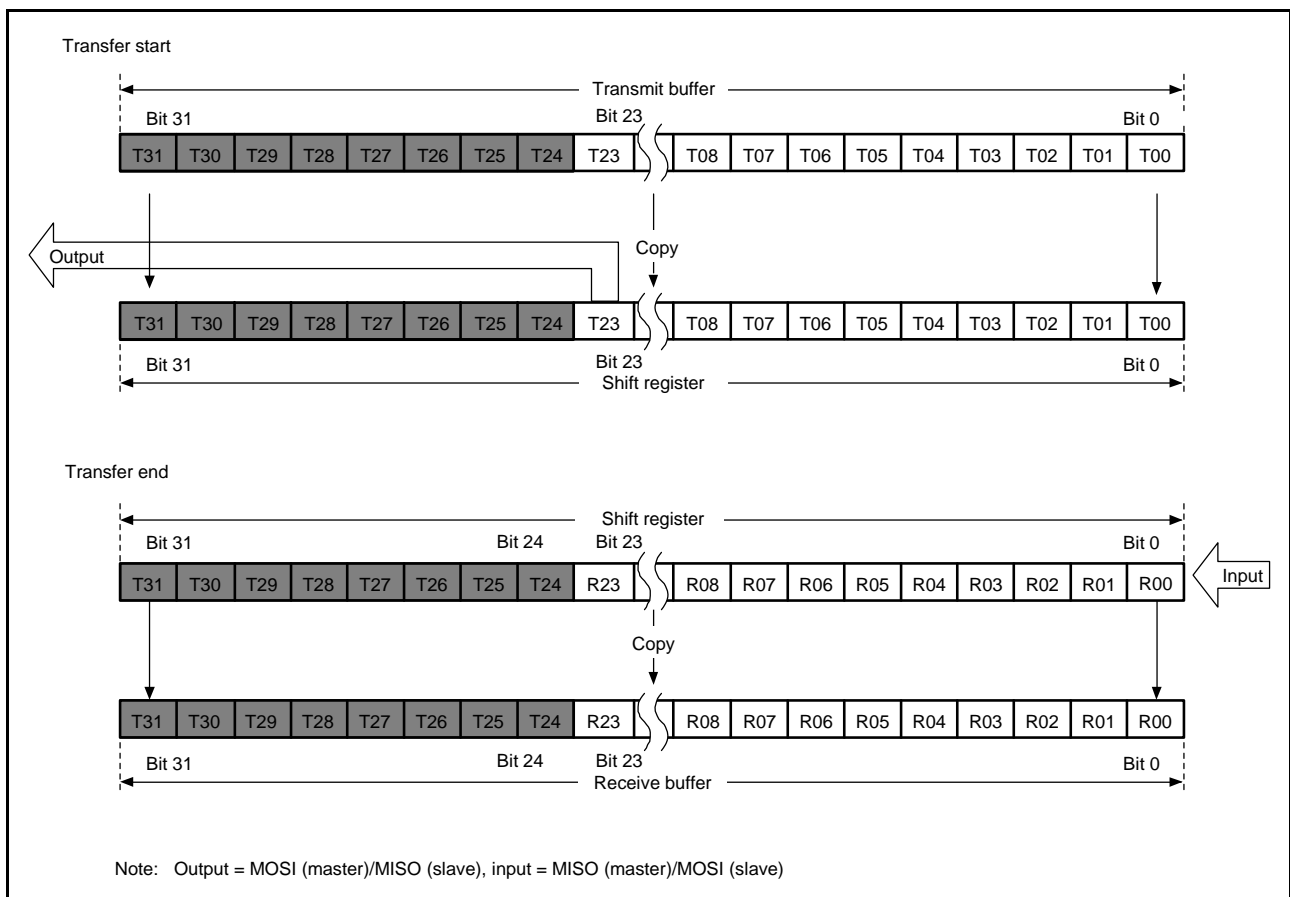


Figure 31.15 MSB First Transfer (24-Bit Data, Parity Disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 31.16 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.

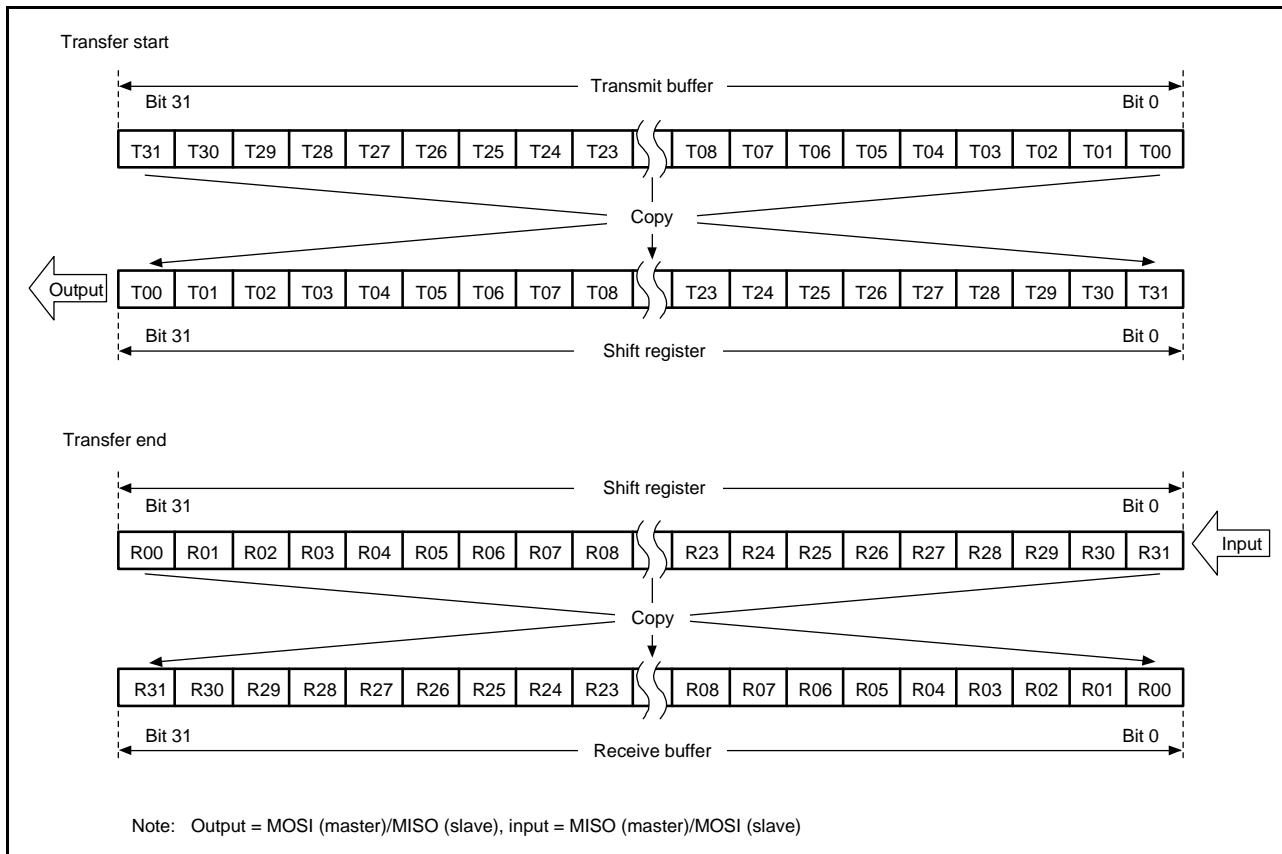


Figure 31.16 LSB First Transfer (32-Bit Data, Parity Disabled)

(4) LSB First Transfer (24-Bit Data)

Figure 31.17 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 8 bits of the transmit buffer are stored in the higher-order 8 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 8 bits of the receive buffer.

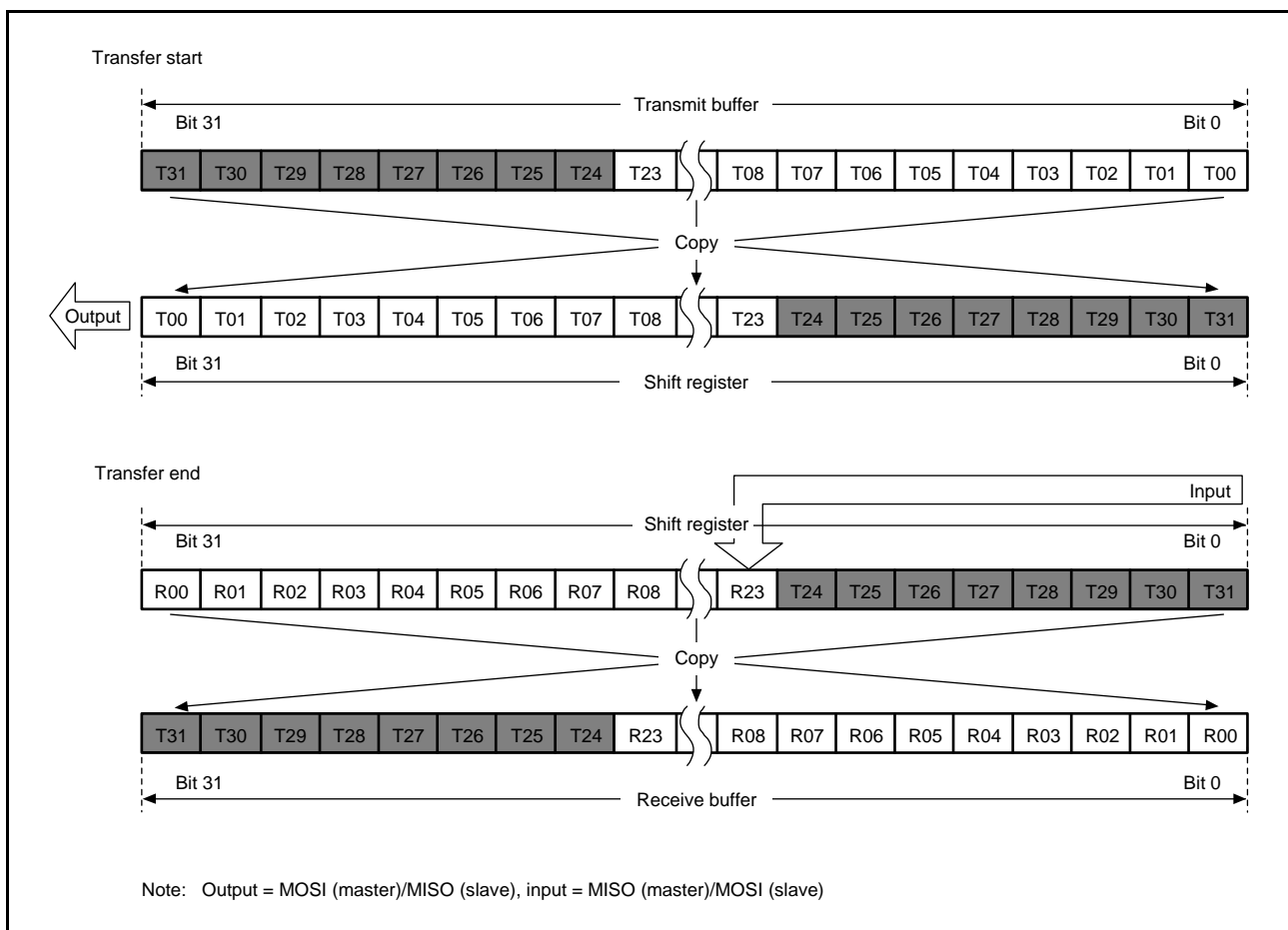


Figure 31.17 LSB First Transfer (24-Bit Data, Parity Disabled)

### 31.3.4.2 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 31.18 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.

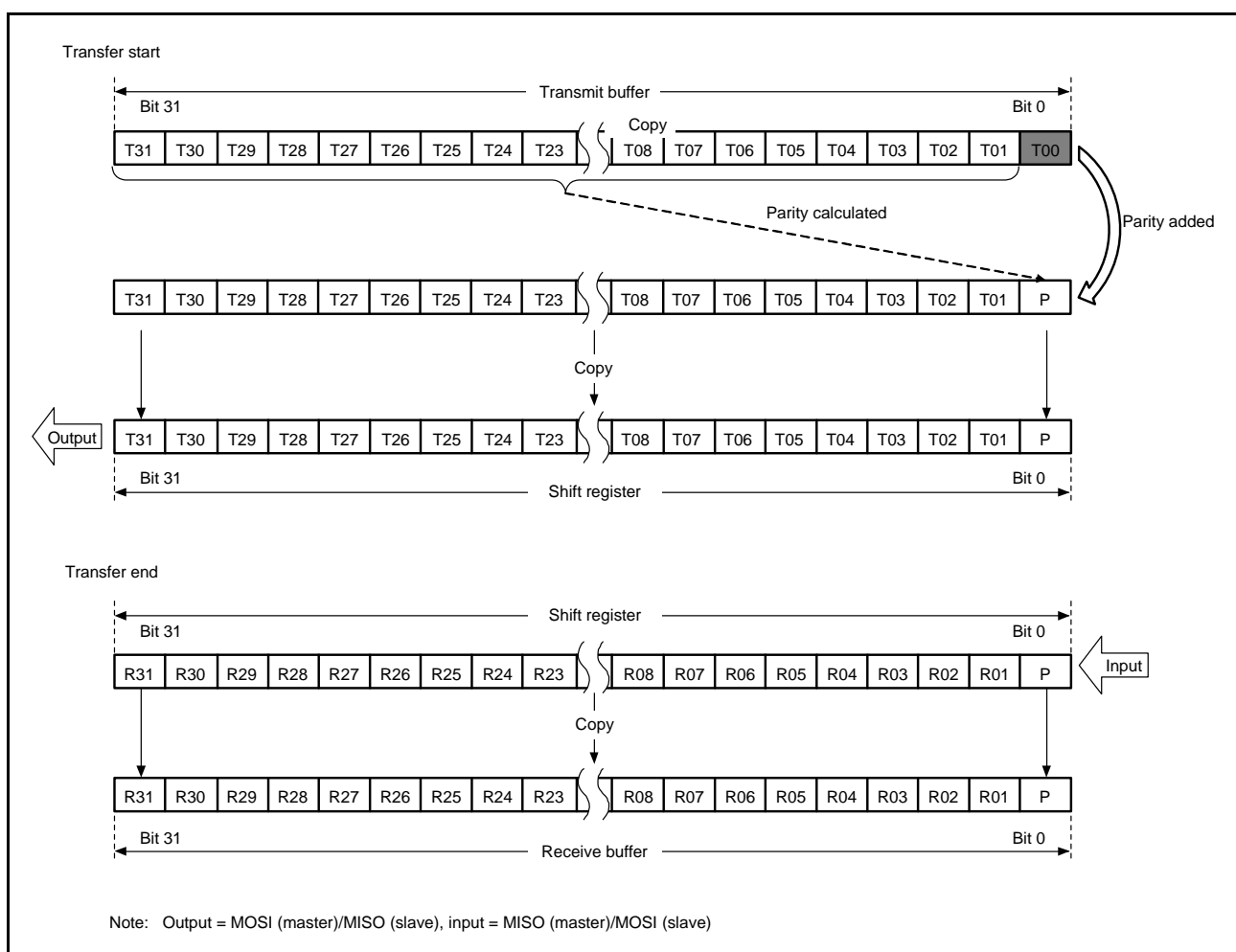


Figure 31.18 MSB First Transfer (32-Bit Data, Parity Enabled)

(2) MSB First Transfer (24-Bit Data)

Figure 31.19 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 8 bits of the transmit buffer are stored in the higher-order 8 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 8 bits of the receive buffer.

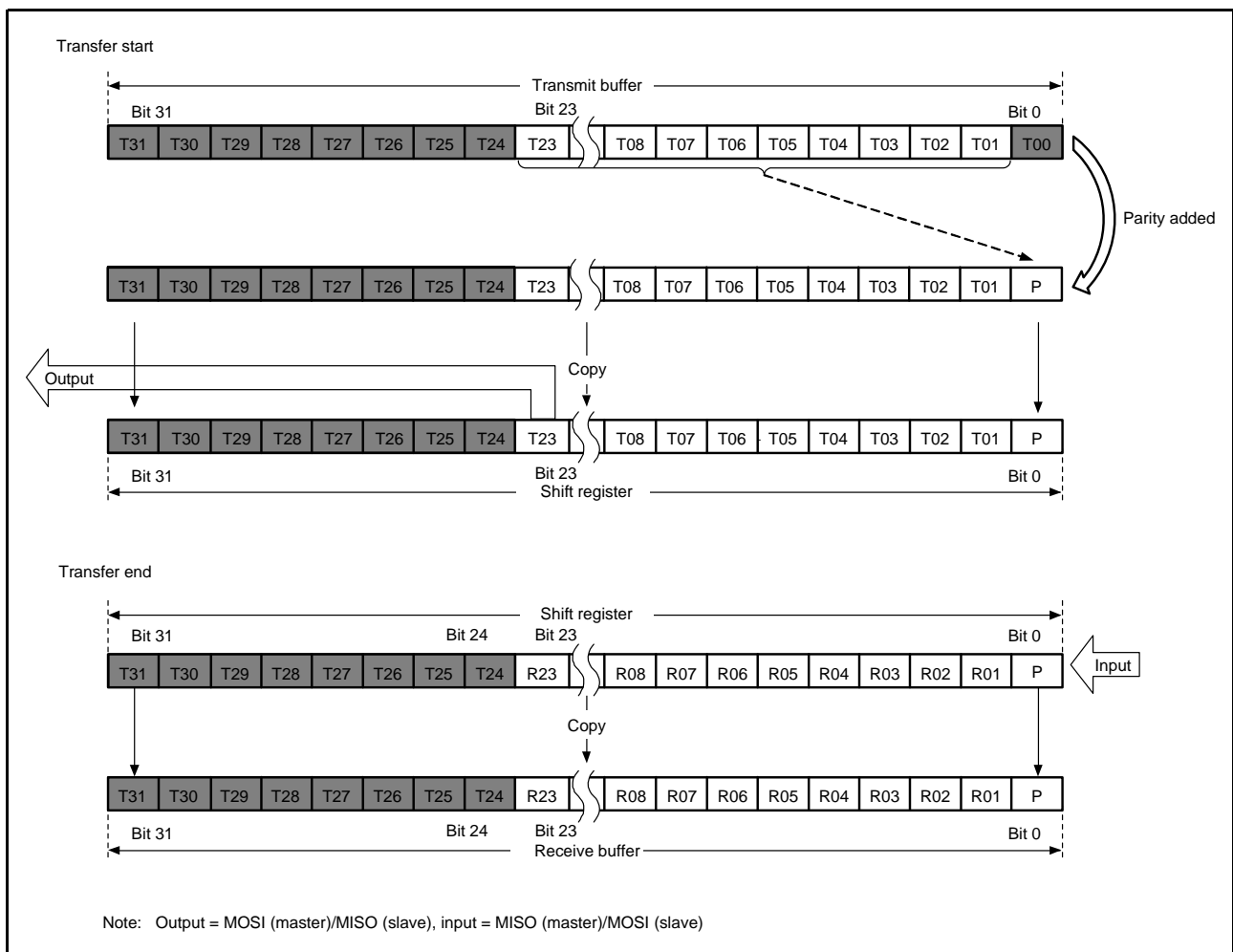


Figure 31.19 MSB First Transfer (24-Bit Data, Parity Enabled)



(3) LSB First Transfer (32-Bit Data)

Figure 31.20 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.

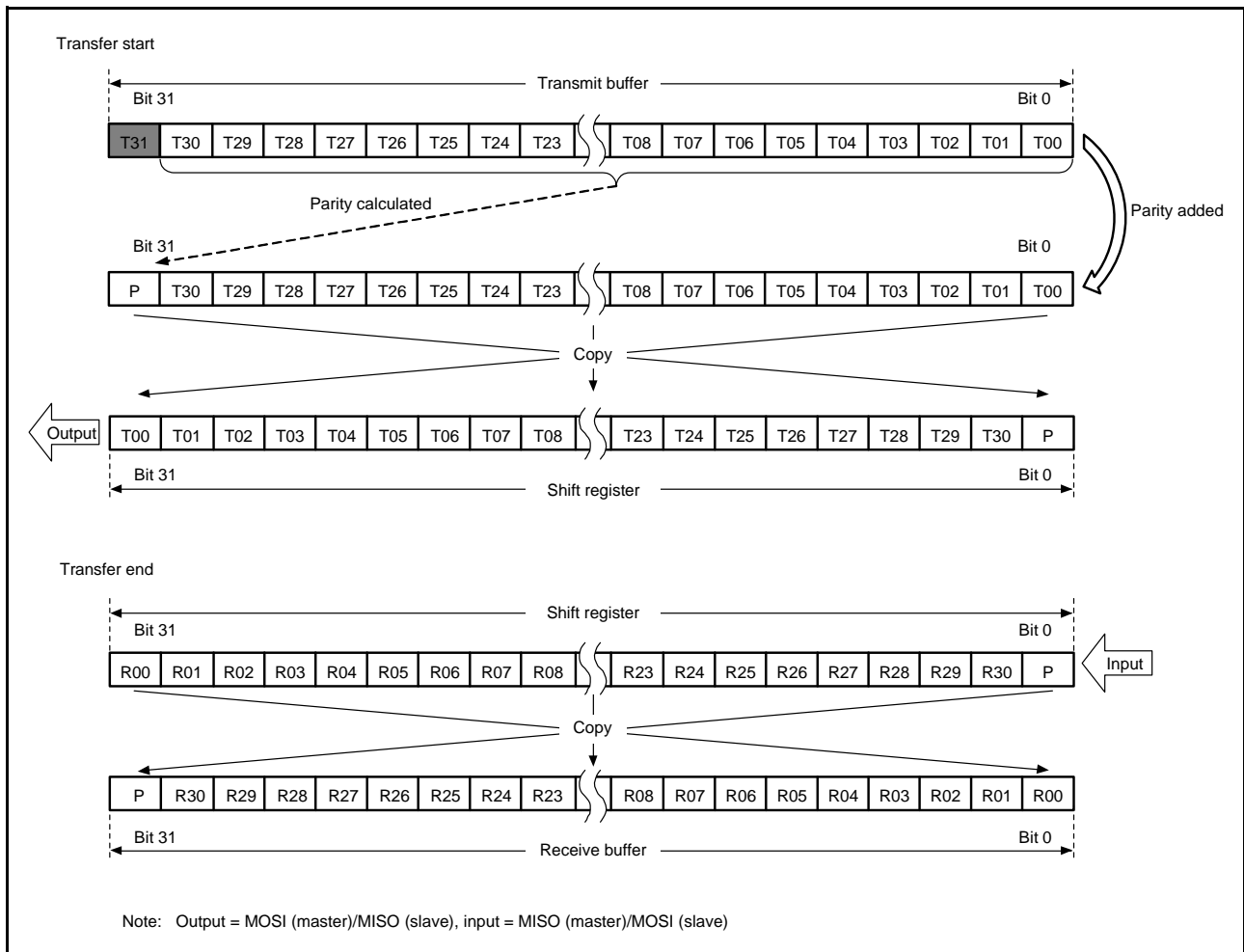


Figure 31.20 LSB First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 31.21 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 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. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 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 8 bits of the receive buffer.

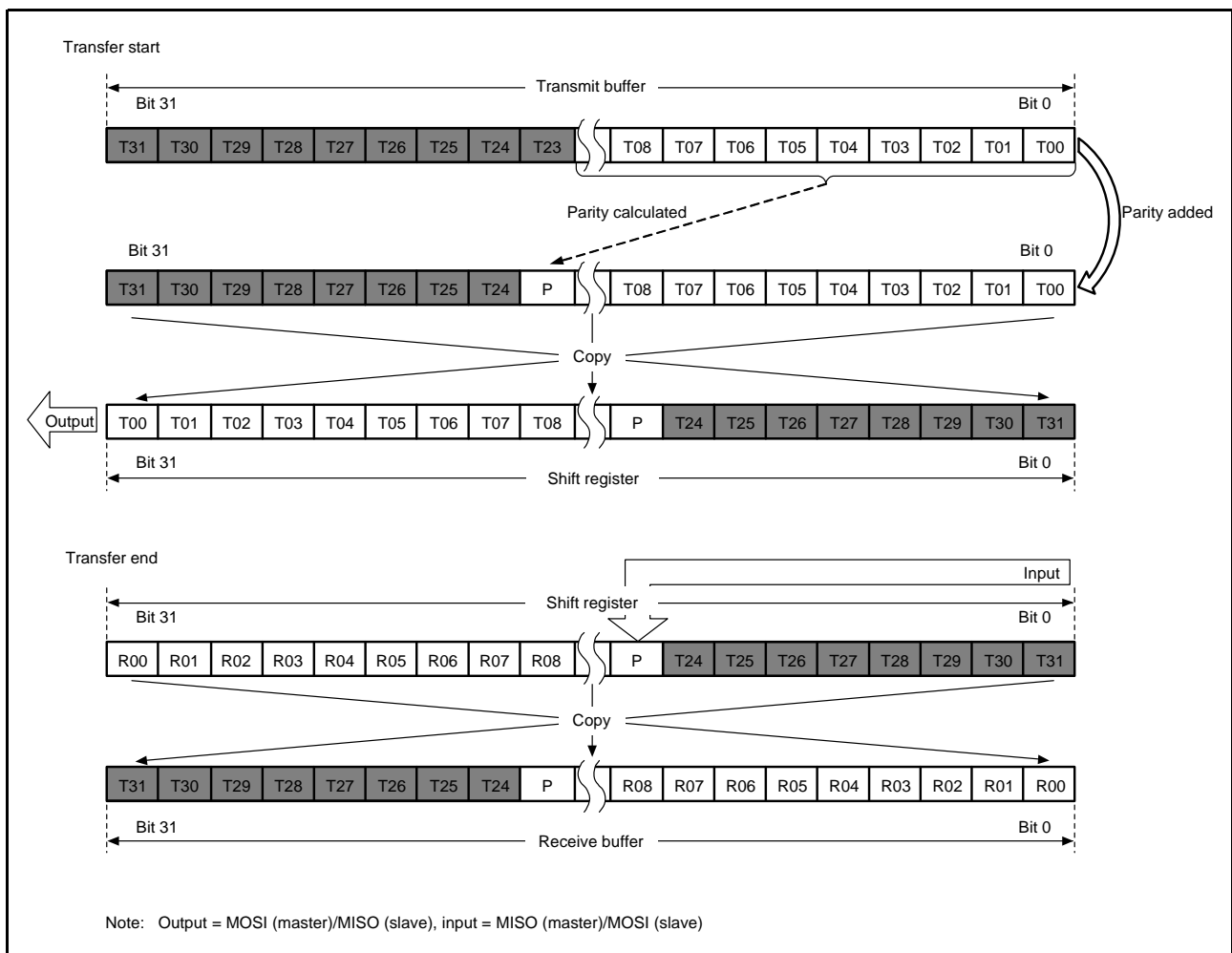


Figure 31.21 LSB First Transfer (24-Bit Data, Parity Enabled)

### 31.3.5 Transfer Format

#### 31.3.5.1 CPHA = 0

Figure 31.22 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) should not be performed when the RSPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 31.22, RSPCKA (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA 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 I/O directions of the signals depend on the RSPI settings. For details, refer to section 31.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIA and MISOA signals commences at an SSLAi signal assertion timing. The first RSPCKA signal change timing that occurs after the SSLAi signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIA and MISOA signals is 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 SSLAi signal assertion to RSPCKA oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKA oscillation to an SSLAi signal negation (SSL negation delay). t3 denotes a period in which SSLAi 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 MCU is in master mode, refer to section 31.3.10.1, Master Mode Operation.

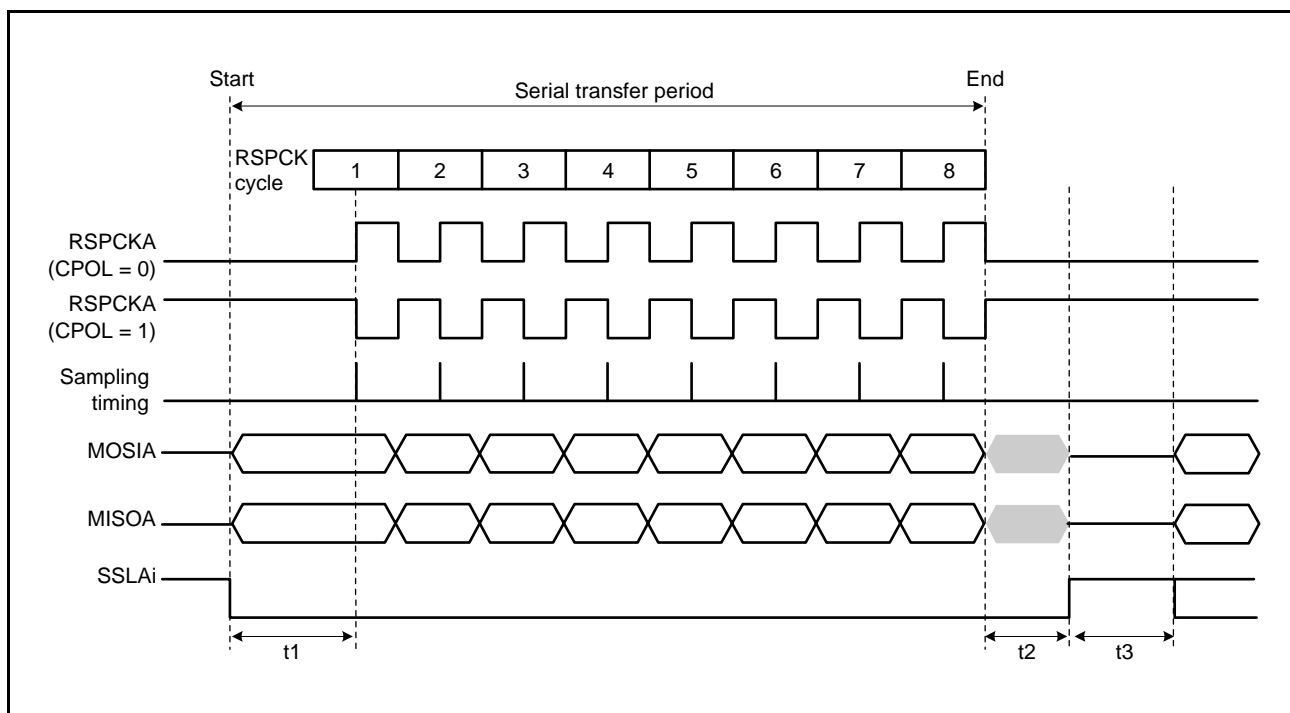


Figure 31.22 RSPI Transfer Format (CPHA = 0)

### 31.3.5.2 CPHA = 1

Figure 31.23 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 SSLAi signals are not used, and only the three signals RSPCKA, MOSIA, and MISOA handle communications. In Figure 31.23, RSPCK (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCK (CPOL = 1) indicates the RSPCKA 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 I/O directions of the signals depend on the RSPI mode (master or slave). For details, refer to section 31.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOA signal commences at an SSLAi signal assertion timing. The output of valid data to the MOSIA and MISOA signals commences at the first RSPCKA signal change timing that occurs after the SSLAi signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKA 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 MCU is in master mode, refer to section 31.3.10.1, Master Mode Operation.

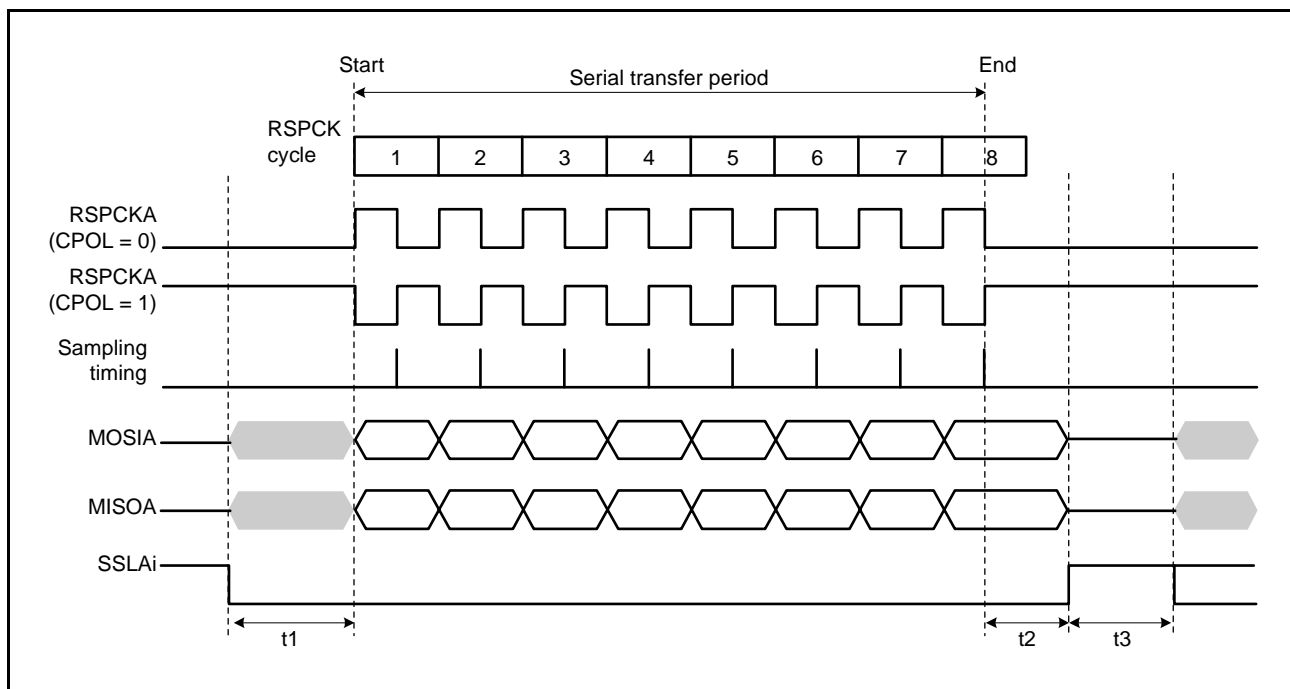


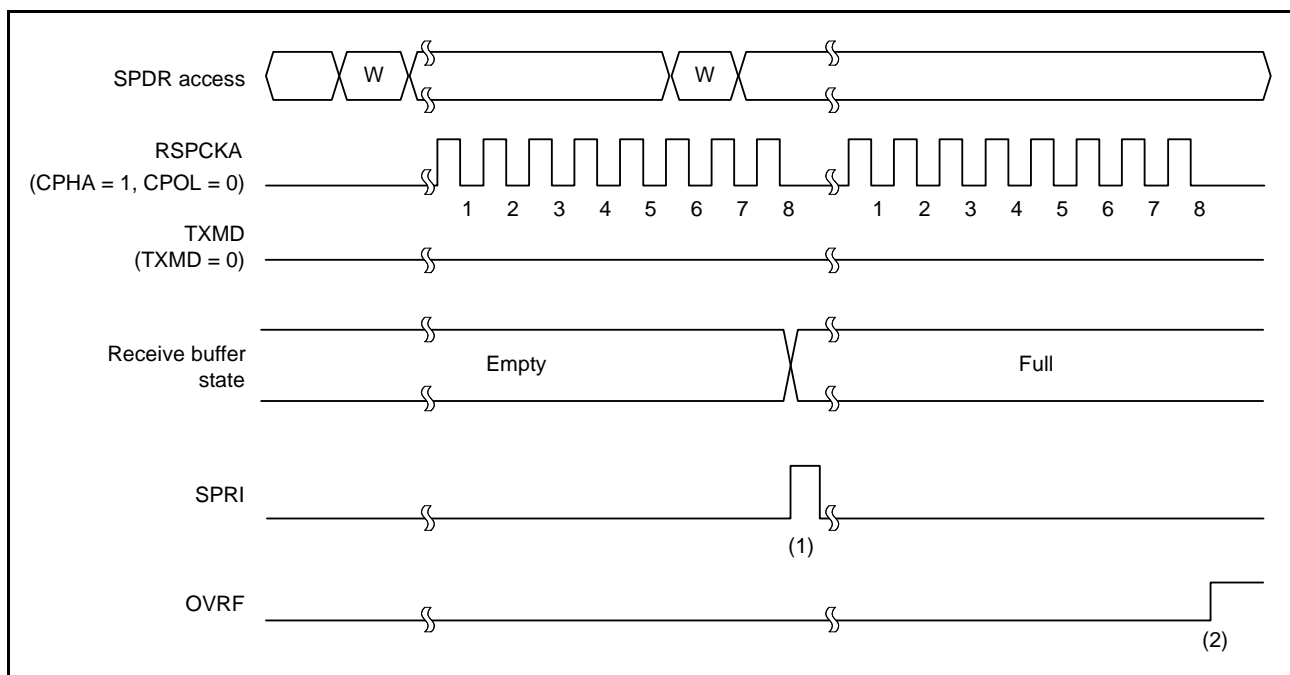
Figure 31.23 RSPI Transfer Format (CPHA = 1)

### 31.3.6 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 31.24 and Figure 31.25 indicate the condition of access to the SPDR register, where W denotes a write cycle.

#### 31.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)

Figure 31.24 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 0. In the example in Figure 31.24, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).



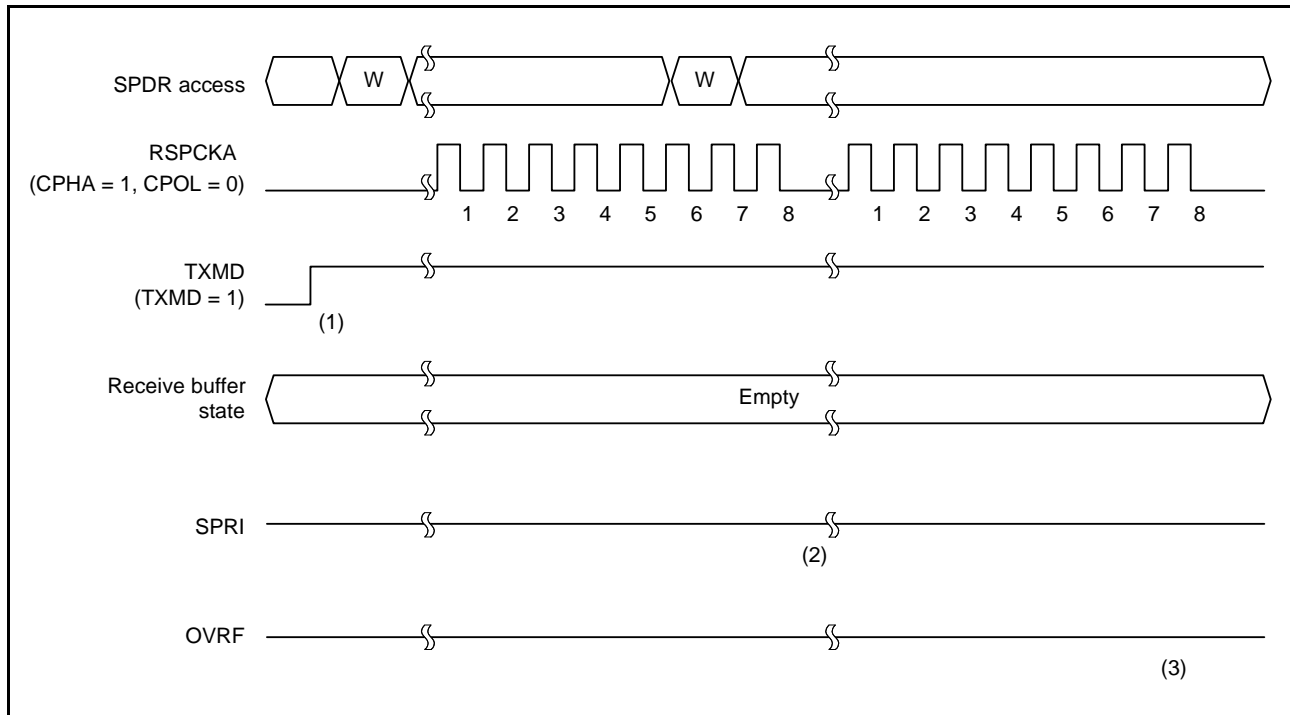
**Figure 31.24** Operation Example of SPCR.TXMD = 0

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.

### 31.3.6.2 Transmit Operations Only (SPCR.TXMD = 1)

Figure 31.25 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 1. In the example in Figure 31.25, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).



**Figure 31.25** Operation Example of SPCR.TXMD = 1

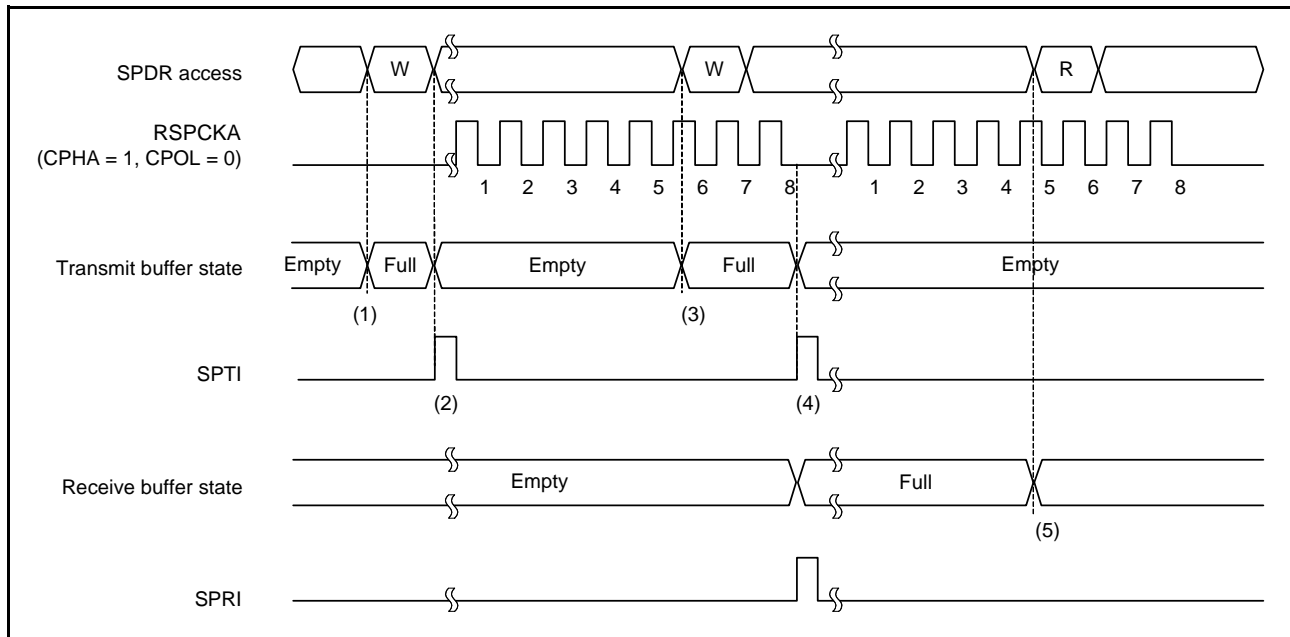
The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

- (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 0 at the timings of (1) to (3).

### 31.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 31.26 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). The SPDR register access shown in Figure 31.26 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 31.26, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).



**Figure 31.26 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, refer to section 31.3.10, SPI Operation, and section 31.3.11, 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 a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1.

If the RSPI function is disabled (the SPCR.SPE bit 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 (refer to section 31.3.8, 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 buffer full interrupt, set the SPCR.SPRIE bit to 1.

Transmission and reception interrupts or the corresponding IRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmission and reception buffers. Refer to section 14, Interrupt Controller (ICUb), for the interrupt vector numbers.



### 31.3.8 Error Detection

In the normal RSPI serial transfer, the data written to the transmit buffer of SPDR is transmitted, and the 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 31.8 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

**Table 31.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function**

	Occurrence Condition	RSPI Operation	Error Detection
1	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
2	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 transmitted.	None
3	SPDR is read when the receive buffer is empty.	Previously received data is output.	None
4	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 receive data.</li> </ul>	Overrun error
5	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
6	The SSLA0 input signal is asserted when the serial transfer is idle in multi-master mode.	<ul style="list-style-type: none"> <li>Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped.</li> <li>RSPI function is disabled.</li> </ul>	Mode fault error
7	The SSLA0 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 RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped.</li> <li>RSPI function is disabled.</li> </ul>	Mode fault error
8	The SSLA0 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 MISOA output signal is stopped.</li> <li>RSPI function is disabled.</li> </ul>	Mode fault error

On operation 1 described in Table 31.8, 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 buffer empty interrupt request.

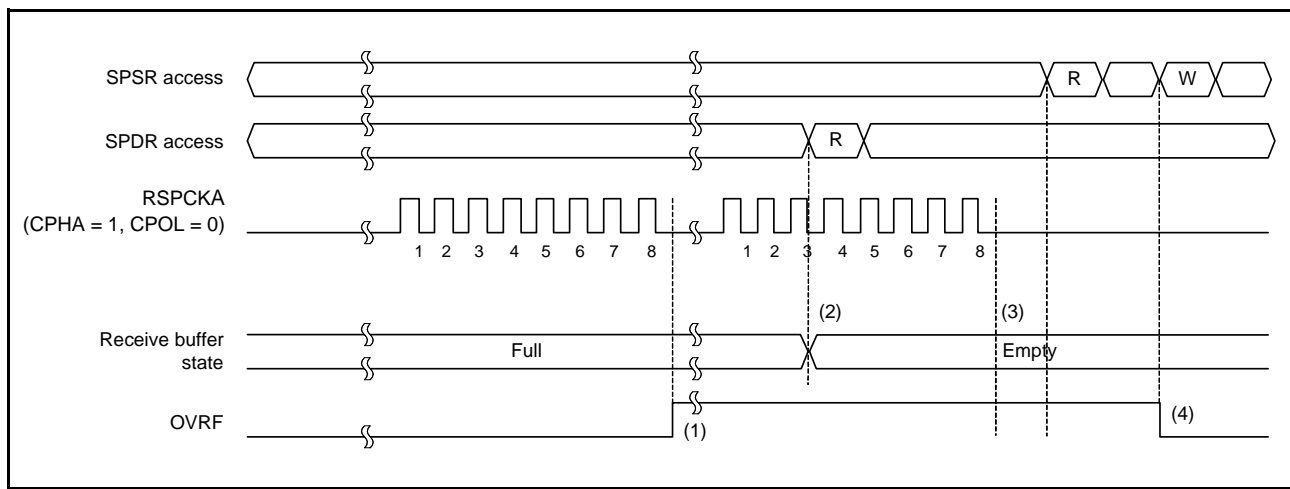
Likewise, the RSPI does not detect an error on operation 2. In a serial transfer that was started before the shift register was updated, the RSPI sends the data that was received in the previous serial transfer, and does not treat the operation indicated in 2 as an error. Note that the received data 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 3. To prevent extraneous data from being read, SPDR read operation should be executed using an RSPI receive buffer full interrupt request.

An overrun error shown in 4 is described in section 31.3.8.1, **Overrun Error**. A parity error shown in 5 is described in section 31.3.8.2, **Parity Error**. A mode fault error shown in 6 to 8 is described in section 31.3.8.3, **Mode Fault Error**. For the transmit and receive interrupts, refer to section 31.3.7, **Transmit Buffer Empty/Receive Buffer Full Interrupts**.

### 31.3.8.1 Overrun Error

If a serial transfer ends when the receive buffer of SPDR is full, the RSPI detects an overrun error, and sets the SPSR.OVRF flag 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 31.27 shows an example of operation of the OVRF flag. The SPSR and SPDR accesses shown in Figure 31.27 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 31.27, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).



**Figure 31.27 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 set the OVRF flag to 0.
3. If the serial transfer ends with the OVRF flag being 1 (an overrun error occurs), the RSPI does not copy the data in the shift register to the receive buffer. A receive buffer full 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 0 is written to the OVRF flag after SPSR is read when the OVRF flag is 1, the OVRF flag is set to 0.

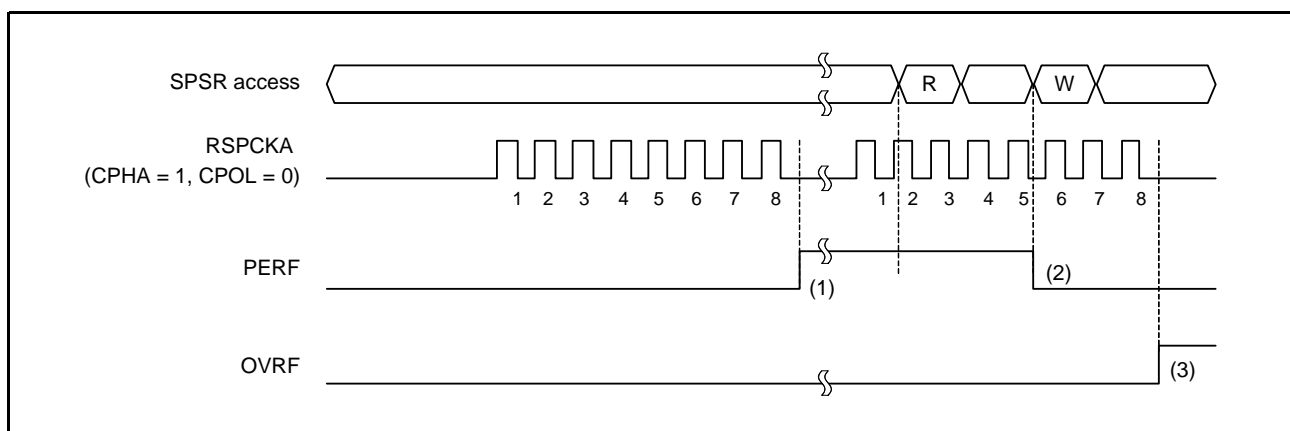
The occurrence of an overrun can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When executing a serial transfer, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read. When the RSPI is used 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 set to 0.

### 31.3.8.2 Parity Error

If full-duplex synchronous serial communications is performed with the SPCR.TXMD bit set 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 31.28 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 31.28 indicates the condition of access to SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 31.28, 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 RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).



**Figure 31.28** 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 0 is written to the PERF flag after SPSR register is read when the PERF flag is 1, the PERF flag is set 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 the SPSR register or by using an RSPI error interrupt and reading the 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.

### 31.3.8.3 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 SSLA0 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 SPSR.MODF flag 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 SSLA0 signal is determined by the SSLP.SSLOP bit.

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit of the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSLA0 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 (refer to section 31.3.9, 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 flag 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 flag must be set to 0.

### 31.3.9 Initializing RSPI

If 0 is written to the SPCR.SPE bit or the RSPI sets 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 SPCR.SPE bit and initialization by a system reset.

#### 31.3.9.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is set to 0, 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 SPSR.OVRF and SPSR.MODF flags 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 SPCR.SPTIE bit is set to 1 after RSPI initialization, a transmit buffer empty interrupt is generated. When the RSPI is initialized, in order to disable any transmit buffer empty interrupt, 0 should be written to the SPTIE bit simultaneously with the writing of 0 to the SPE bit. To disable any transmit buffer empty interrupt after a mode fault error is detected, use an error handling routine to write 0 to the SPTIE bit.

#### 31.3.9.2 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 31.3.9.1, Initialization by Clearing the SPE Bit.

### 31.3.10 SPI Operation

#### 31.3.10.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (refer to section 31.3.8, 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 transmit 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, refer to section 31.3.5, Transfer Format. The polarity of the SSLAi output pins depends on the SSLP register settings.

##### (2) Terminating a Serial Transfer

Irrespective of the SPCMDm.CPHA bit, the RSPI terminates the serial transfer after transmitting an RSPCKA 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 SPDR register.

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.SPB[3:0] bit setting. The polarity of the SSLAi output pin depends on the SSLP register settings.

For details on the RSPI transfer format, refer to section 31.3.5, Transfer Format.

### (3) Sequence Control

The transfer format that is employed in master mode is determined by SPSCR, SPCMDm, 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 SPCMDm register: SSLAi 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 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 SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 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 SPCMD0, and in this manner the sequence is executed repeatedly.

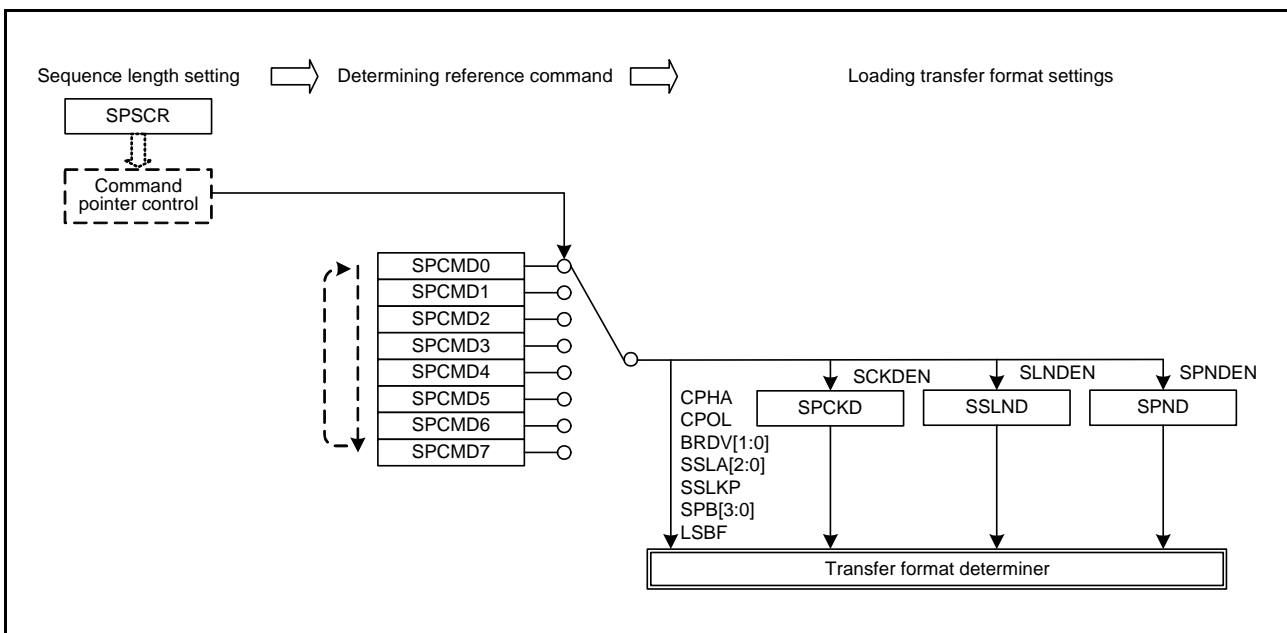


Figure 31.29 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

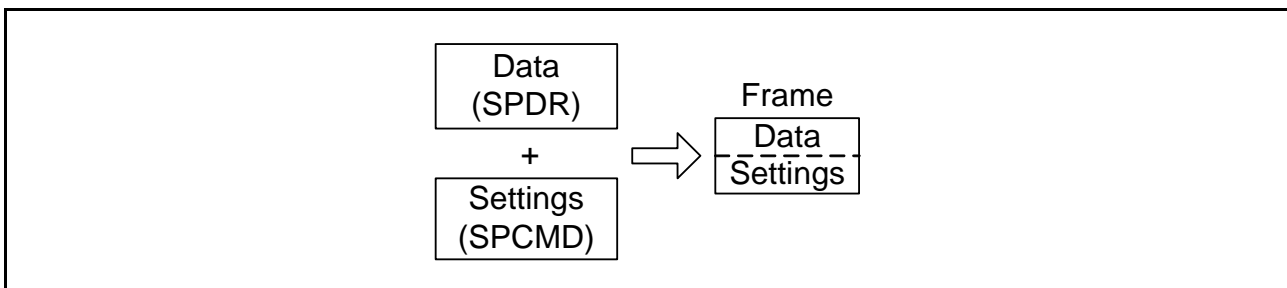
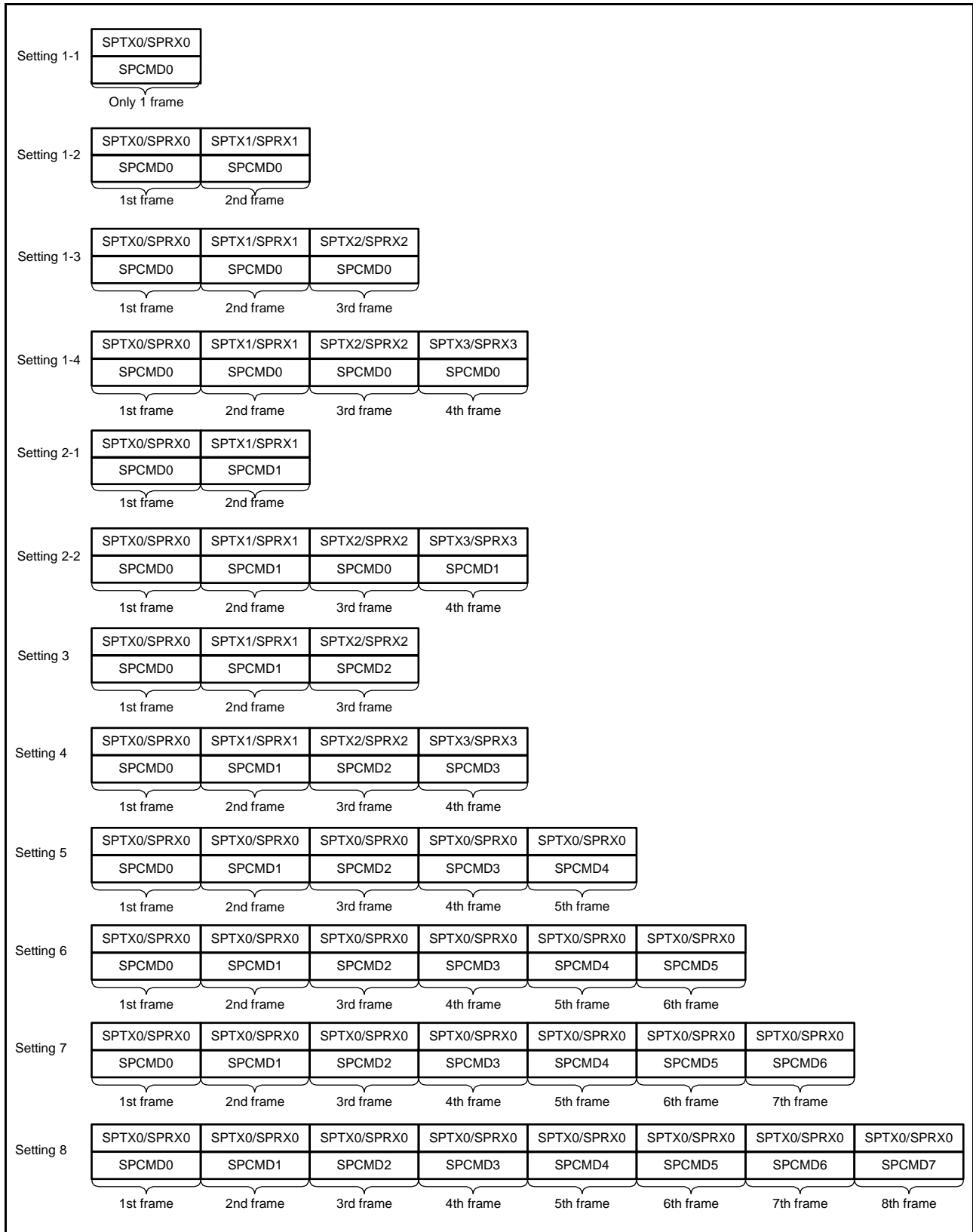


Figure 31.30 Concept of a Frame

Figure 31.31 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 31.4.



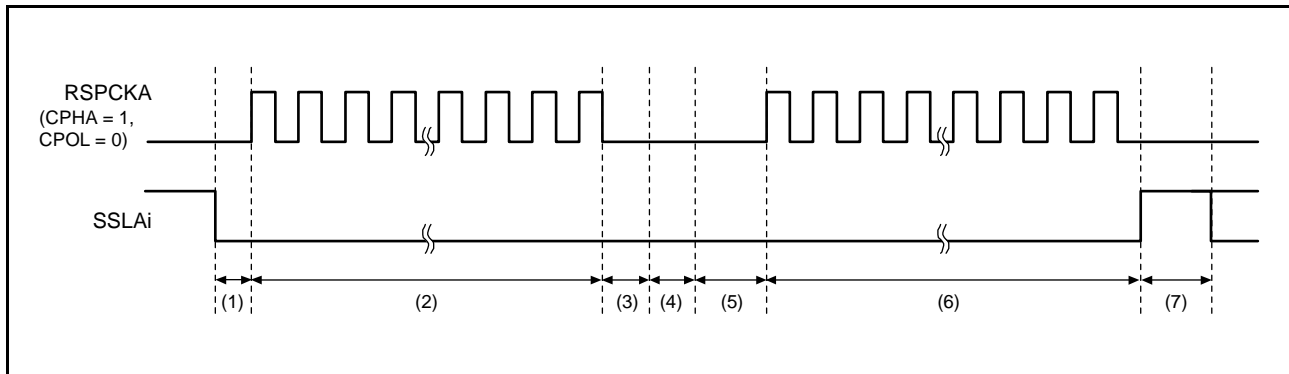
**Figure 31.31 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations**



#### (4) Burst Transfer

If the SPCMDm.SSLKP bit that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLAi signal level during the serial transfer until the beginning of the SSLAi signal assertion for the next serial transfer. If the SSLAi signal level for the next serial transfer is the same as the SSLAi signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLAi signal assertion status (burst transfer).

Figure 31.32 shows an example of an SSLAi 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 31.32. It should be noted that the polarity of the SSLAi output signal depends on the SSLP register settings.



**Figure 31.32 Example of Burst Transfer Operation Using SSLKP Bit**

- (1) Based on SPCMD0, the RSPI asserts the SSLAi 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 SSLAi 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 SSLAi 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 SSLAi signal. In addition, a next-access delay is inserted according to SPCMD1.

If the SSLAi signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLAi signal output settings in the SPCMDm register to be used in the next transfer, the RSPI switches the SSLAi signal status to SSLAi signal assertion ((5) in Figure 31.32) corresponding to the command for the next transfer. Note that if such an SSLAi signal switching occurs, the slaves that drive the MISOA signal compete, and collision of signal levels may occur.

The RSPI in master mode references the SSLAi 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 SSLAi signal assertion for the next transfer that is detected internally.

**(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 31.9. For a definition of RSPCK delay, refer to section 31.3.5, Transfer Format.

**Table 31.9 Relationship among SCKDEN Bit, SPCKD, and RSPCK Delay Value**

SPCMDm.SCKDEN Bit	SPCKD.SCKDL[2:0] Bits	RSPCK Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

**(6) SSL Negation Delay (t2)**

The SSL negation delay value of the RSPI in master mode depends on the SPCMDm.SLNDEN 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.SLNDEN bit and SSLND, as listed in Table 31.10. For a definition of SSL negation delay, refer to section 31.3.5, Transfer Format.

**Table 31.10 Relationship among SLNDEN Bit, SSLND, and SSL Negation Delay Value**

SPCMDm.SLNDEN Bit	SSLND.SLNDL[2:0] Bits	SSL Negation Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	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 31.11. For a definition of next-access delay, refer to section 31.3.5, Transfer Format.

**Table 31.11 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value**

SPCMDm.SPNDEN Bit	SPND.SPNDL[2:0] Bits	Next-Access Delay Value
0	000b to 111b	1 RSPCK + 2 PCLK
1	000b	1 RSPCK + 2 PCLK
	001b	2 RSPCK + 2 PCLK
	010b	3 RSPCK + 2 PCLK
	011b	4 RSPCK + 2 PCLK
	100b	5 RSPCK + 2 PCLK
	101b	6 RSPCK + 2 PCLK
	110b	7 RSPCK + 2 PCLK
	111b	8 RSPCK + 2 PCLK

## (8) Initialization Flowchart

Figure 31.33 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 and I/O ports, refer to the descriptions given in the individual blocks.

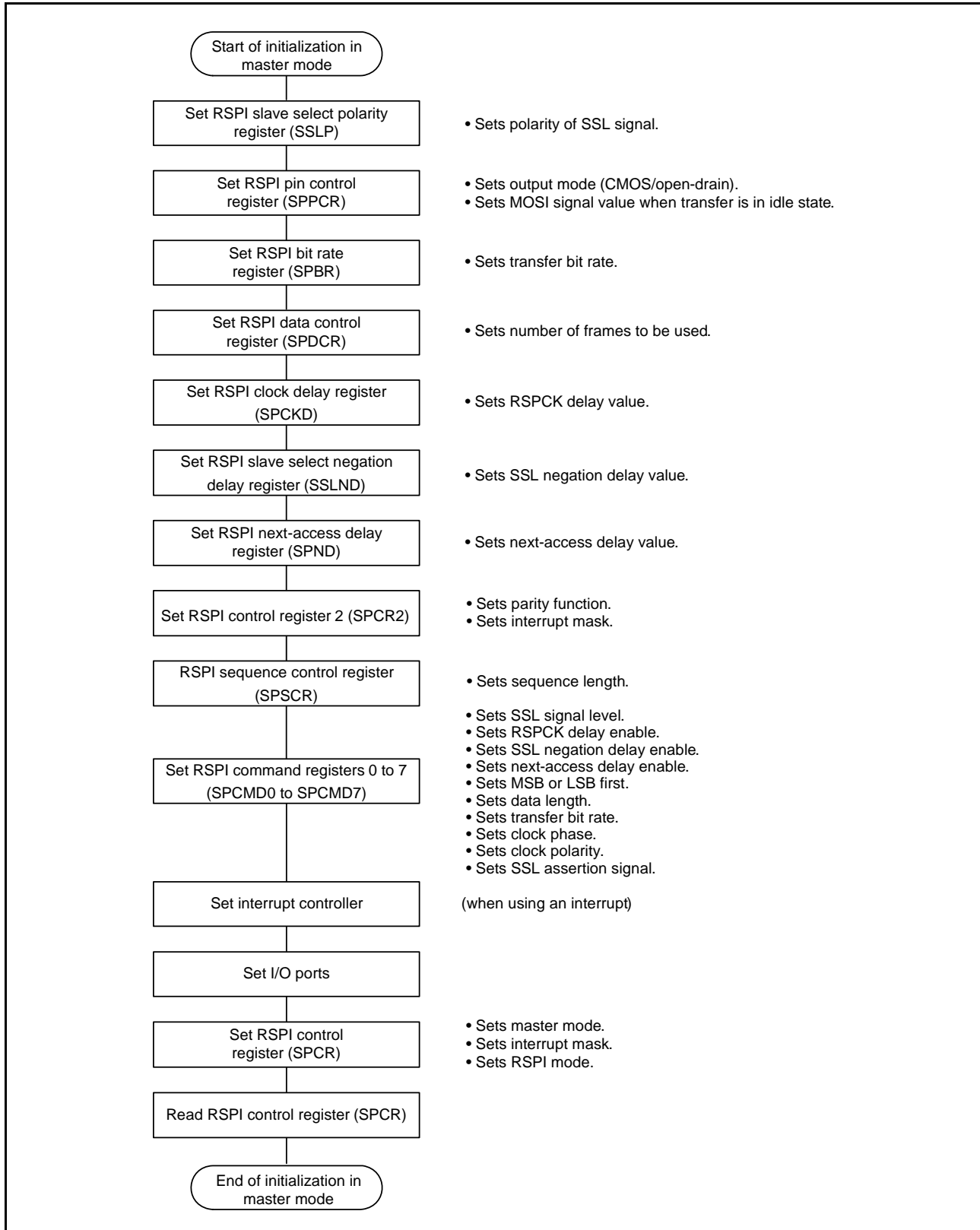


Figure 31.33 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Software Processing Flow

Figure 31.34 to Figure 31.36 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.

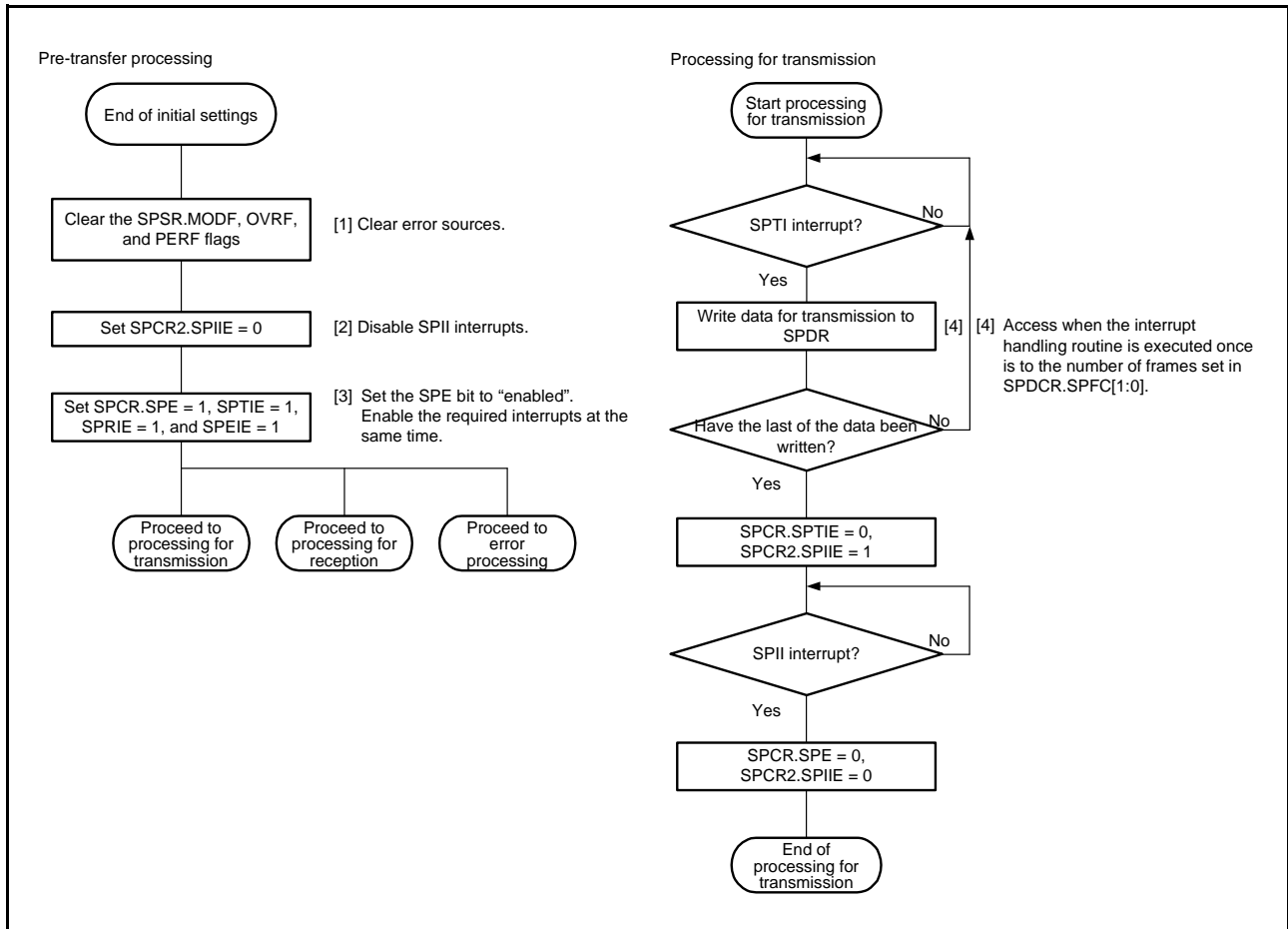


Figure 31.34 Flowchart in Master Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required.

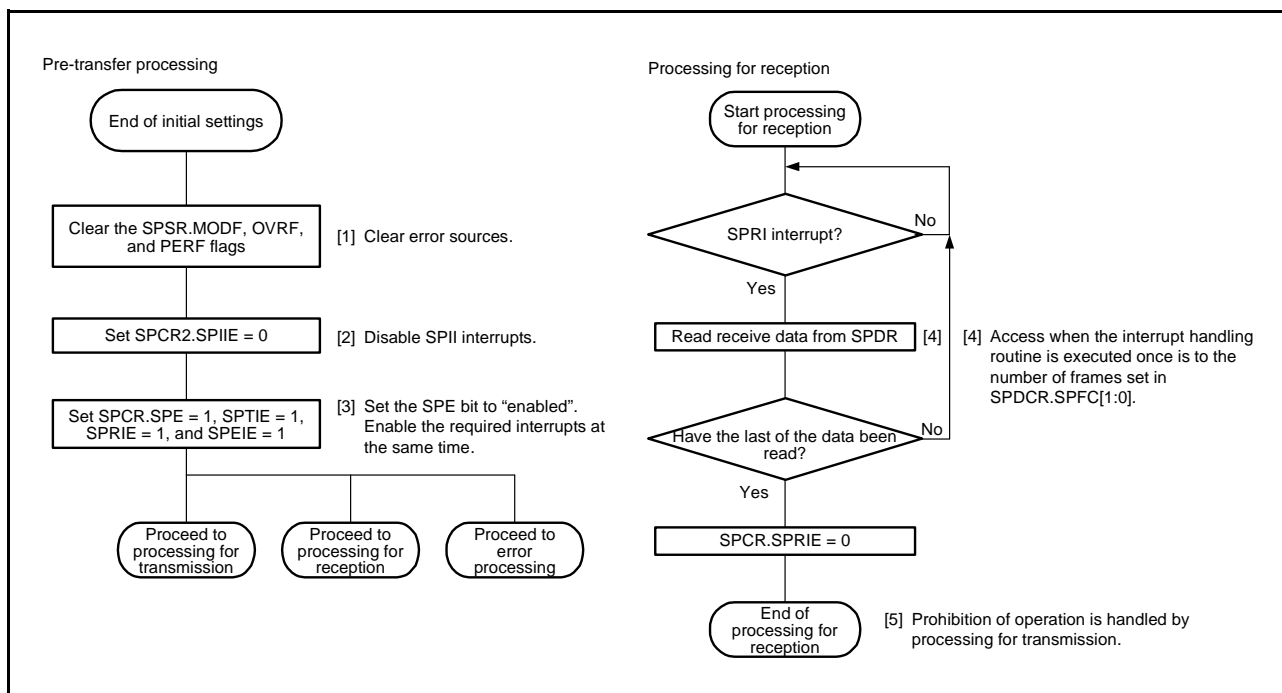


Figure 31.35 Flowchart in Master Mode (Reception)

(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 in 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. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

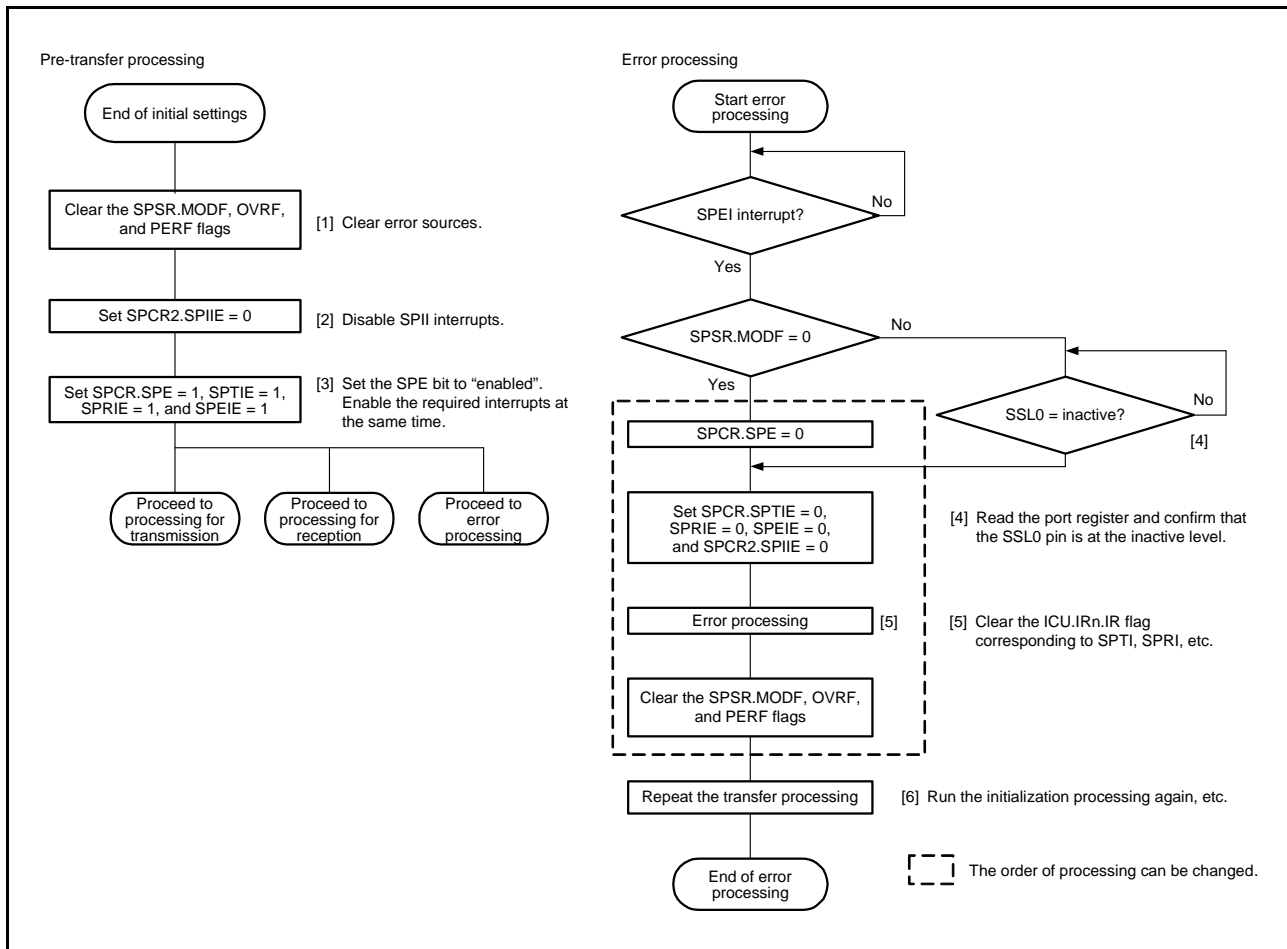


Figure 31.36 Flowchart for Master Mode (Error Processing)

### 31.3.10.2 Slave Mode Operation

#### (1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLA0 input signal assertion, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLA0 input signal triggers the start of a serial transfer.

If the CPHA bit is 1, when detecting the first RSPCKA edge in an SSLA0 signal asserted condition, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 1, the first RSPCKA edge in an SSLA0 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 the CPHA bit setting, the timing at which the RSPI starts driving of the MISOA output signal is the SSLA0 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, refer to section 31.3.5, Transfer Format. The polarity of the SSLA0 input signal depends on the setting of the SSLP.SSL0P bit.

#### (2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCKA 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 SPDR register. 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 SSLA0 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 31.3.8, 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 SSLA0 input signal depends on the SSLP.SSL0P bit setting.

For details on the RSPI transfer format, refer to section 31.3.5, 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 SSLA0 input signal. In the type of configuration shown in Figure 31.7 as an example, if the RSPI is used in single-slave mode, the SSLA0 signal is 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 SSLA0 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 SSLA0 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 SSLA0 input signal. If the CPHA bit is 1, the period from the first RSPCKA edge to the sampling timing for the reception of the final bit in an SSLA0 signal active state corresponds to a serial transfer period. Even when the SSLA0 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 31.37 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 and I/O ports, refer to the descriptions given in the individual blocks.

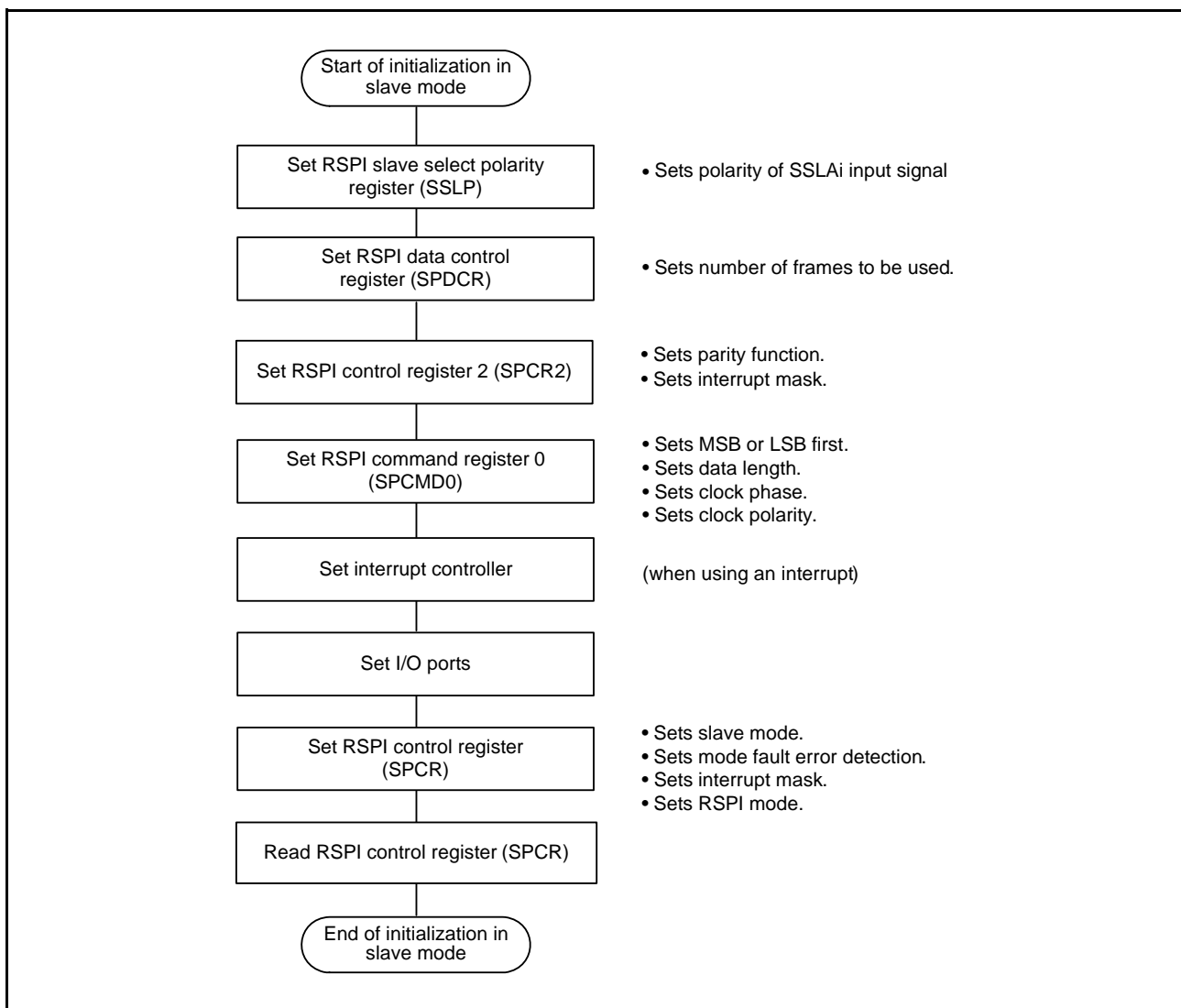


Figure 31.37 Example of Initialization Flowchart in Slave Mode (SPI Operation)

(6) Software Processing Flow

Figure 31.38 to Figure 31.40 show examples of the flow of software processing.

(a) Transmit Processing Flow

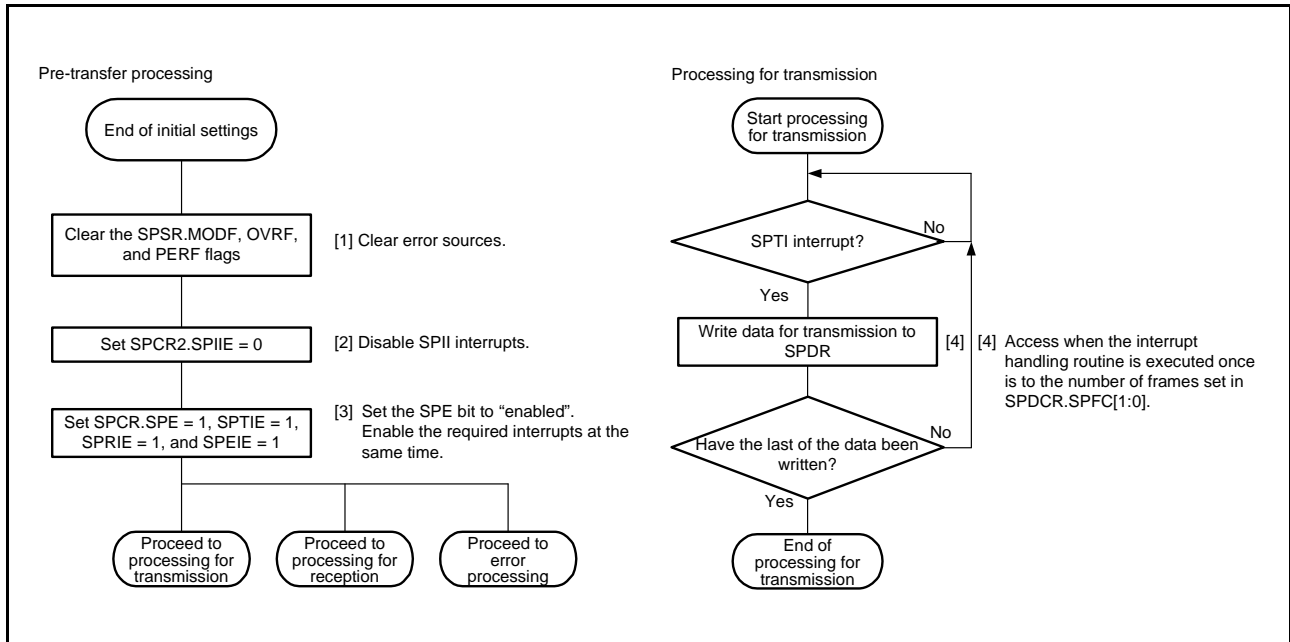


Figure 31.38 Flowchart in Slave Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required.

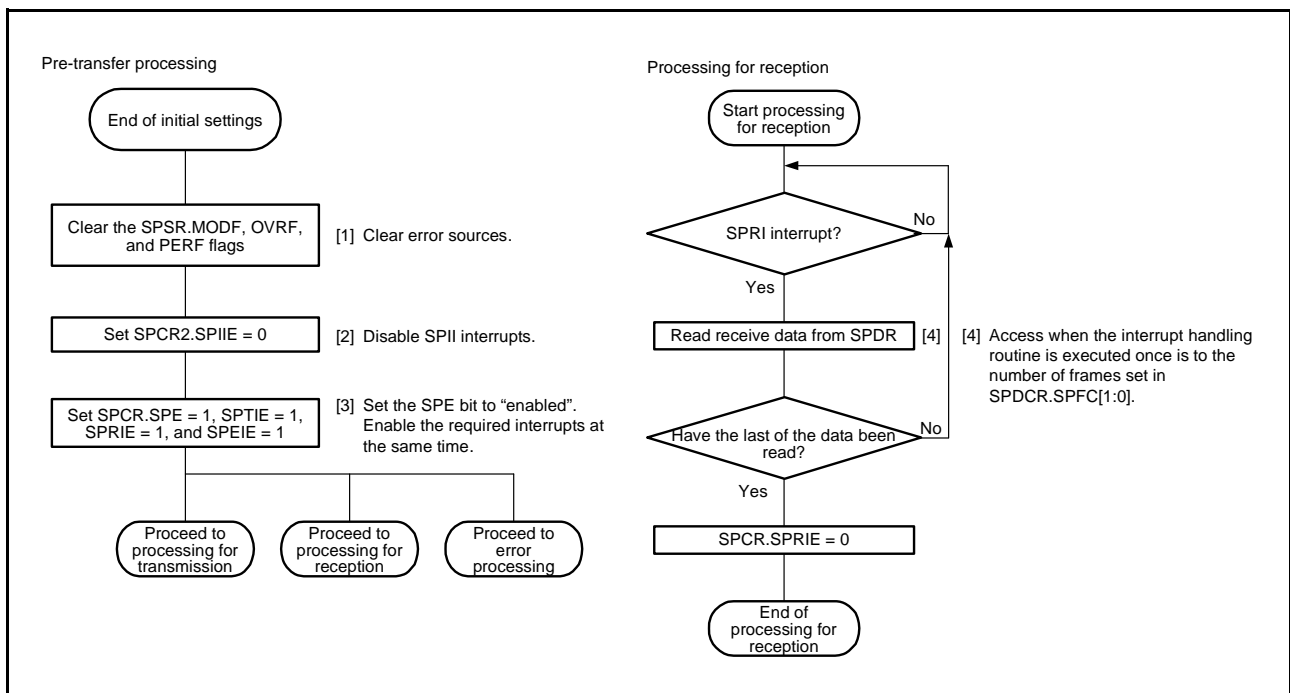


Figure 31.39 Flowchart in Slave Mode (Reception)

(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.

When an error occurs, clear the ICU.IRn.IR flag in 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. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

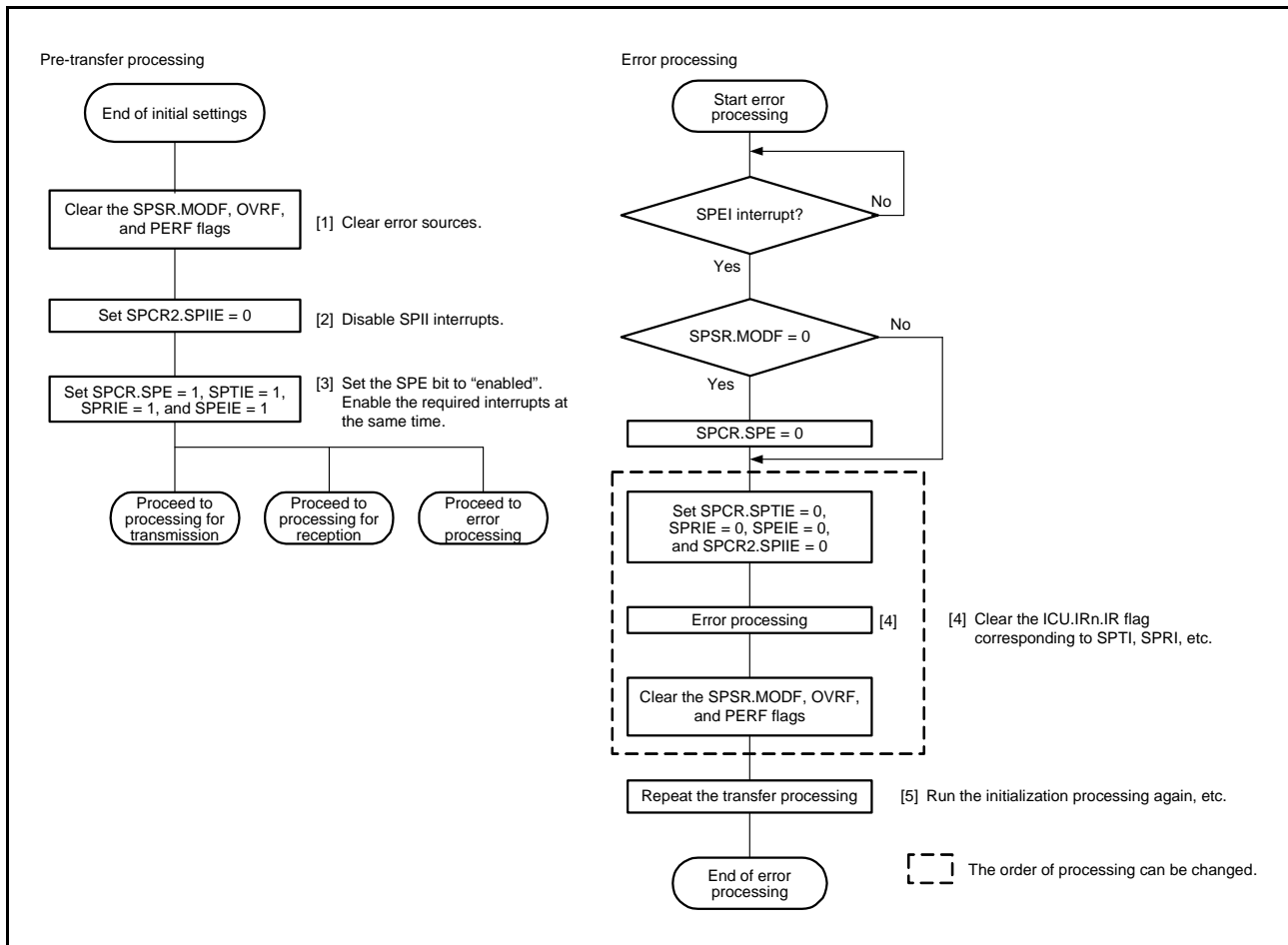


Figure 31.40 Flowchart for Slave Mode (Error Processing)

### 31.3.11 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLAi pin is not used, and the three pins of RSPCKA, MOSIA, and MISOA handle communications. The SSLAi pin is available as I/O port pins.

Although clock synchronous operation does not require use of the SSLAi 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 SSLAi pin is not used.

Furthermore, operation should not be performed if clock synchronous operation proceeds when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

#### 31.3.11.1 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 SPDR register 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, refer to section 31.3.5, Transfer Format.

However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

##### (2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKA 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.SPB[3:0] bit setting.

For details on the RSPI transfer format, refer to section 31.3.5, Transfer Format.

However, transfer in clock synchronous operation is conducted without the SSLA0 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 SSLAi 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: SSLAi output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCKA 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 SPCR.SPE bit 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.

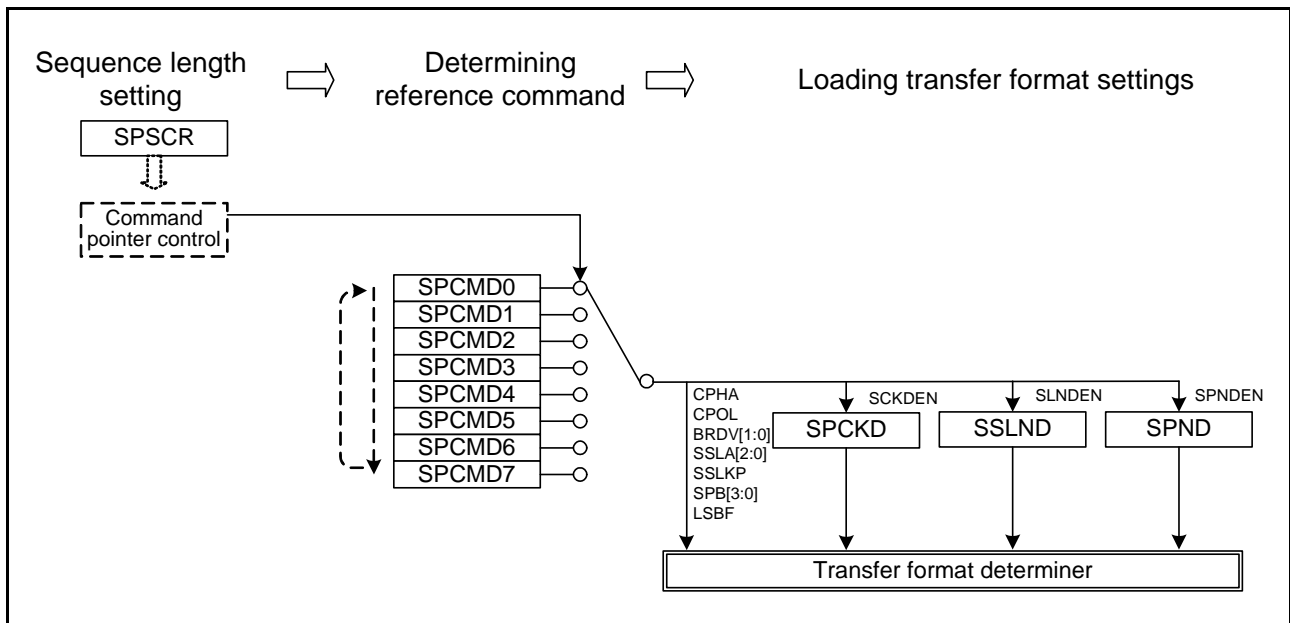


Figure 31.41 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

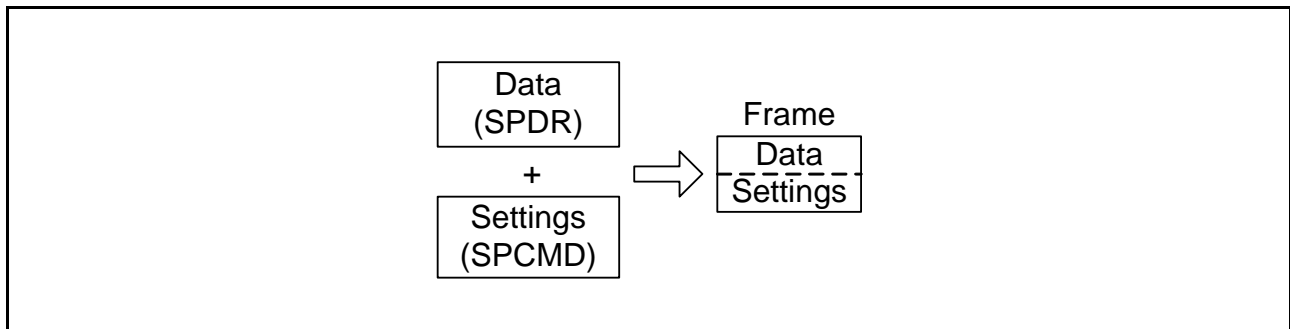
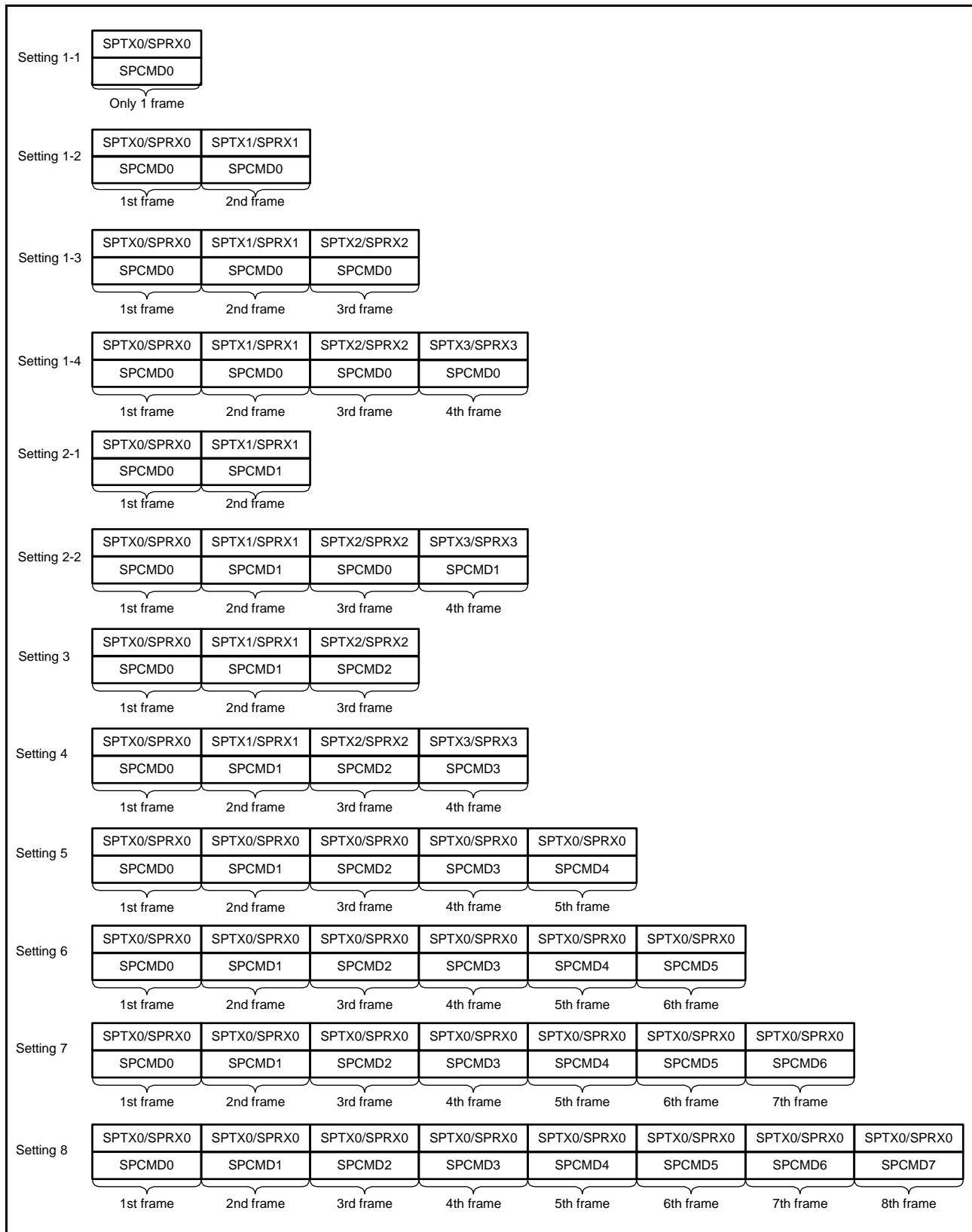


Figure 31.42 Concept of a Frame

Figure 31.43 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 31.4.



**Figure 31.43 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations**

## (4) Initialization Flowchart

Figure 31.44 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 and I/O ports, refer to the descriptions given in the individual blocks.

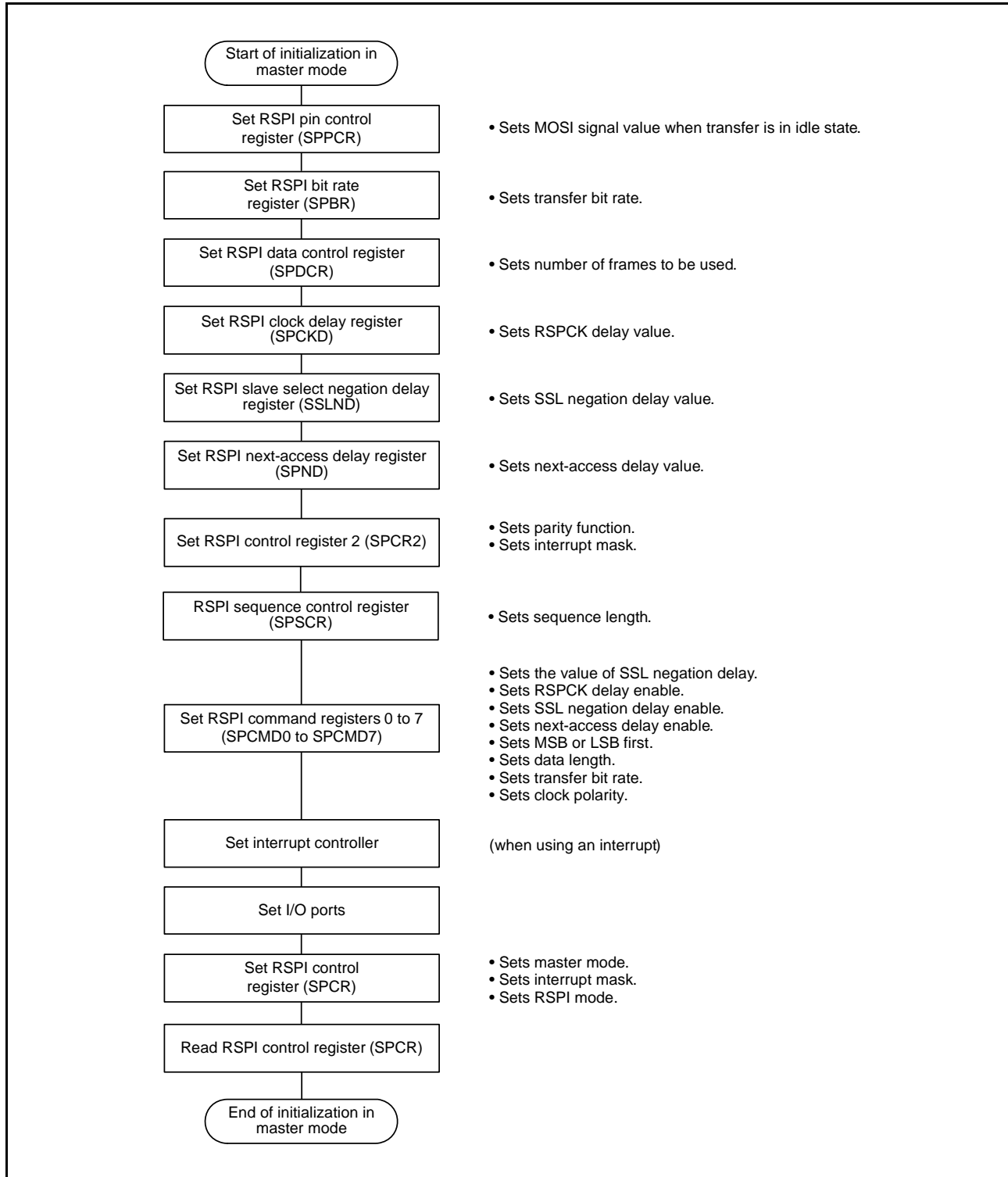


Figure 31.44 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

### (5) Flow of Software Processing

Software processing during clock-synchronous master operation is the same as that for SPI master operation. For details, refer to section 31.3.10.1, (9) Software Processing Flow. Note that mode-fault errors will not occur.

## 31.3.11.2 Slave Mode Operation

### (1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKA 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 drives the MISOA output signal.

For details on the RSPI transfer format, refer to section 31.3.5, 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 RSPCKA edge corresponding to the final sampling timing.

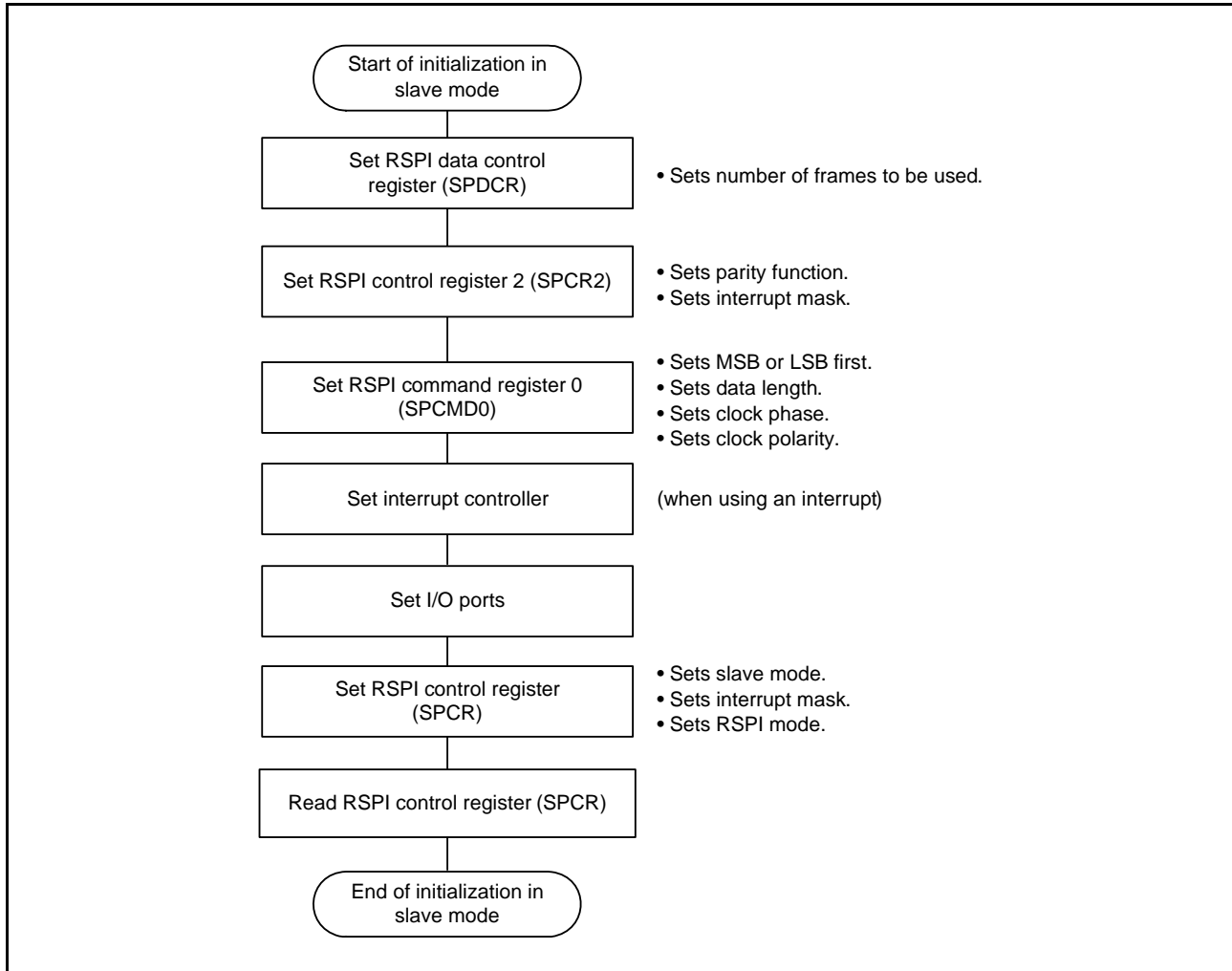
When free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”. The final sampling 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, refer to section 31.3.5, Transfer Format.



### (3) Initialization Flowchart

Figure 31.45 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 and I/O ports, refer to the descriptions given in the individual blocks.



**Figure 31.45 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, refer to section 31.3.10.2, (6) Software Processing Flow. Note that mode-fault errors will not occur.

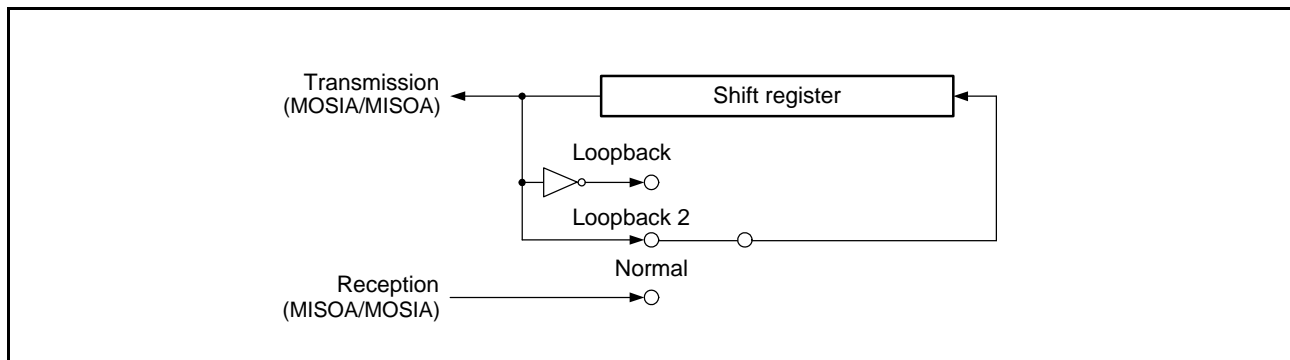
### 31.3.12 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA 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 MOSIA pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOA 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 31.12 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 31.46 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 31.12 SPLP2 and SPLP Bit Settings and Received Data**

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSIA pin or MISOA pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data



**Figure 31.46 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)**

### 31.3.13 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 31.47.

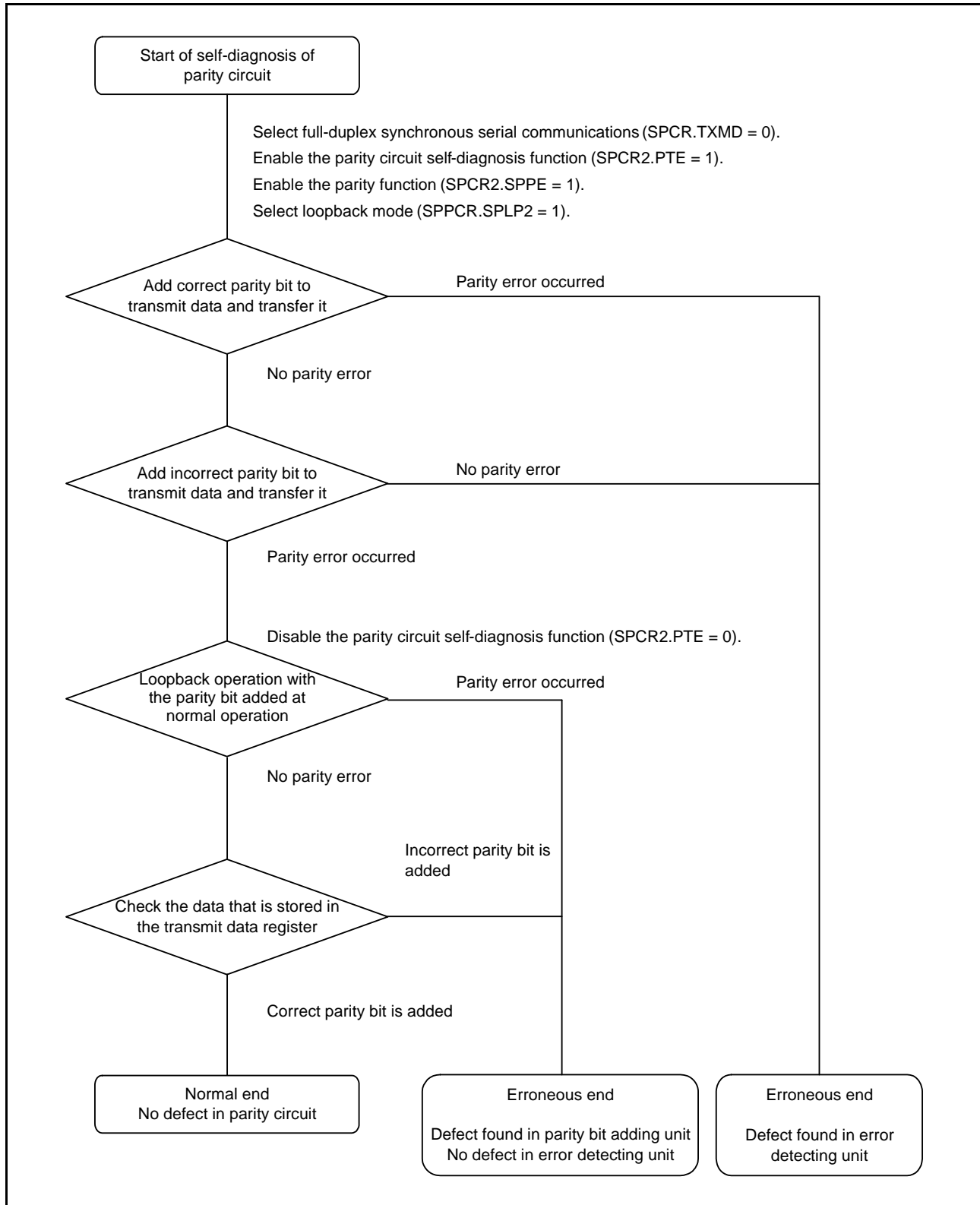


Figure 31.47 Flowchart for Self-Diagnosis of Parity Circuit

### 31.3.14 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 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 31.13. An interrupt is generated on satisfaction of an interrupt condition in Table 31.13. Clear the receive buffer full and transmit buffer empty sources through data transfer.

When using the DTC to perform data transmission/reception, the DTC 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, refer to section 16, 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 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 31.13 Interrupt Sources of RSPI**

Interrupt Source	Symbol	Interrupt Condition	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 SPCR2.SPIIE bit is 1.	Impossible

## 31.4 Usage Notes

### 31.4.1 Setting Module Stop Function

Module stop control register B (MSTPCRB) can be used to enable or disable operation of the RSPI. The RSPI is stopped after a reset. Register access is enabled by releasing the module stop state. For details, refer to **section 11, Low Power Consumption**.

### 31.4.2 Note on Low Power Consumption Functions

When using the module stop function and entering a low power consumption mode other than sleep mode, set the SPCR.SPE bit to 0 before completing communication.

### 31.4.3 Notes 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. Set 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. Set the ICU.IRn.IR flag to 0.

## 32. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

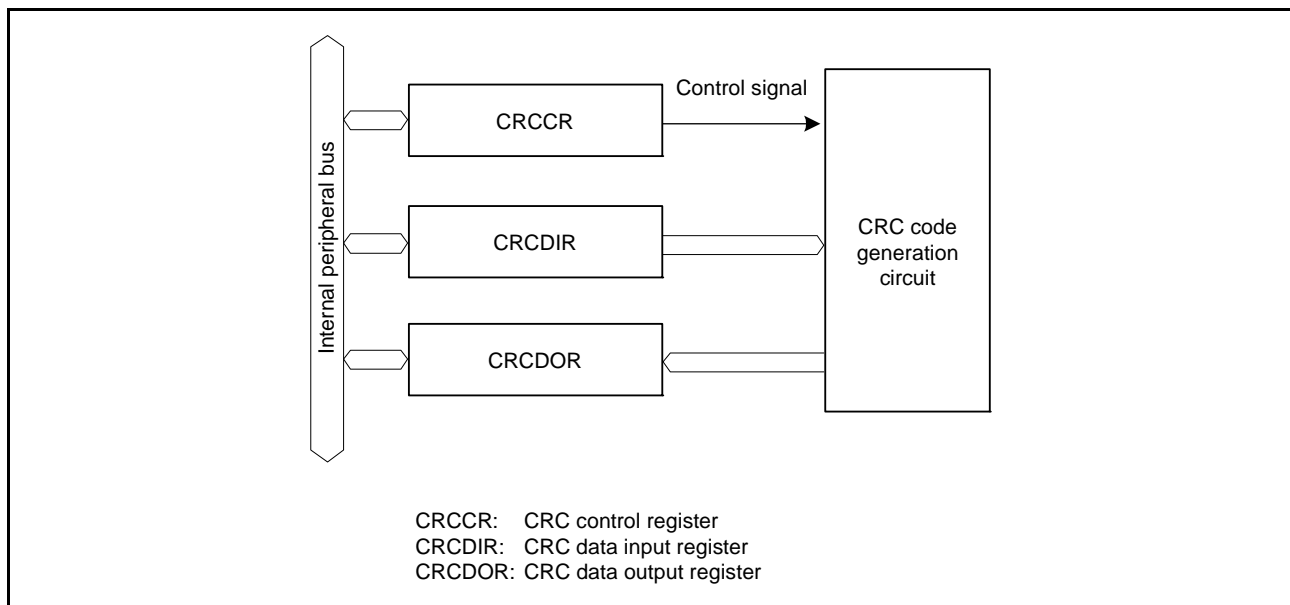
### 32.1 Overview

Table 32.1 lists the specifications of the CRC calculator, and Figure 32.1 shows a block diagram of the CRC calculator.

**Table 32.1 CRC Specifications**

Item	Description
Data for CRC calculation*1	CRC code generated for any desired data in 8n-bit units (where n is a whole number)
CRC processor unit	Operation executed on 8 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	The bit order of CRC calculation results can be switched for LSB first or MSB first communication
Low power consumption function	Module stop state can be set.

Note 1. The circuit does not have functionality to divide data for calculation into CRC calculation units. Write data in 8-bit units.

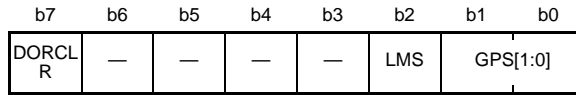


**Figure 32.1 CRC Block Diagram**

## 32.2 Register Descriptions

### 32.2.1 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: 8-bit CRC ( $X^8 + X^2 + X + 1$ ) 1 0: 16-bit CRC ( $X^{16} + X^{15} + X^2 + 1$ ) 1 1: 16-bit CRC ( $X^{16} + X^{12} + X^5 + 1$ )	R/W
b2	LMS	CRC Calculation Switching	0: Generates CRC for LSB first communication. 1: Generates CRC for MSB first communication.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DORCLR	CRCDOR Register Clear	1: Clears the CRCDOR register. This bit is read as 0.	W*1

Note 1. Only 1 can be written.

#### DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is set to 0000h.

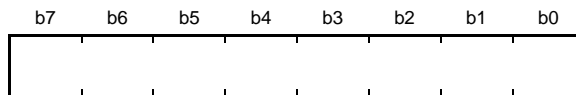
This bit is read as 0. Only 1 can be written.

#### LMS Bit (CRC Calculation Switching)

Set this bit to select the bit order of generated 16-bit CRC code. Transmit the lower-order byte (b7 to b0) of the CRC code first for LSB first communication and the higher-order byte (b15 to b8) first for MSB first communication. For details on transmitting and receiving CRC code, refer to section 32.3, Operation.

### 32.2.2 CRC Data Input Register (CRCDIR)

Address(es): 0008 8281h

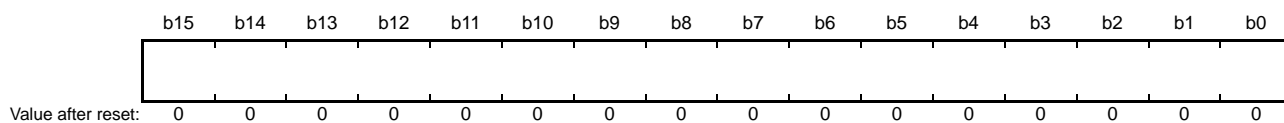


Value after reset: 0 0 0 0 0 0 0 0

CRCDIR is a readable/writable register. Write data for CRC calculation to this register.

### 32.2.3 CRC Data Output Register (CRCDOR)

Address(es): 0008 8282h



CRCDOR is a readable/writable register.

Since its initial value is 0000h, rewrite the CRCDOR register to perform calculation using a value other than the initial value.

Data written to the CRCDIR register is CRC calculated and the result is stored in the CRCDOR register. If the CRC code is calculated following the transferred data and the result is 0000h, there is no CRC error.

When an 8-bit CRC ( $X^8 + X^2 + X + 1$  polynomial) is in use, the valid CRC code is obtained in the low-order byte (b7 to b0). The high-order byte (b15 to b8) is not updated.



### 32.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first transfer.

The following shows examples of generating the CRC code for input data (F0h) using the 16-bit CRC generating polynomial ( $X^{16} + X^{12} + X^5 + 1$ ). In these examples, the value of the CRC data output register (CRCDOR) is cleared before CRC calculation.

When an 8-bit CRC (with the polynomial  $X^8 + X^2 + X + 1$ ) is in use, the valid bits of the CRC code are obtained in the lower-order byte of CRCDOR.

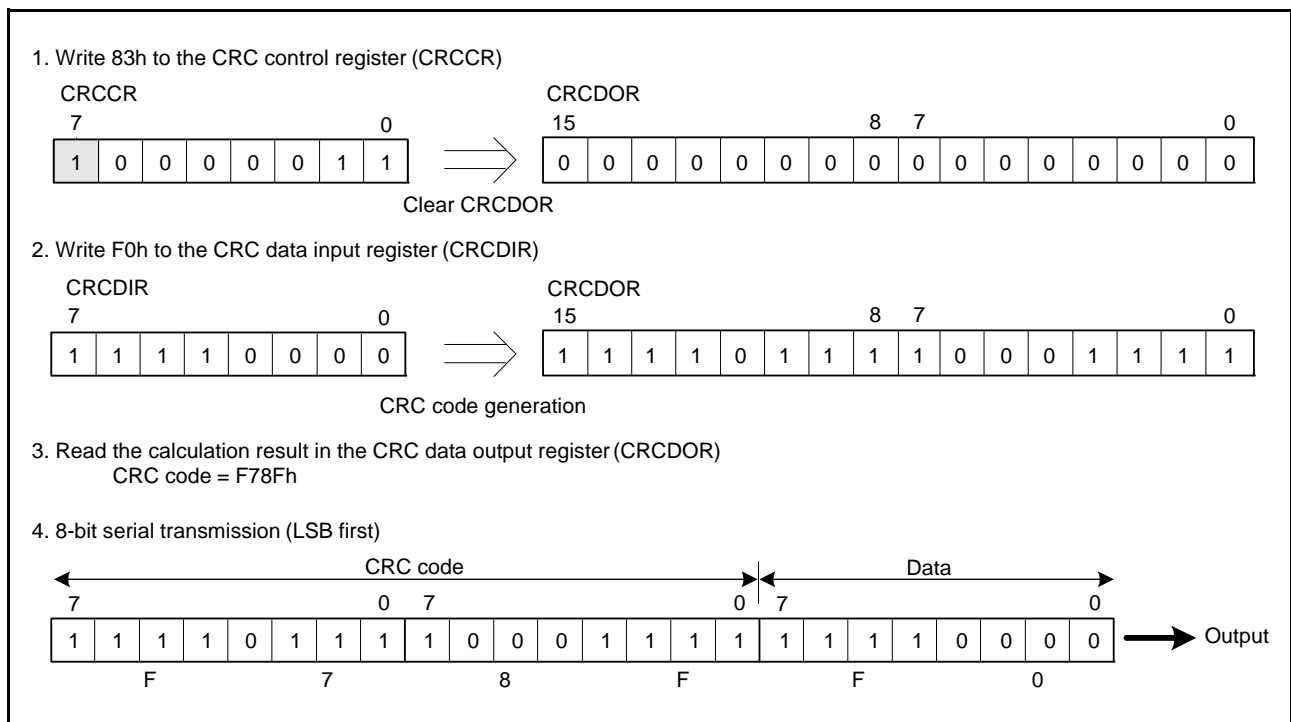


Figure 32.2 LSB First Data Transmission

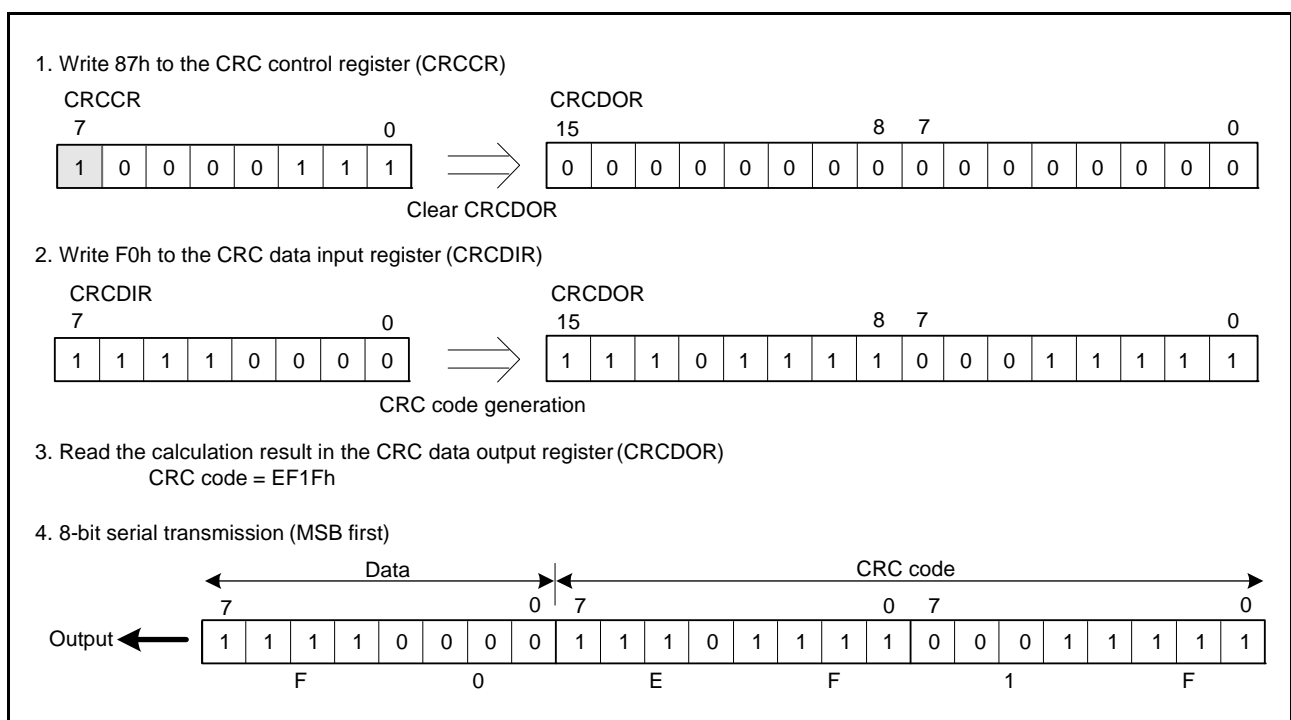


Figure 32.3 MSB First Data Transmission

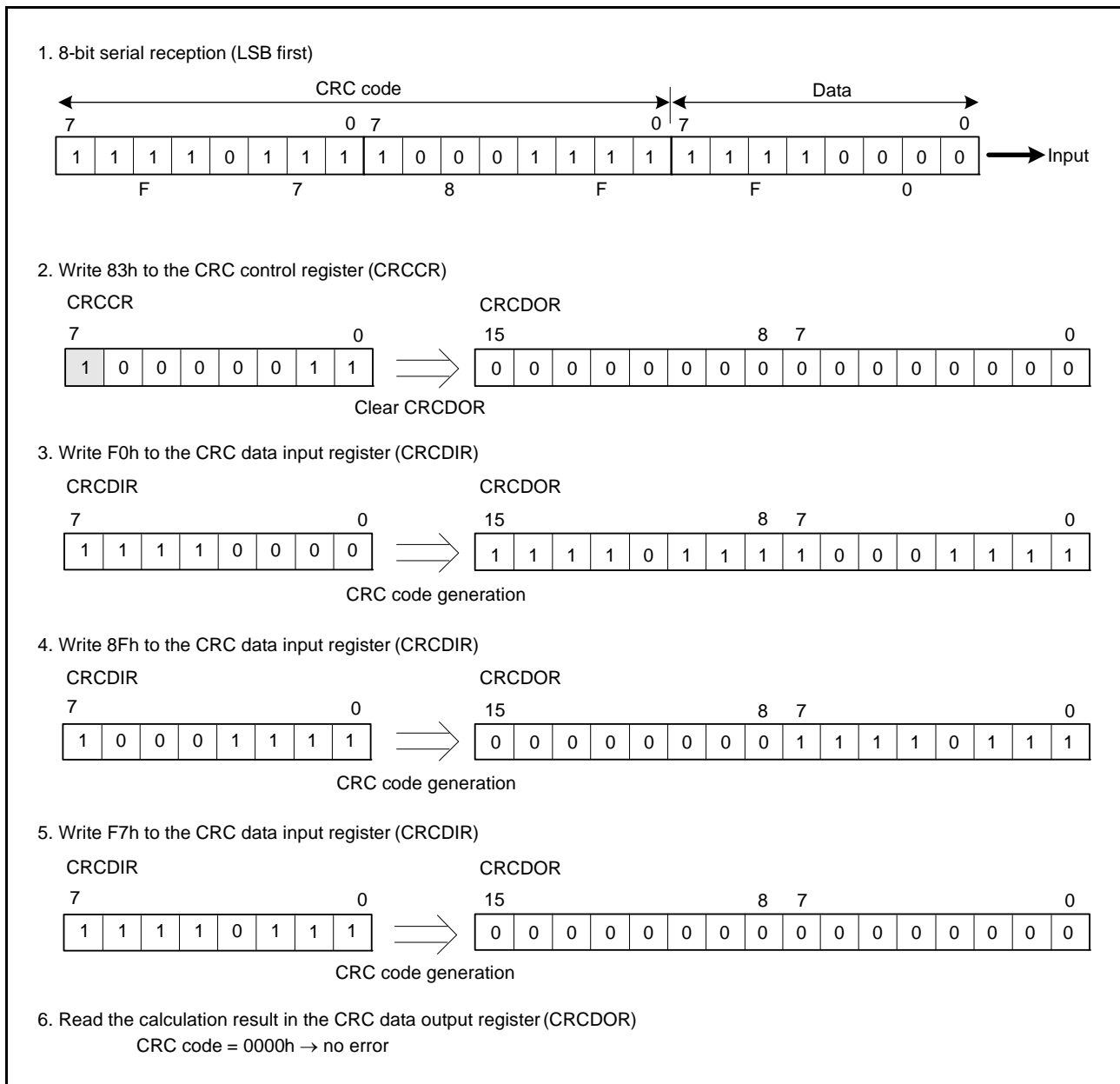


Figure 32.4 LSB First Data Reception

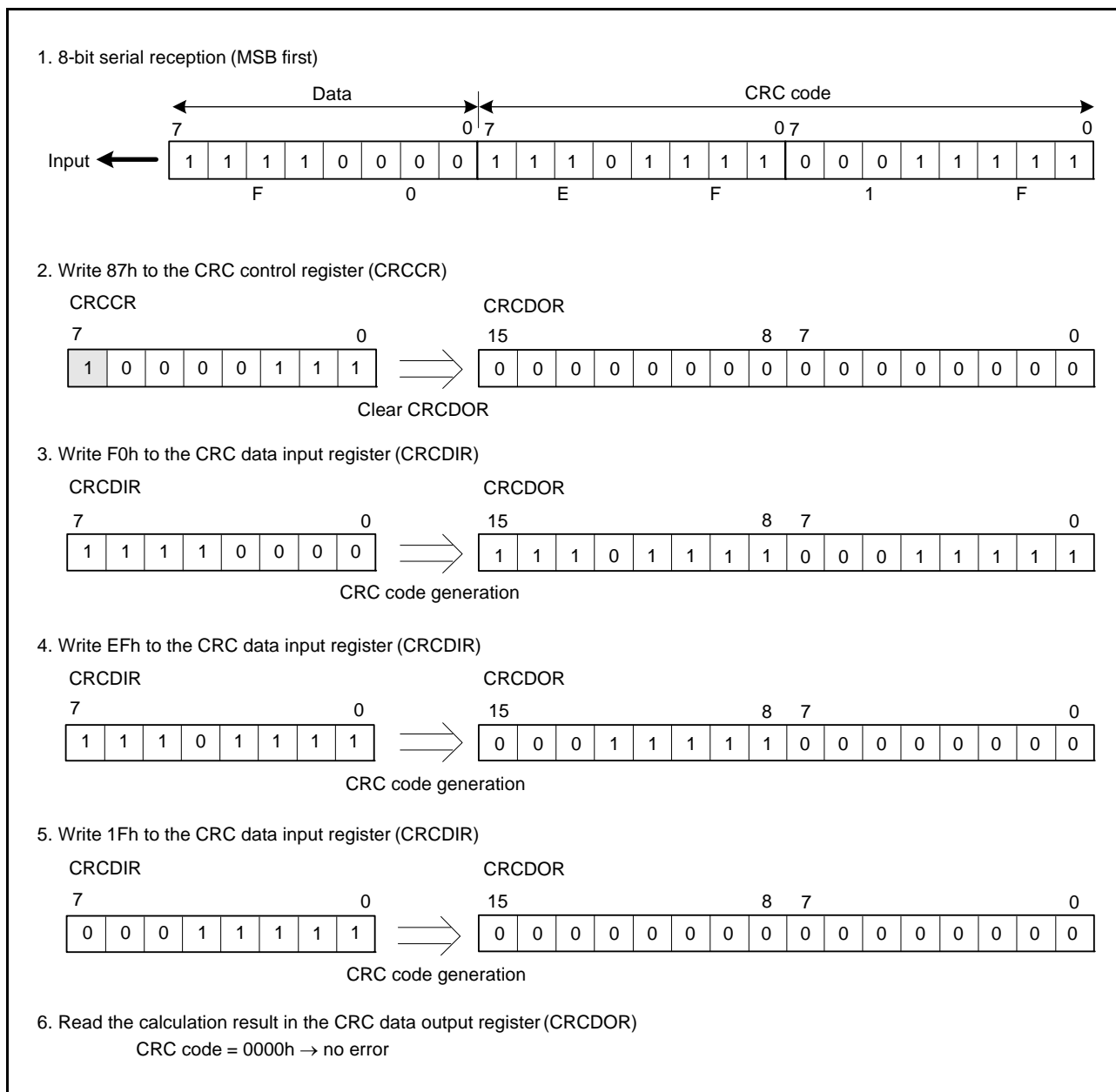


Figure 32.5 MSB First Data Reception

### 32.4 Usage Notes

#### 32.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using module stop control register B (MSTPCRB). After a reset, the CRC is in the module stop state. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

#### 32.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

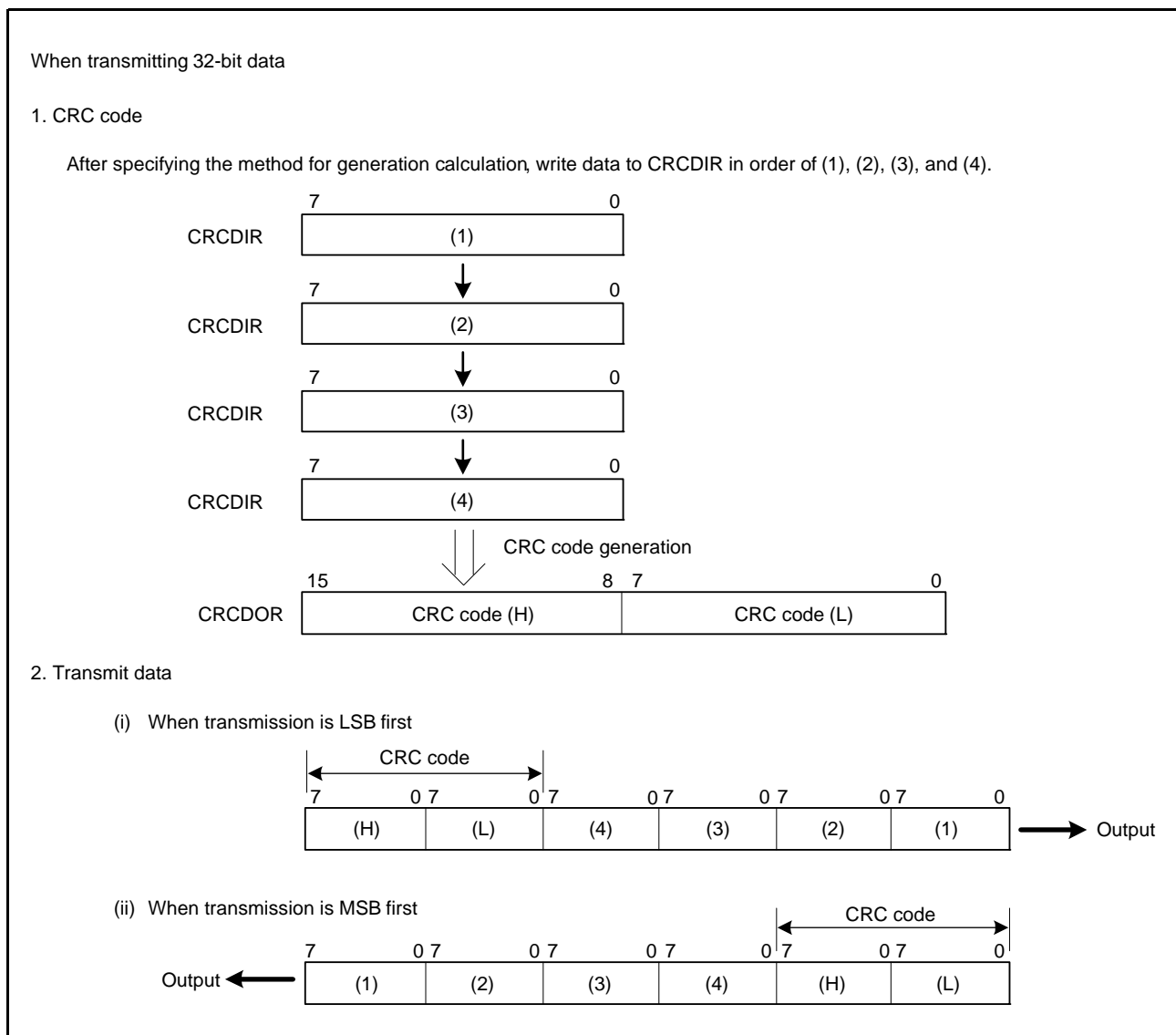


Figure 32.6 LSB First and MSB First Data Transmission

## 33. LCD Controller/Driver (LCDC)

This MCU incorporates a controller for LCD display and display pins.

### 33.1 Overview

**Table 33.1 LCD Controller/Driver Specifications**

Item	Description
Features	<ul style="list-style-type: none"> <li>Liquid crystal waveform (waveform A or B) can be selected.</li> <li>The LCD driver voltage generator can switch internal voltage boosting method, capacitor split method, and external resistance division method.</li> <li>Automatic output of segment and common signals based on automatic display data register read</li> <li>The reference voltage to be generated when operating the voltage boost circuit can be selected from 16 steps (contrast adjustment).</li> <li>LCD blinking and display selectable</li> </ul>
Number of pins	For details on the number of pins, see Table 33.2, Number of LCD Display Function Pins of Each Product.
Source clocks	Main clock oscillator, sub-clock oscillator, low-speed on-chip oscillator, high-speed on-chip oscillator, and IWDTC-dedicated on-chip oscillator
Low power consumption function	Module stop state can be set.

The number of LCD display function pins differs depending on the package. Table 33.2 shows the number of pins of each product.

**Table 33.2 Number of LCD Display Function Pins of Each Product**

Item	This MCU															
	64 pins								100 pins							
LCD controller/driver	Number of segment pins (SEG): 20 (16)*1 Number of common pins (COM): 8								Number of segment pins (SEG): 40 (36)*1 Number of common pins (COM): 8							
Multiplexed I/O port	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Port 1	—	—	—	—	—	—	—	—	—	—	—	—	SEG 00	SEG 01	SEG 02	SEG 03
Port 3	—	—	—	—	—	—	CAP H*2	CAP L*2	—	—	—	—	—	—	CAP H*2	CAP L*2
Port 5	—	—	VL1 *3	VL2 *3	—	—	—	—	—	SEG 04	VL1 *3	VL2 *3	SEG 05	SEG 06	SEG 07	SEG 08
Port A	—	—	—	SEG 20	SEG 21	—	SEG 23	SEG 24	SEG 18	—	SEG 19	SEG 20	SEG 21	SEG 22	SEG 23	SEG 24
Port B	SEG 11/ COM4	SEG 12/ COM5	SEG 13/ COM6	—	SEG 15/ COM7	—	SEG 17	—	SEG 11/ COM4	SEG 12/ COM5	SEG 13/ COM6	SEG 14	SEG 15/ COM7	SEG 16	SEG 17	—
Port C	VL3 *3	VL4 *3	COM 0	COM 1	COM 2	COM 3	—	—	VL3 *3	VL4 *3	COM 0	COM 1	COM 2	COM 3	SEG 09	SEG 10
Port D	—	—	—	—	—	SEG 37	SEG 38	SEG 39	—	—	—	SEG 35	SEG 36	SEG 37	SEG 38	SEG 39
Port E	SEG 33	SEG 34	SEG 27	SEG 28	SEG 29	SEG 30	SEG 31	SEG 32	SEG 33	SEG 34	SEG 27	SEG 28	SEG 29	SEG 30	SEG 31	SEG 32
Port F	—	—	—	—	—	—	—	—	SEG 25	SEG 26	—	—	—	—	—	—

Note 1. ( ) indicates the number of signal output pins when 8-time slice is selected.

Note 2. CAPH and CAPL are capacitor connection pins for the LCD controller/driver.

Note 3. VL1, VL2, VL3, and VL4 are power supply pins for driving the LCD.

Table 33.3 and Table 33.4 show the maximum number of pixels that can be displayed in each display mode, and Figure 33.1 shows a block diagram.

**Table 33.3 Maximum Number of Pixels of 64-Pin Products**

Drive Waveform for LCD Driver	LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Maximum Number of Pixels
Liquid crystal waveform (waveform A)	External resistance division	—	Static	20 (20 segment signals, 1 common signal)
		1/2	2	40 (20 segment signals, 2 common signals)
			3	60 (20 segment signals, 3 common signals)
		1/3	3	80 (20 segment signals, 4 common signals)
			4	128 (16 segment signals, 8 common signals)
	Internal voltage boosting	1/3	3	60 (20 segment signals, 3 common signals)
			4	80 (20 segment signals, 4 common signals)
		1/4	8	128 (16 segment signals, 8 common signals)
	Capacitor split	1/3	3	60 (20 segment signals, 3 common signals)
			4	80 (20 segment signals, 4 common signals)
Liquid crystal waveform (waveform B)	External resistance division, internal voltage boosting	1/3	4	128 (16 segment signals, 8 common signals)
		1/4	8	
	Capacitor split	1/3	4	80 (20 segment signals, 4 common signals)

**Table 33.4 Maximum Number of Pixels of 100-Pin Products**

Drive Waveform for LCD Driver	LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Maximum Number of Pixels
Liquid crystal waveform (waveform A)	External resistance division	—	Static	40 (40 segment signals, 1 common signal)
		1/2	2	80 (40 segment signals, 2 common signals)
			3	120 (40 segment signals, 3 common signals)
		1/3	3	160 (40 segment signals, 4 common signals)
			4	288 (36 segment signals, 8 common signals)
	Internal voltage boosting	1/3	3	120 (40 segment signals, 3 common signals)
			4	160 (40 segment signals, 4 common signals)
		1/4	8	288 (36 segment signals, 8 common signals)
	Capacitor split	1/3	3	120 (40 segment signals, 3 common signals)
			4	160 (40 segment signals, 4 common signals)
Liquid crystal waveform (waveform B)	External resistance division, internal voltage boosting	1/3	4	288 (36 segment signals, 8 common signals)
		1/4	8	
	Capacitor split	1/3	4	160 (40 segment signals, 4 common signals)

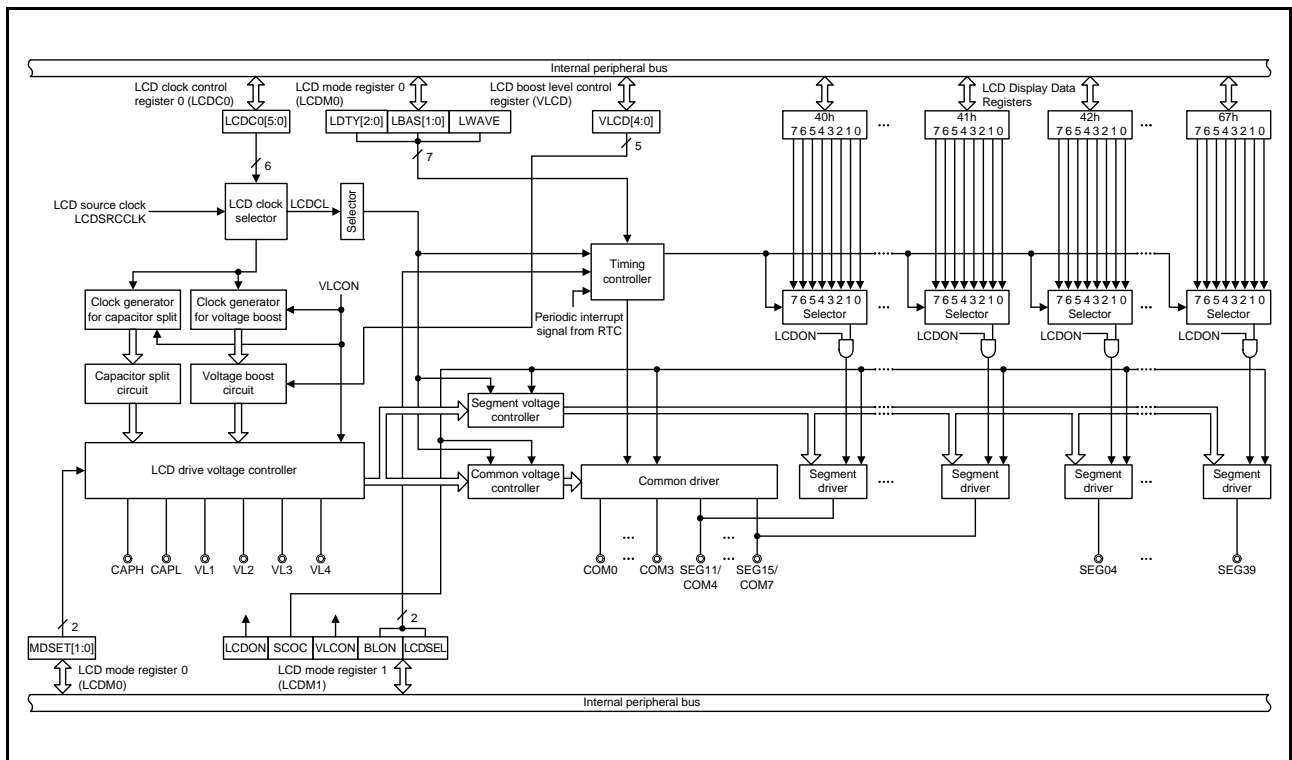
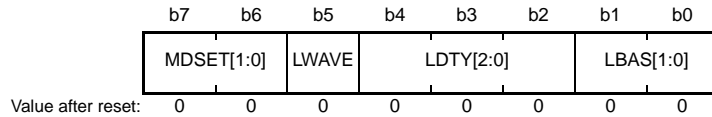


Figure 33.1 Block Diagram of LCD Controller/Driver

### 33.2 Register Descriptions

#### 33.2.1 LCD Mode Register 0 (LCDM0)

Address(es): 000A 0800h



Bit	Symbol	Bit Name	Description	R/W
b0	LBAS[1:0]	LCD Display Bias Method Select	b1 b0 0 0: 1/2 bias method	R/W
b1			0 1: 1/3 bias method 1 0: 1/4 bias method 1 1: Setting prohibited	R/W
b2	LDTY[2:0]	Time Slice of LCD Display Select	b4 b2 0 0 0: Static	R/W
b3			0 0 1: 2-time slice	R/W
b4			0 1 0: 3-time slice 0 1 1: 4-time slice 1 0 1: 8-time slice Settings other than above are prohibited.	R/W
b5	LWAVE	LCD Display Waveform Select	0: Waveform A 1: Waveform B	R/W
b6	MDSET[1:0]	LCD Drive Voltage Generator Select	b7 b6 0 0: External resistance division method	R/W
b7			0 1: Internal voltage boosting method*1 1 0: Capacitor split method*1 1 1: Setting prohibited	R/W

Note: Do not rewrite the LCDM0 register value while the LCDM1.SCOC bit = 1.

Note: When "Static" is selected (LDTY[2:0] bits = 000b), be sure to set the LBAS[1:0] bits to the value after reset (00b). Otherwise, the operation will not be guaranteed.

Note: Only the combinations of display waveform, number of time slices, and bias method shown in Table 33.5 are supported. Combinations of settings not shown in Table 33.5 are prohibited.

Note 1. Selecting IWDTCCLK as the LCD source clock (LCDSRCCLK) is prohibited.

**Table 33.5 Combinations of Display Waveform, Time Slices, and Bias Method**

Display Mode			Set Value						Driving Voltage Generation Method		
Display Waveform	Number of Time Slices	Bias Mode	LWAVE	LDTY[2:0]			LBAS[1:0]		External Resistance Division	Internal Voltage Boosting	Capacitor Split
Waveform A	8	1/4	0	1	0	1	1	0	A	A	N/A
Waveform A	4	1/3	0	0	1	1	0	1	A	A	A
Waveform A	3	1/3	0	0	1	0	0	1	A	A	A
Waveform A	3	1/2	0	0	1	0	0	0	A	N/A	N/A
Waveform A	2	1/2	0	0	0	1	0	0	A	N/A	N/A
Waveform A	Static		0	0	0	0	0	0	A	N/A	N/A
Waveform B	8	1/4	1	1	0	1	1	0	A	A	N/A
Waveform B	4	1/3	1	0	1	1	0	1	A	A	A

A: Available, N/A: Not available



## 33.2.2 LCD Mode Register 1 (LCDM1)

Address(es): 000A 0801h

b7	b6	b5	b4	b3	b2	b1	b0
LCDON	SCOC	VLCON	BLON	LCDSEL	—	—	LCDVLM
0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b0	LCDVLM*1	Voltage Boosting Pin Initial Value Switching Control	0: VCC voltage > 3.8 V 1: VCC voltage ≤ 3.8 V Set this bit to 1 to set the internal voltage boosting method.	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	LCDSEL	Display Data Area Control	b4 b3 0 0: Displaying an A-pattern area data (lower 4 bits of LCD display data register)	R/W
b4	BLON*2	Display Data Area Control	0 1: Displaying a B-pattern area data (higher 4 bits of LCD display data register) 1 0: Alternately displaying A-pattern and B-pattern area data (blinking display corresponding to the periodic interrupt (PRD) timing of the real-time clock (RTC)) 1 1: Alternately displaying A-pattern and B-pattern area data (blinking display corresponding to the periodic interrupt (PRD) timing of the real-time clock (RTC))	R/W
b5	VLCON	Voltage boost Circuit or Capacitor Split Circuit Operation Enable/Disable	0: Stops voltage boost circuit or capacitor split circuit operation 1: Enables voltage boost circuit or capacitor split circuit operation	R/W
b6	SCOC	LCD Display Enable/Disable	b7 b6 0 0: Output ground level to segment/common pin	R/W
b7	LCDON	LCD Display Enable/Disable	0 1: Display off (all segment outputs are deselected.) 1 0: Output ground level to segment/common pin 1 1: Display on	R/W

Note: To reduce power consumption when the liquid crystal waveform is not displayed on the LCD, set the SCOC and VLCON bits to 0.

Note: When the external resistance division method has been set (LCDM0.MDSET[1:0] bits = 00b) or capacitor split method has been set (MDSET[1:0] bits = 10b), set the LCDVLM bit to 0.

Note: Do not rewrite the VLCON and LCDVLM bits while SCOC bit = 1.

Note: Set the LCDM1.BLON and LCDSEL bits to 0 when 8 has been selected as the number of time slices for the display mode.

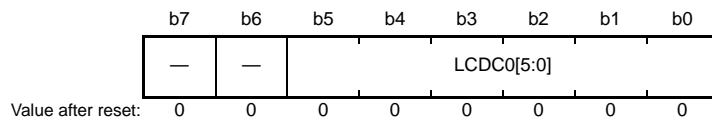
Note: To use the internal voltage boosting method, specify the reference voltage by using the VLCD register (select the internal boosting method (by setting the LCDM0.MDSET[1:0] bits to 01b) if the default reference voltage is used), wait for the reference voltage setup time (5 ms (min.)), and then set the VLCON bit to 1.

Note 1. This function is used to shorten the boost stabilization time by setting the VLx pin to the default status when the voltage boost circuit is used.  
If the VCC voltage is higher than 3.8 V when boosting is started, set the LCDVLM bit to 0; if the VCC voltage is 3.8 V or less, set the LCDVLM bit to 1.

Note 2. When IWDTCCLK is selected as the LCD source clock (LCDSRCCLK), set the BLON bit to 0.

## 33.2.3 LCD Clock Control Register 0 (LCDC0)

Address(es): 000A 0802h



Bit	Symbol	Bit Name	Description	R/W
b0	LCDC0[5:0]	LCD Clock (LCDCL) Setting	When the main clock, HOCO clock, or LOCO clock is selected as the LCD source clock	R/W
b1			b5            b0 0 1 0 0 0 1: x1/256	R/W
b2			0 1 0 0 1 0: x1/512	R/W
b3			0 1 0 0 1 1: x1/1024	R/W
b4			0 1 0 1 0 0: x1/2048	R/W
b5			0 1 0 1 0 1: x1/4096 0 1 0 1 1 0: x1/8192 0 1 0 1 1 1: x1/16384 0 1 1 0 0 0: x1/32768 0 1 1 0 0 1: x1/65536 0 1 1 0 1 0: x1/131072 0 1 1 0 1 1: x1/262144 1 0 1 0 1 1: x1/524288	R/W
Settings other than above are prohibited.				
-----				
When the sub-clock is selected as the LCD source clock				
			b5            b0 0 0 0 1 0 0: x1/32 0 0 0 1 0 1: x1/64 0 0 0 1 1 0: x1/128 0 0 0 1 1 1: x1/256 0 0 1 0 0 0: x1/512 0 0 1 0 0 1: x1/1024	
Settings other than above are prohibited.				
-----				
When the IWDT-dedicated on-chip oscillator is selected as the LCD source clock				
			b5            b0 0 0 0 0 1 1: x1/16*1 0 0 0 1 0 0: x1/32*1 0 0 0 1 0 1: x1/64*1 0 0 0 1 1 0: x1/128*1 0 0 0 1 1 1: x1/256*1	
Settings other than above are prohibited.				
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the frame frequency in a range from 32 Hz to 128 Hz. If the internal voltage boosting method is specified, set the LCD clock (LCDCL) to 512 Hz or lower.

Note: Do not set LCDC0 when the LCDM1.SCOC bit is 1.

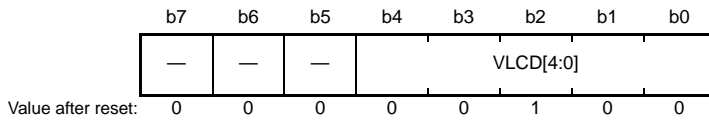
Note 1. When the Internal voltage boosting method has been specified (LCDM0.MDSET[1:0] bits = 01b) or capacitor split method has been specified (LCDM0.MDSET[1:0] bits = 10b), selecting IWDTCLK as the LCD source clock (LCDSRCCLK) is prohibited.

The frame frequency is determined according to the LCD clock and the number of time slices.

For details on selecting the LCD source clock, refer to section 9, Clock Generation Circuit.

### 33.2.4 LCD Boost Level Control Register (VLCD)

Address(es): 000A 0803h



Bit	Symbol	Bit Name	Description	R/W																																																																																																																																																
b0	VLCD[4:0]	Reference Voltage (Contrast Adjustment) Select		R/W																																																																																																																																																
b1				R/W																																																																																																																																																
b2				R/W																																																																																																																																																
b3				R/W																																																																																																																																																
b4				R/W																																																																																																																																																
			<table border="1"> <thead> <tr> <th>b4</th> <th>b3</th> <th>b2</th> <th>b1</th> <th>b0</th> <th>VL1 Voltage</th> <th colspan="2">VL4 Voltage</th> </tr> <tr> <th colspan="5"></th> <th>Reference Voltage</th> <th>1/3 Bias Method</th> <th>1/4 Bias Method</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1.00 V</td> <td>3.00 V</td> <td>4.00 V</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>1.05 V</td> <td>3.15 V</td> <td>4.20 V</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1.10 V</td> <td>3.30 V</td> <td>4.40 V</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1.15 V</td> <td>3.45 V</td> <td>4.60 V</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1.20 V</td> <td>3.60 V</td> <td>4.80 V</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1.25 V</td> <td>3.75 V</td> <td>5.00 V</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>1.30 V</td> <td>3.90 V</td> <td>5.20 V</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>1.35 V</td> <td>4.05 V</td> <td>Setting prohibited</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>1.40 V</td> <td>4.20 V</td> <td>Setting prohibited</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>1.45 V</td> <td>4.35 V</td> <td>Setting prohibited</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>1.50 V</td> <td>4.50 V</td> <td>Setting prohibited</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1.55 V</td> <td>4.65 V</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1.60 V</td> <td>4.80 V</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1.65 V</td> <td>4.95 V</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1.70 V</td> <td>5.10 V</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1.75 V</td> <td>5.25 V</td> <td>Setting prohibited</td> </tr> </tbody> </table>	b4	b3	b2	b1	b0	VL1 Voltage	VL4 Voltage							Reference Voltage	1/3 Bias Method	1/4 Bias Method	0	0	1	0	0	1.00 V	3.00 V	4.00 V	0	0	1	0	1	1.05 V	3.15 V	4.20 V	0	0	1	1	0	1.10 V	3.30 V	4.40 V	0	0	1	1	1	1.15 V	3.45 V	4.60 V	0	1	0	0	0	1.20 V	3.60 V	4.80 V	0	1	0	0	1	1.25 V	3.75 V	5.00 V	0	1	0	1	0	1.30 V	3.90 V	5.20 V	0	1	0	1	1	1.35 V	4.05 V	Setting prohibited	0	1	1	0	0	1.40 V	4.20 V	Setting prohibited	0	1	1	0	1	1.45 V	4.35 V	Setting prohibited	0	1	1	1	0	1.50 V	4.50 V	Setting prohibited	0	1	1	1	1	1.55 V	4.65 V	Setting prohibited	1	0	0	0	0	1.60 V	4.80 V	Setting prohibited	1	0	0	0	1	1.65 V	4.95 V	Setting prohibited	1	0	0	1	0	1.70 V	5.10 V	Setting prohibited	1	0	0	1	1	1.75 V	5.25 V	Setting prohibited	
b4	b3		b2	b1	b0	VL1 Voltage	VL4 Voltage																																																																																																																																													
					Reference Voltage	1/3 Bias Method	1/4 Bias Method																																																																																																																																													
0	0		1	0	0	1.00 V	3.00 V	4.00 V																																																																																																																																												
0	0		1	0	1	1.05 V	3.15 V	4.20 V																																																																																																																																												
0	0		1	1	0	1.10 V	3.30 V	4.40 V																																																																																																																																												
0	0		1	1	1	1.15 V	3.45 V	4.60 V																																																																																																																																												
0	1		0	0	0	1.20 V	3.60 V	4.80 V																																																																																																																																												
0	1		0	0	1	1.25 V	3.75 V	5.00 V																																																																																																																																												
0	1		0	1	0	1.30 V	3.90 V	5.20 V																																																																																																																																												
0	1		0	1	1	1.35 V	4.05 V	Setting prohibited																																																																																																																																												
0	1	1	0	0	1.40 V	4.20 V	Setting prohibited																																																																																																																																													
0	1	1	0	1	1.45 V	4.35 V	Setting prohibited																																																																																																																																													
0	1	1	1	0	1.50 V	4.50 V	Setting prohibited																																																																																																																																													
0	1	1	1	1	1.55 V	4.65 V	Setting prohibited																																																																																																																																													
1	0	0	0	0	1.60 V	4.80 V	Setting prohibited																																																																																																																																													
1	0	0	0	1	1.65 V	4.95 V	Setting prohibited																																																																																																																																													
1	0	0	1	0	1.70 V	5.10 V	Setting prohibited																																																																																																																																													
1	0	0	1	1	1.75 V	5.25 V	Setting prohibited																																																																																																																																													
			Settings other than above are prohibited.																																																																																																																																																	
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W																																																																																																																																																

- Note: The VLCD setting is valid only when the voltage boost circuit is operating.
- Note: Be sure to change the VLCD value after having stopped the operation of the voltage boost circuit (LCDM1.VLCON bit = 0).
- Note: To use the internal voltage boosting method, specify the reference voltage by using the VLCD register (select the internal boosting method (by setting the LCDM0.MDSET[1:0] bits to 01b) if the default reference voltage is used), wait for the reference voltage setup time (5 ms (min.)), and then set the LCDM1.VLCON bit to 1.

VLCD selects the reference voltage that is to be generated when operating the voltage boost circuit (contrast adjustment). The reference voltage can be selected from 16 steps. For details on the operating voltage range, refer to section 42, Electrical Characteristics.

### 33.3 LCD Display Data Registers

The LCD display data registers are mapped as shown in Table 33.6 and Table 33.7. The contents displayed on the LCD can be changed by changing the contents of the LCD display data registers.

**Table 33.6 Relationship Between LCD Display Data Register Contents and Segment/Common Outputs**

Other than 8-time-slice (static, 2-time slice, 3-time slice, and 4-time slice)

Register Name	Address	b7	b6	b5	b4	b3	b2	b1	b0	After Reset	100-pin	64-pin
		COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0			
SEG00	000A 0840h	SEG00 (B-pattern area)				SEG00 (A-pattern area)				00h	A	A
SEG01	000A 0841h	SEG01 (B-pattern area)				SEG01 (A-pattern area)				00h	A	A
SEG02	000A 0842h	SEG02 (B-pattern area)				SEG02 (A-pattern area)				00h	A	A
SEG03	000A 0843h	SEG03 (B-pattern area)				SEG03 (A-pattern area)				00h	A	A
SEG04	000A 0844h	SEG04 (B-pattern area)				SEG04 (A-pattern area)				00h	A	N/A
SEG05	000A 0845h	SEG05 (B-pattern area)				SEG05 (A-pattern area)				00h	A	N/A
SEG06	000A 0846h	SEG06 (B-pattern area)				SEG06 (A-pattern area)				00h	A	A
SEG07	000A 0847h	SEG07 (B-pattern area)				SEG07 (A-pattern area)				00h	A	A
SEG08	000A 0848h	SEG08 (B-pattern area)				SEG08 (A-pattern area)				00h	A	A
SEG09	000A 0849h	SEG09 (B-pattern area)				SEG09 (A-pattern area)				00h	A	N/A
SEG10	000A 084Ah	SEG10 (B-pattern area)				SEG10 (A-pattern area)				00h	A	N/A
SEG11	000A 084Bh	SEG11 (B-pattern area)				SEG11 (A-pattern area)				00h	A	A
SEG12	000A 084Ch	SEG12 (B-pattern area)				SEG12 (A-pattern area)				00h	A	A
SEG13	000A 084Dh	SEG13 (B-pattern area)				SEG13 (A-pattern area)				00h	A	N/A
SEG14	000A 084Eh	SEG14 (B-pattern area)				SEG14 (A-pattern area)				00h	A	N/A
SEG15	000A 084Fh	SEG15 (B-pattern area)				SEG15 (A-pattern area)				00h	A	N/A
SEG16	000A 0850h	SEG16 (B-pattern area)				SEG16 (A-pattern area)				00h	A	A
SEG17	000A 0851h	SEG17 (B-pattern area)				SEG17 (A-pattern area)				00h	A	A
SEG18	000A 0852h	SEG18 (B-pattern area)				SEG18 (A-pattern area)				00h	A	A
SEG19	000A 0853h	SEG19 (B-pattern area)				SEG19 (A-pattern area)				00h	A	A
SEG20	000A 0854h	SEG20 (B-pattern area)				SEG20 (A-pattern area)				00h	A	A
SEG21	000A 0855h	SEG21 (B-pattern area)				SEG21 (A-pattern area)				00h	A	A
SEG22	000A 0856h	SEG22 (B-pattern area)				SEG22 (A-pattern area)				00h	A	A
SEG23	000A 0857h	SEG23 (B-pattern area)				SEG23 (A-pattern area)				00h	A	A
SEG24	000A 0858h	SEG24 (B-pattern area)				SEG24 (A-pattern area)				00h	A	N/A
SEG25	000A 0859h	SEG25 (B-pattern area)				SEG25 (A-pattern area)				00h	A	N/A
SEG26	000A 085Ah	SEG26 (B-pattern area)				SEG26 (A-pattern area)				00h	A	N/A
SEG27	000A 085Bh	SEG27 (B-pattern area)				SEG27 (A-pattern area)				00h	A	N/A
SEG28	000A 085Ch	SEG28 (B-pattern area)				SEG28 (A-pattern area)				00h	A	N/A
SEG29	000A 085Dh	SEG29 (B-pattern area)				SEG29 (A-pattern area)				00h	A	A
SEG30	000A 085Eh	SEG30 (B-pattern area)				SEG30 (A-pattern area)				00h	A	N/A
SEG31	000A 085Fh	SEG31 (B-pattern area)				SEG31 (A-pattern area)				00h	A	N/A
SEG32	000A 0860h	SEG32 (B-pattern area)				SEG32 (A-pattern area)				00h	A	N/A
SEG33	000A 0861h	SEG33 (B-pattern area)				SEG33 (A-pattern area)				00h	A	A
SEG34	000A 0862h	SEG34 (B-pattern area)				SEG34 (A-pattern area)				00h	A	A
SEG35	000A 0863h	SEG35 (B-pattern area)				SEG35 (A-pattern area)				00h	A	A
SEG36	000A 0864h	SEG36 (B-pattern area)				SEG36 (A-pattern area)				00h	A	A
SEG37	000A 0865h	SEG37 (B-pattern area)				SEG37 (A-pattern area)				00h	A	A
SEG38	000A 0866h	SEG38 (B-pattern area)				SEG38 (A-pattern area)				00h	A	A
SEG39	000A 0867h	SEG39 (B-pattern area)				SEG39 (A-pattern area)				00h	A	A

A: Available, N/A: Not available

**Table 33.7 Relationship Between LCD Display Data Register Contents and Segment/Common Outputs**

8-time-slice

Register Name	Address	b7	b6	b5	b4	b3	b2	b1	b0	After Reset	100-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0			
SEG00	000A 0840h	SEG00								00h	A	A
SEG01	000A 0841h	SEG01								00h	A	A
SEG02	000A 0842h	SEG02								00h	A	A
SEG03	000A 0843h	SEG03								00h	A	A
SEG04	000A 0844h	SEG04								00h	A	N/A
SEG05	000A 0845h	SEG05								00h	A	N/A
SEG06	000A 0846h	SEG06								00h	A	A
SEG07	000A 0847h	SEG07								00h	A	A
SEG08	000A 0848h	SEG08								00h	A	A
SEG09	000A 0849h	SEG09								00h	A	N/A
SEG10	000A 084Ah	SEG10								00h	A	N/A
SEG11	000A 084Bh	SEG11*1								00h	N/A	N/A
SEG12	000A 084Ch	SEG12*1								00h	N/A	N/A
SEG13	000A 084Dh	SEG13*1								00h	N/A	N/A
SEG14	000A 084Eh	SEG14								00h	A	N/A
SEG15	000A 084Fh	SEG15*1								00h	N/A	N/A
SEG16	000A 0850h	SEG16								00h	A	A
SEG17	000A 0851h	SEG17								00h	A	A
SEG18	000A 0852h	SEG18								00h	A	A
SEG19	000A 0853h	SEG19								00h	A	A
SEG20	000A 0854h	SEG20								00h	A	A
SEG21	000A 0855h	SEG21								00h	A	A
SEG22	000A 0856h	SEG22								00h	A	A
SEG23	000A 0857h	SEG23								00h	A	A
SEG24	000A 0858h	SEG24								00h	A	N/A
SEG25	000A 0859h	SEG25								00h	A	N/A
SEG26	000A 085Ah	SEG26								00h	A	N/A
SEG27	000A 085Bh	SEG27								00h	A	N/A
SEG28	000A 085Ch	SEG28								00h	A	N/A
SEG29	000A 085Dh	SEG29								00h	A	A
SEG30	000A 085Eh	SEG30								00h	A	N/A
SEG31	000A 085Fh	SEG31								00h	A	N/A
SEG32	000A 0860h	SEG32								00h	A	N/A
SEG33	000A 0861h	SEG33								00h	A	A
SEG34	000A 0862h	SEG34								00h	A	A
SEG35	000A 0863h	SEG35								00h	A	A
SEG36	000A 0864h	SEG36								00h	A	A
SEG37	000A 0865h	SEG37								00h	A	A
SEG38	000A 0866h	SEG38								00h	A	A
SEG39	000A 0867h	SEG39								00h	A	A

A: Available, N/A: Not available

Note 1. The COM4 to COM7, SEG11 to SEG13, and SEG15 pins are used alternatively.

To use the LCD display data register when the number of time slices is static, two, three, or four, the lower 4 bits and higher 4 bits of each address of the LCD display data register become an A-pattern area and a B-pattern area, respectively.

The correspondences between A-pattern area data and COM signals are as follows: bit 0 ↔ COM0, bit 1 ↔ COM1, bit 2 ↔ COM2, and bit 3 ↔ COM3.

The correspondences between B-pattern area data and COM signals are as follows: bit 4 ↔ COM0, bit 5 ↔ COM1, bit 6 ↔ COM2, and bit 7 ↔ COM3.

A-pattern area data will be displayed on the LCD panel when LCDM1.BLON bit = LCDM1.LCDSEL bit = 0 has been selected, and B-pattern area data will be displayed on the LCD panel when LCDM1.BLON bit = 0 and LCDM1.LCDSEL bit = 1 have been selected.

### 33.4 Selection of LCD Display Data Register

With this MCU, to use the LCD display data registers when the number of time slices is static, two, three, or four, the LCD display data register can be selected from the following three types, according to the LCDM1.BLON and LCDSEL bit settings.

- Displaying an A-pattern area data (lower 4 bits of LCD display data register)
- Displaying a B-pattern area data (higher 4 bits of LCD display data register)
- Alternately displaying A-pattern and B-pattern area data (blinking display corresponding to the periodic interrupt timing of the real-time clock (RTC))

Note: When 8-time slice is used, LCD display data registers (A-pattern, B-pattern, or blinking display) cannot be selected.

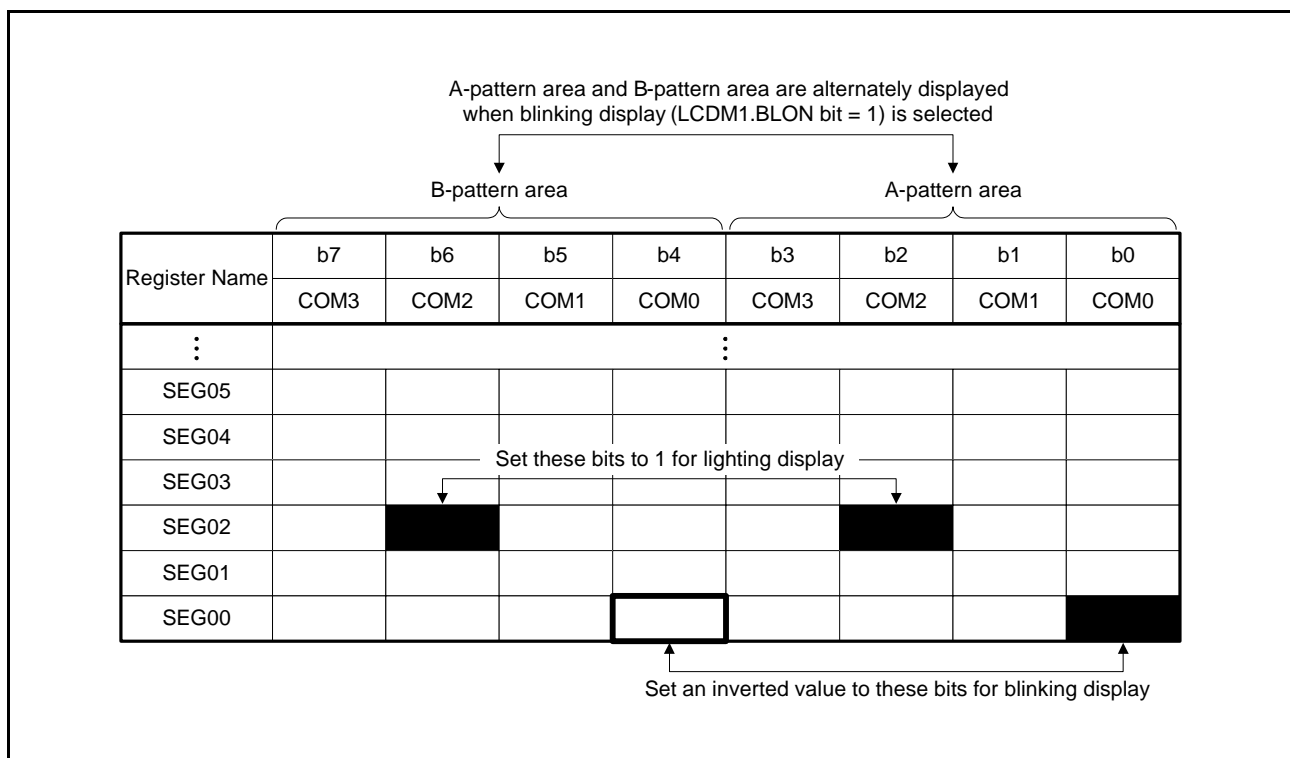


Figure 33.2 Example of Setting LCD Display Data Registers When Pattern is Changed

### 33.4.1 A-Pattern Area and B-pattern Area Data Display

When LCDM1.BLON bit = LCDM1.LCDSEL bit = 0, A-pattern area (lower 4 bits of the LCD display data register) data will be output as the display data.

When LCDM1.BLON bit = 0, and LCDM1.LCDSEL bit = 1, B-pattern area (higher 4 bits of the LCD display data register) data will be output as the display data.

For details on the display area, refer to section 33.3, LCD Display Data Registers.

### 33.4.2 Blinking Display (Alternately Displaying A-Pattern and B-Pattern Area Data)

When LCDM1.BLON bit = 1 has been set, A-pattern and B-pattern area data will be alternately displayed, according to the periodic interrupt (PRD) timing of the real-time clock (RTC). For the periodic interrupt (0.5-second setting only) timing of the RTC, refer to section 24, Realtime Clock (RTCA).

Figure 33.2 shows an example of setting LCD display data registers when a pattern is changed. For blinking display of the LCD, set inverted values to the B-pattern area bits corresponding to the A-pattern area bits. (Example: Write 1 to bit 0 of the SEG00 register, and set 0 to bit 4 of the SEG00 register.) When not setting blinking display of the LCD, set the same values. (Example: Write 1 to bit 2 of the SEG02 register, and write 1 to bit 6 of the SEG02 register.)

For details on the display area, refer to section 33.3, LCD Display Data Registers.

Next, the timing operation of display switching is shown.

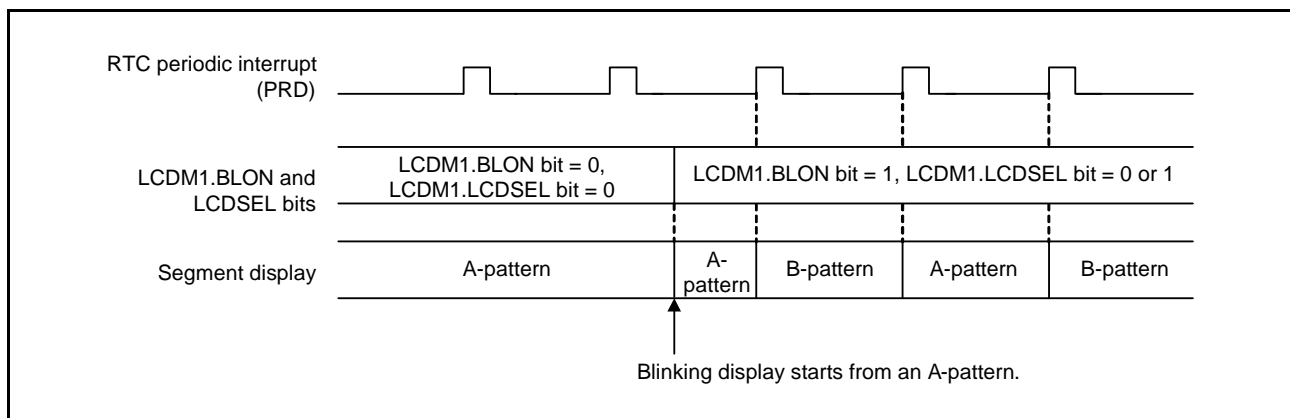


Figure 33.3 Switching Operation from A-Pattern Display to Blinking Display

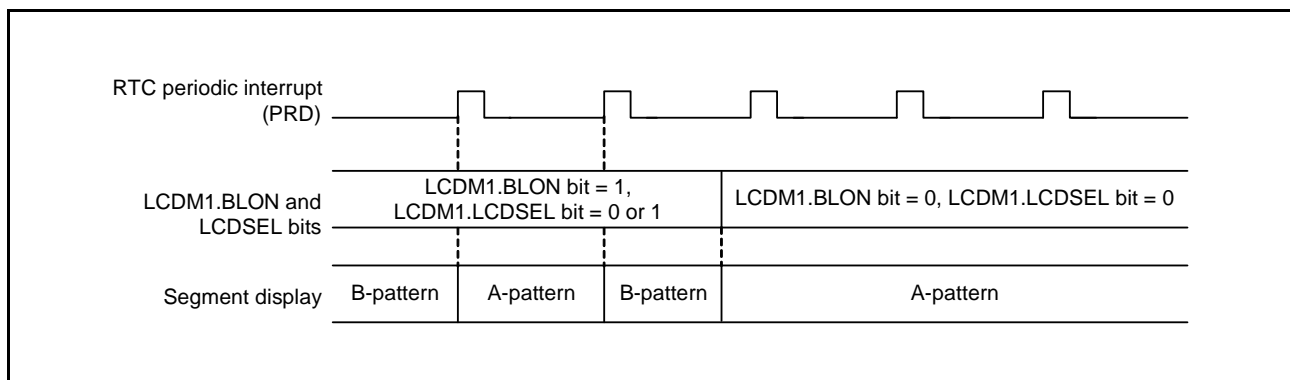


Figure 33.4 Switching Operation from Blinking Display to A-Pattern Display



### 33.5 Setting the LCD Controller/Driver

Set the LCD controller/driver using the following procedure.

Note: To operate the LCD controller/driver, be sure to follow procedures (1) to (3). Unless these procedures are observed, the operation will not be guaranteed.

Note: The steps shown in the flowcharts in (1) to (3) are performed by the CPU.

#### (1) External resistance division method

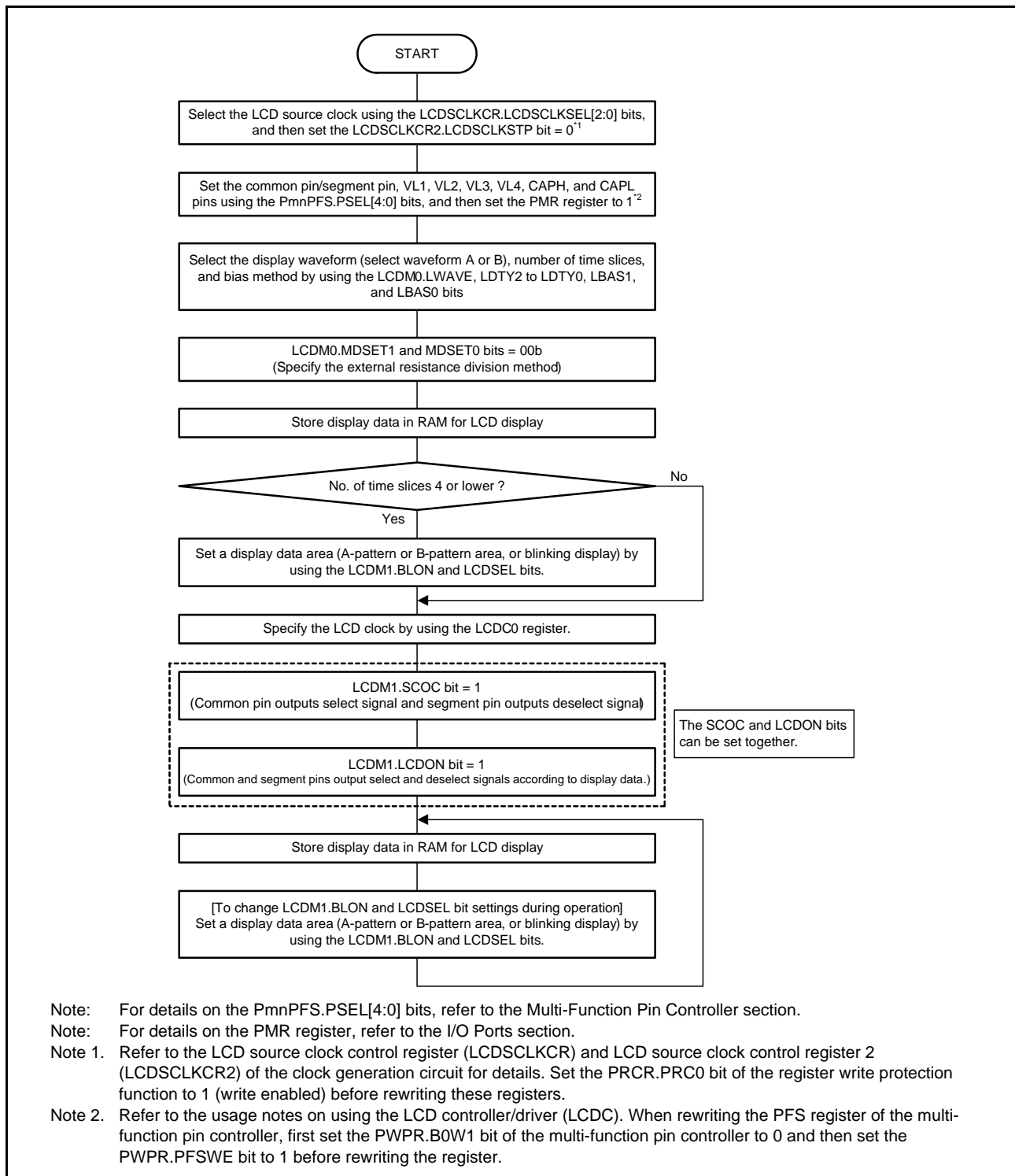


Figure 33.5 External Resistance Division Method Setting Procedure

## (2) Internal voltage boosting method

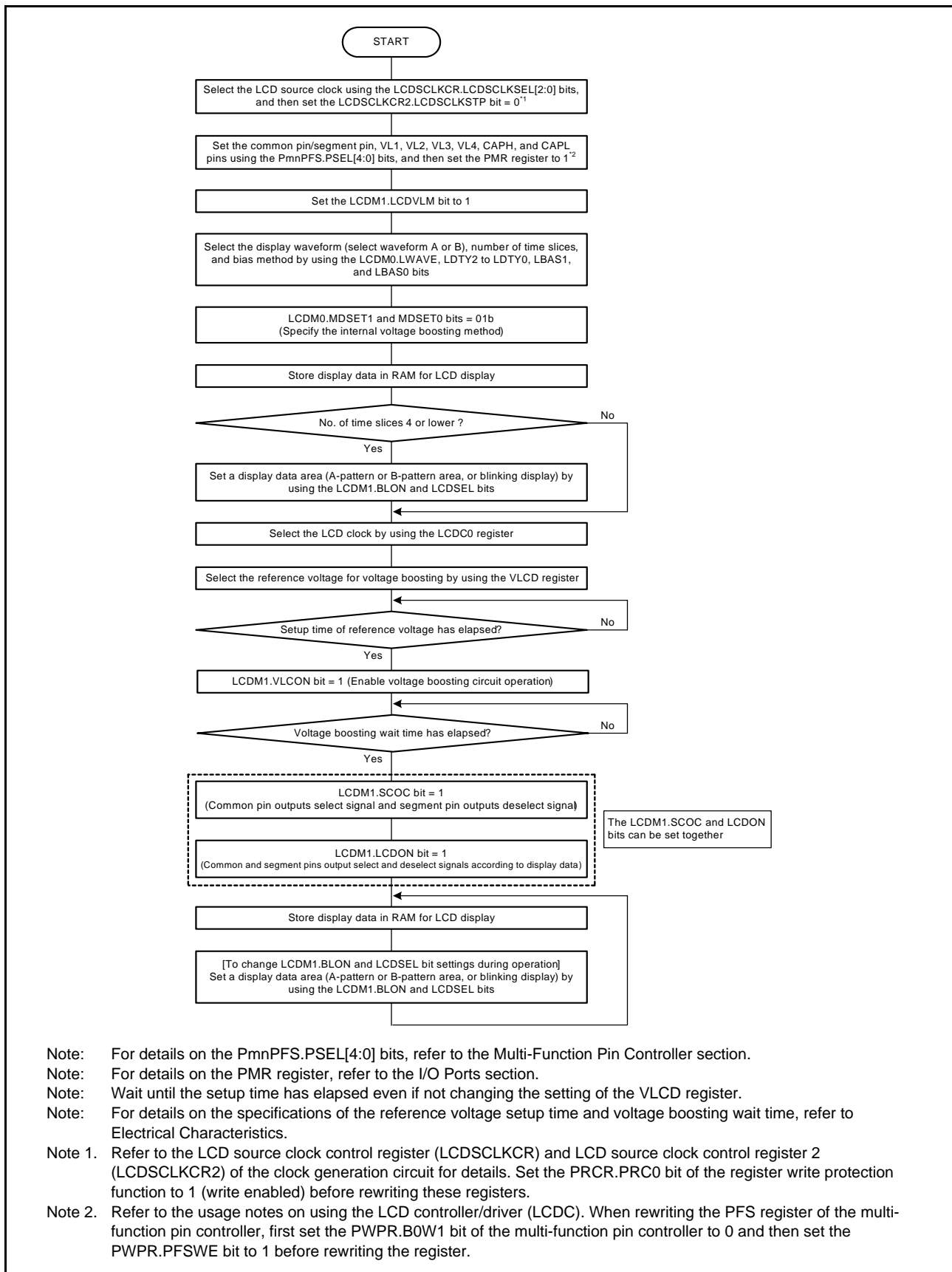


Figure 33.6 Internal Voltage Boosting Method Setting Procedure

(3) Capacitor split method

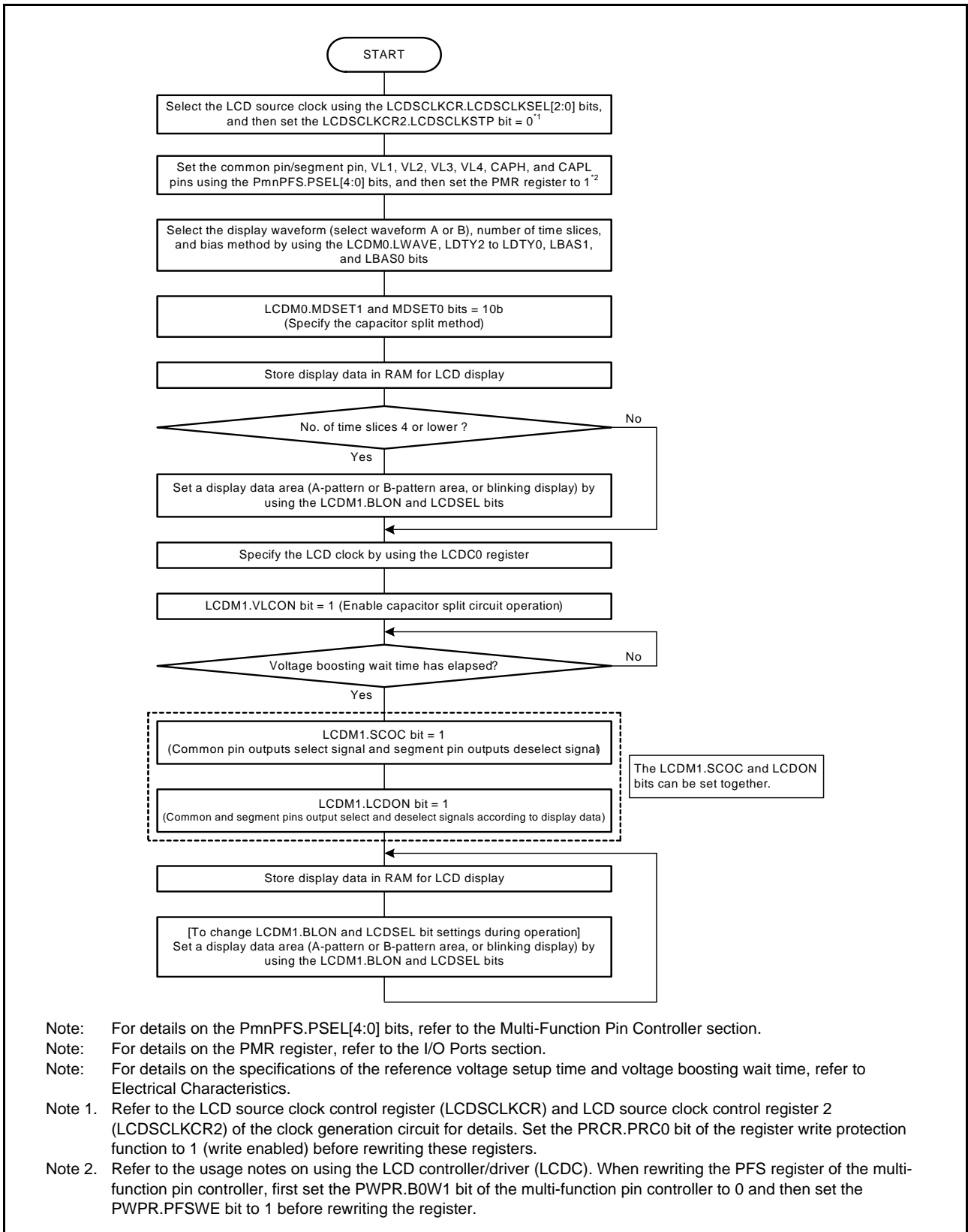


Figure 33.7 Capacitor Split Method Setting Procedure

### 33.6 Operation Stop Procedure

To stop the operation of the LCD while it is displaying waveforms, follow the steps shown in the flowchart below. The LCD stops operating when the LCDM1.LCDON and SCOC bits are set to 0.

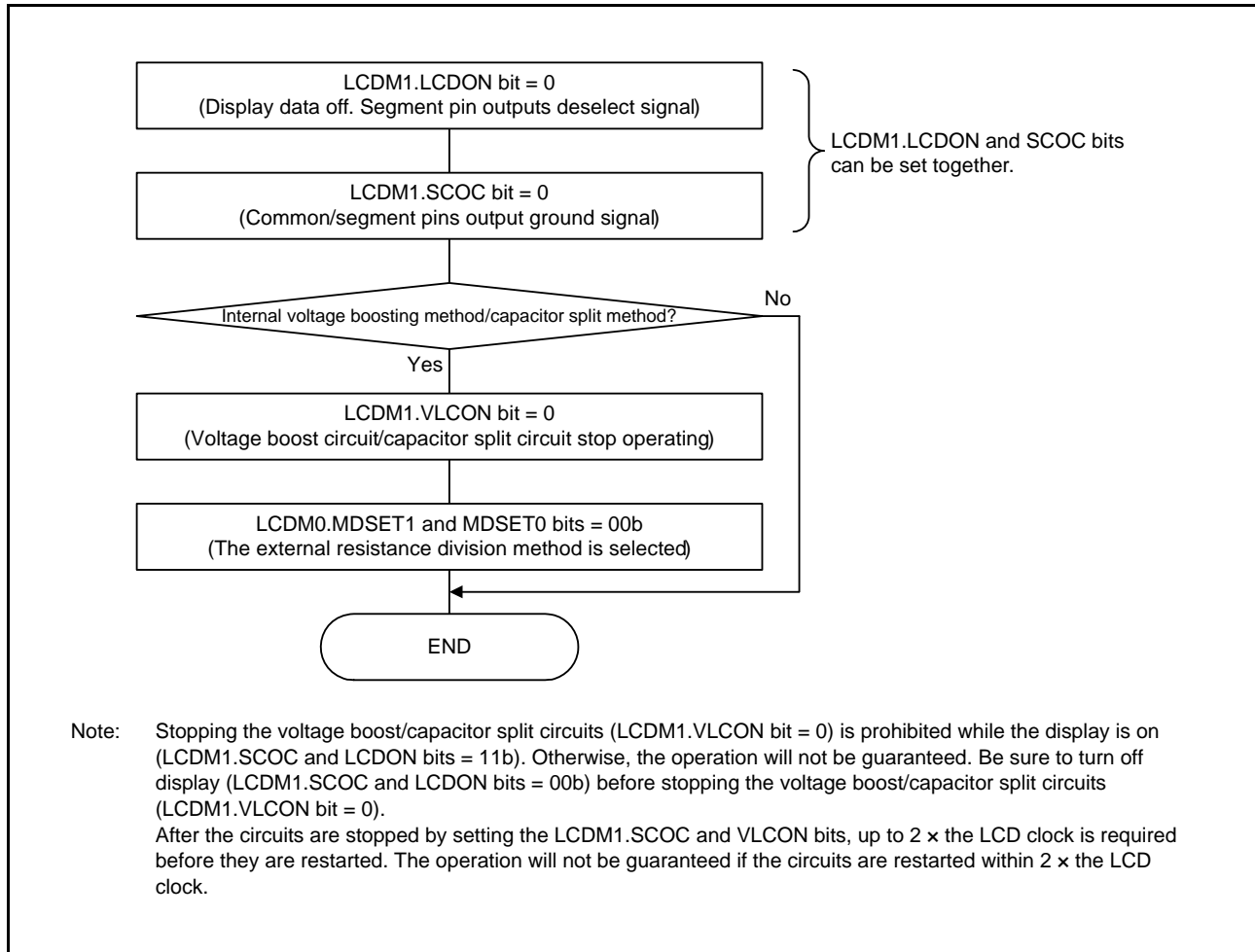


Figure 33.8 Operation Stop Procedure

### 33.7 Supplying LCD Drive Voltages VL1, VL2, VL3, and VL4

#### 33.7.1 External Resistance Division Method

Figure 33.9 and Figure 33.10 show examples of LCD drive power supply connection, corresponding to each bias method.

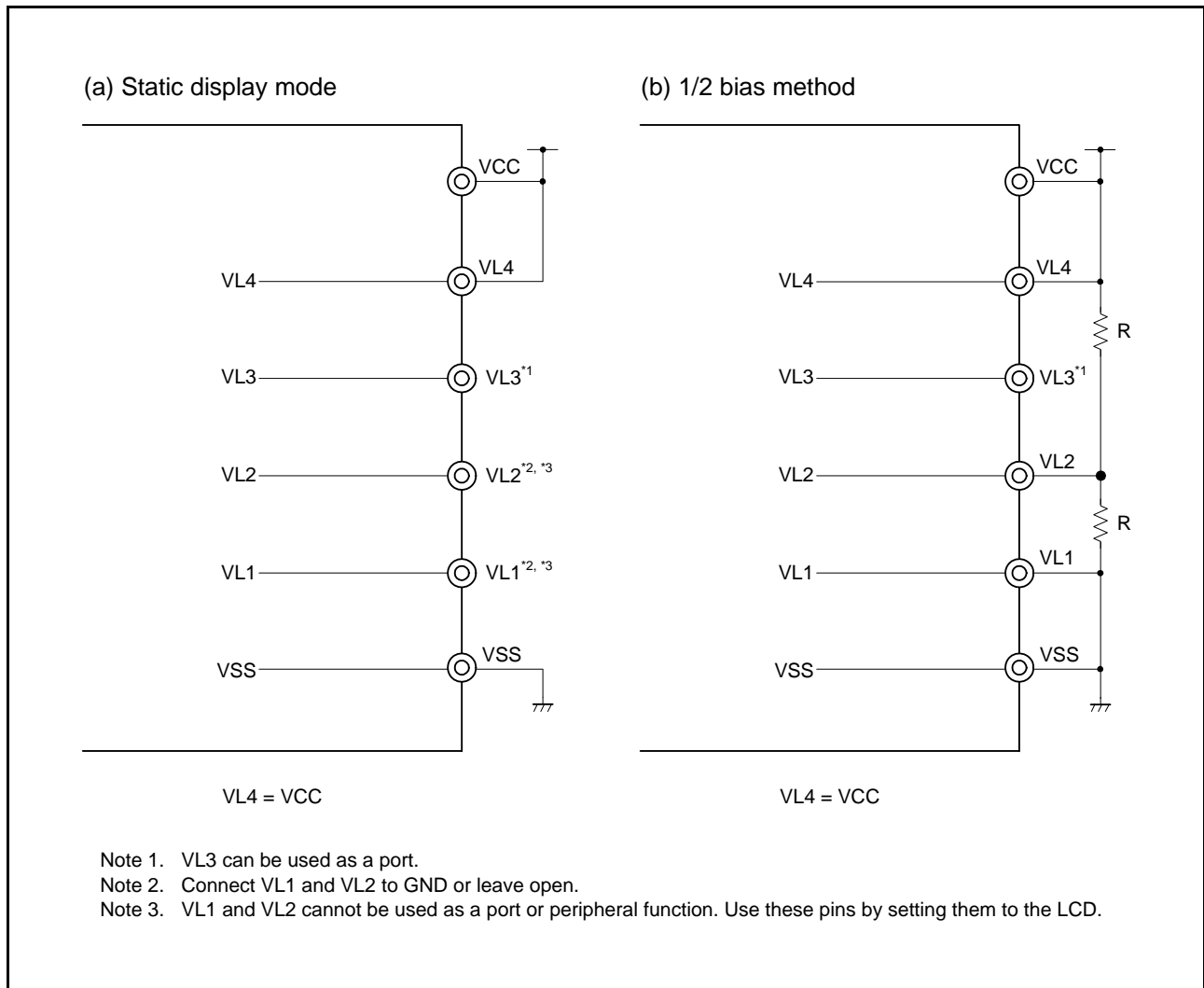
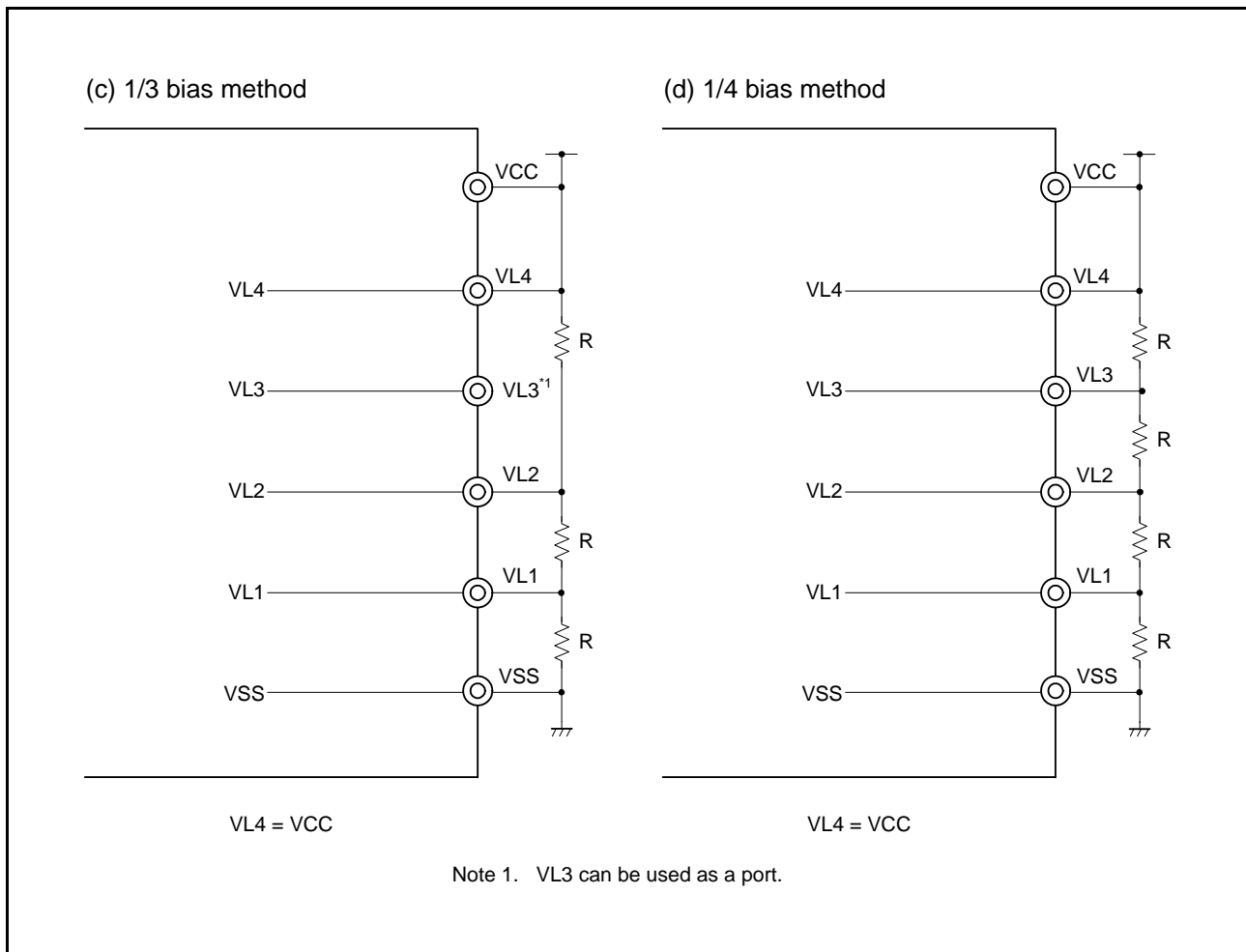


Figure 33.9 Examples of LCD Drive Power Connections (External Resistance Division Method) (1/2)



**Figure 33.10** Examples of LCD Drive Power Connections (External Resistance Division Method) (2/2)

**Note:** The reference resistance “R” value for external resistance division is 10 kΩ to 1 MΩ. In addition, to stabilize the potential of the VL1 to VL4 pins, connect a capacitor between each of pins VL1 to VL4 and the GND pin as needed. The reference capacitance is about 0.22 μF but it depends on the LCD panel used, the number of segment pins, the number of common pins, the frame frequency, and the operating environment. Thoroughly evaluate these values in accordance with your system and adjust and determine the capacitance.

### 33.7.2 Internal Voltage Boosting Method

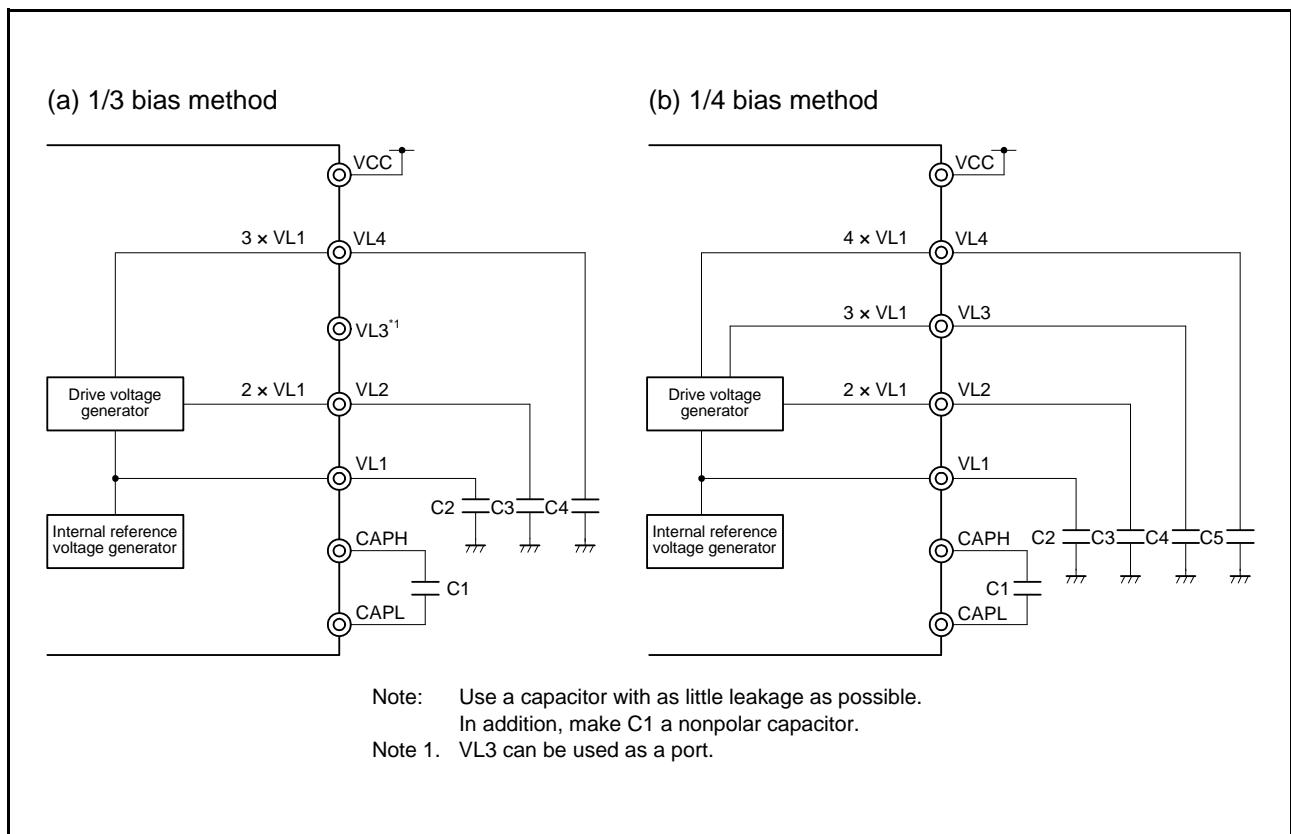
This MCU contains an internal voltage boost circuit for generating LCD drive power supplies. The internal voltage boost circuit and external capacitors ( $0.47 \mu\text{F} \pm 30\%$ ) are used to generate an LCD drive voltage. Only 1/3 bias mode or 1/4 bias mode can be set for the internal voltage boosting method.

The LCD drive voltage of the internal voltage boosting method can supply a constant voltage, regardless of changes in VCC, because it is a power supply separate from the main unit.

In addition, a contrast can be adjusted by using the LCD boost level control register (VLCD).

**Table 33.8 LCD Drive Voltages (Internal Voltage Boosting Method)**

LCD Drive Voltage Pin	Bias Method	
	1/3 Bias Method	1/4 Bias Method
VL4	$3 \times \text{VL1}$	$4 \times \text{VL1}$
VL3	—	$3 \times \text{VL1}$
VL2	$2 \times \text{VL1}$	$2 \times \text{VL1}$
VL1	LCD reference voltage	LCD reference voltage



**Figure 33.11 Examples of LCD Drive Power Connections (Internal Voltage Boosting Method)**

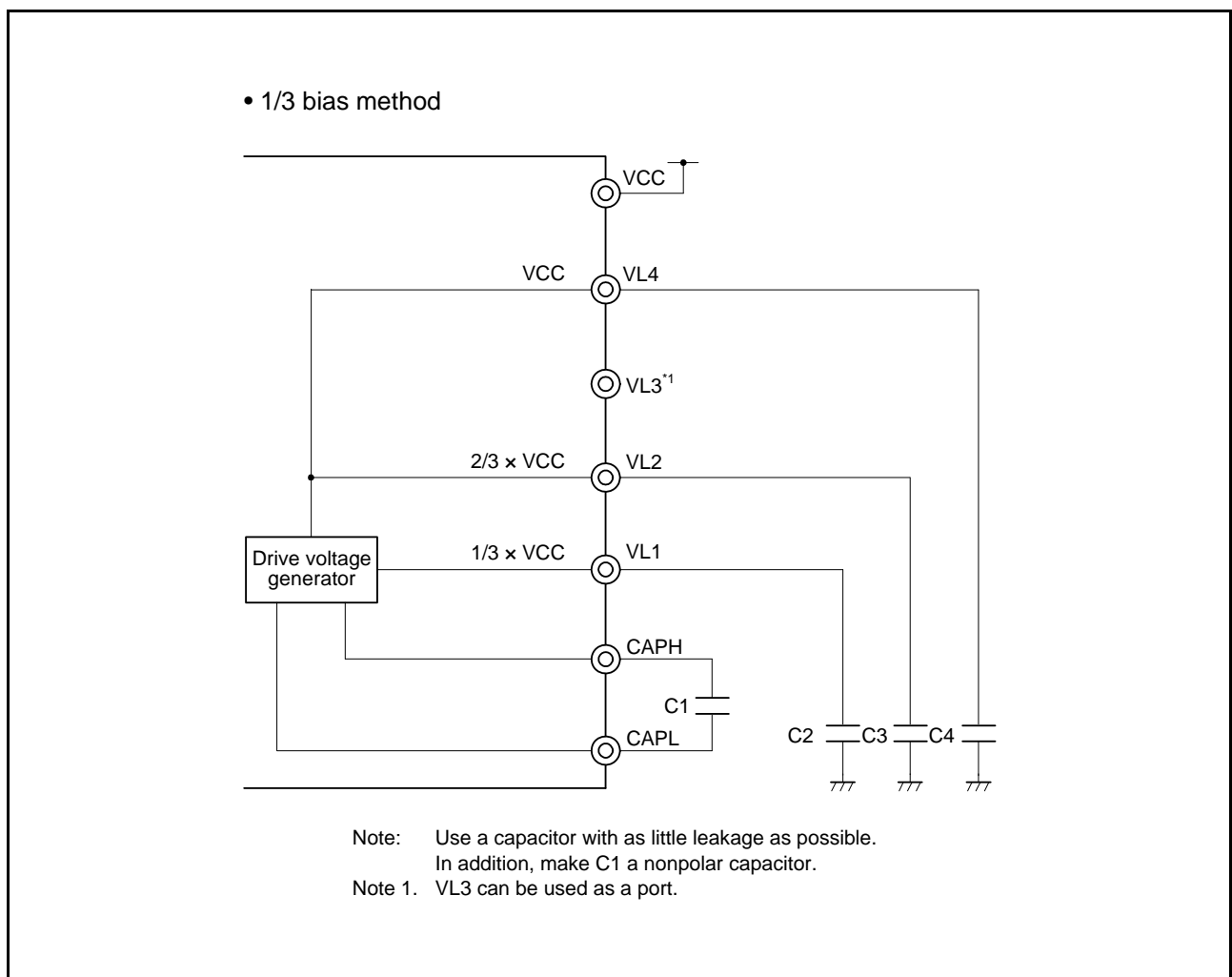
### 33.7.3 Capacitor Split Method

This MCU contains an internal voltage reduction circuit for generating LCD drive power supplies. The internal voltage reduction circuit and external capacitors ( $0.47 \mu\text{F} \pm 30\%$ ) are used to generate an LCD drive voltage. Only 1/3 bias mode can be set for the capacitor split method.

Different from the external resistance division method, there is no current flowing with the capacitor split method, so current consumption can be reduced.

**Table 33.9 LCD Drive Voltages (Capacitor Split Method)**

LCD Drive Voltage Pin	Bias Method	1/3 Bias Method
VL4		VCC
VL3		—
VL2		$2/3 \times \text{VL4}$
VL1		$1/3 \times \text{VL4}$



**Figure 33.12 Examples of LCD Drive Power Connections (Capacitor Split Method)**



### 33.8 Common and Segment Signals

Each pixel of the LCD panel turns on when the potential difference between the corresponding common and segment signals becomes higher than a specific voltage (LCD drive voltage, VLCD). The pixels turn off when the potential difference becomes lower than VLCD.

Applying DC voltage to the common and segment signals of an LCD panel causes deterioration. To avoid this problem, this LCD panel is driven by AC voltage.

(1) Common signals

Each common signal is selected sequentially according to a specified number of time slices at the timing listed in Table 33.10. In the static display mode, the same signal is output to COM0 to COM3.

In the two-time-slice mode, leave the COM2 and COM3 pins open. In the three-time-slice mode, leave the COM3 pin open.

Use the COM4 to COM7 pins other than in the eight-time-slice mode as open or segment pins.

**Table 33.10 COM Signal**

COM Signal	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
Static display mode					*1	*1	*1	*1
Two-time-slice mode			Open	Open	*1	*1	*1	*1
Three-time-slice mode				Open	*1	*1	*1	*1
Four-time-slice mode					*1	*1	*1	*1
Eight-time-slice mode								

Note 1. Use the pins as open or segment pins.

## (2) Segment signals

The segment signals correspond to the LCD display data register (refer to section 33.3, LCD Display Data Registers). When the number of time slices is eight, bits 0 to 7 of each display data register are read in synchronization with the common signals COM0 to COM7, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG00 to SEG10, SEG14, and SEG16 to SEG39).

When the number of time slices is number other than eight, bits 0 to 3 of each byte in A-pattern area are read in synchronization with COM0 to COM3, and bits 4 to 7 of each byte in B-pattern area are read in synchronization with COM0 to COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG00 to SEG39).

Check, with the information given above, what combination of front-surface electrodes (corresponding to the segment signals) and rear-surface electrodes (corresponding to the common signals) forms display patterns in the LCD display data register, and write the bit data that corresponds to the desired display pattern on a one-to-one basis.

Note: The mounted segment output pins vary depending on the product.

- 64-pin products: SEG00 to SEG03, SEG06 to SEG08, SEG11, SEG12, SEG16 to SEG23, SEG29, SEG33 to SEG39
- 100-pin products: SEG00 to SEG39

## (3) Output waveforms of common and segment signals

The voltages listed in Table 33.11 are output as common and segment signals.

When both common and segment signals are at the select voltage, a display on-voltage of  $\pm$ VLCD is obtained. The other combinations of the signals correspond to the display off-voltage.

**Table 33.11 LCD Drive Voltage**

(a) Static display mode

Segment Signal		Select Signal Level	Deselect Signal Level
		Common Signal	
VL4/VSS		-VLCD/+VLCD	0 V/0 V

(b) 1/2 bias method

Segment Signal		Select Signal Level	Deselect Signal Level
		Common Signal	
VL4/VSS		-VLCD/+VLCD	0 V/0 V
Select Signal Level	VL4/VSS	-VLCD/+VLCD	0 V/0 V
Deselect Signal Level	VL2	$-\frac{1}{2}$ VLCD/+ $\frac{1}{2}$ VLCD	$+\frac{1}{2}$ VLCD/- $\frac{1}{2}$ VLCD

(c) 1/3 bias method (waveform A or B)

Segment Signal		Select Signal Level	Deselect Signal Level
		Common Signal	
VL4/VSS		-VLCD/+VLCD	$-\frac{1}{3}$ VLCD/+ $\frac{1}{3}$ VLCD
Select Signal Level	VL4/VSS	-VLCD/+VLCD	$-\frac{1}{3}$ VLCD/+ $\frac{1}{3}$ VLCD
Deselect Signal Level	VL1/VL2	$-\frac{1}{3}$ VLCD/+ $\frac{1}{3}$ VLCD	$+\frac{1}{3}$ VLCD/- $\frac{1}{3}$ VLCD

(d) 1/4 bias method (waveform A or B)

Segment Signal		Select Signal Level	Deselect Signal Level
		Common Signal	
VL4/VSS		-VLCD/+VLCD	$-\frac{1}{2}$ VLCD/+ $\frac{1}{2}$ VLCD
Select Signal Level	VL4/VSS	-VLCD/+VLCD	$-\frac{1}{2}$ VLCD/+ $\frac{1}{2}$ VLCD
Deselect Signal Level	VL1/VL3	$-\frac{1}{4}$ VLCD/+ $\frac{1}{4}$ VLCD	$+\frac{1}{4}$ VLCD/- $\frac{1}{4}$ VLCD

Figure 33.13 and Figure 33.14 show the common signal waveforms, and Figure 33.15 to Figure 33.17 show the voltages and phases of the common and segment signals.

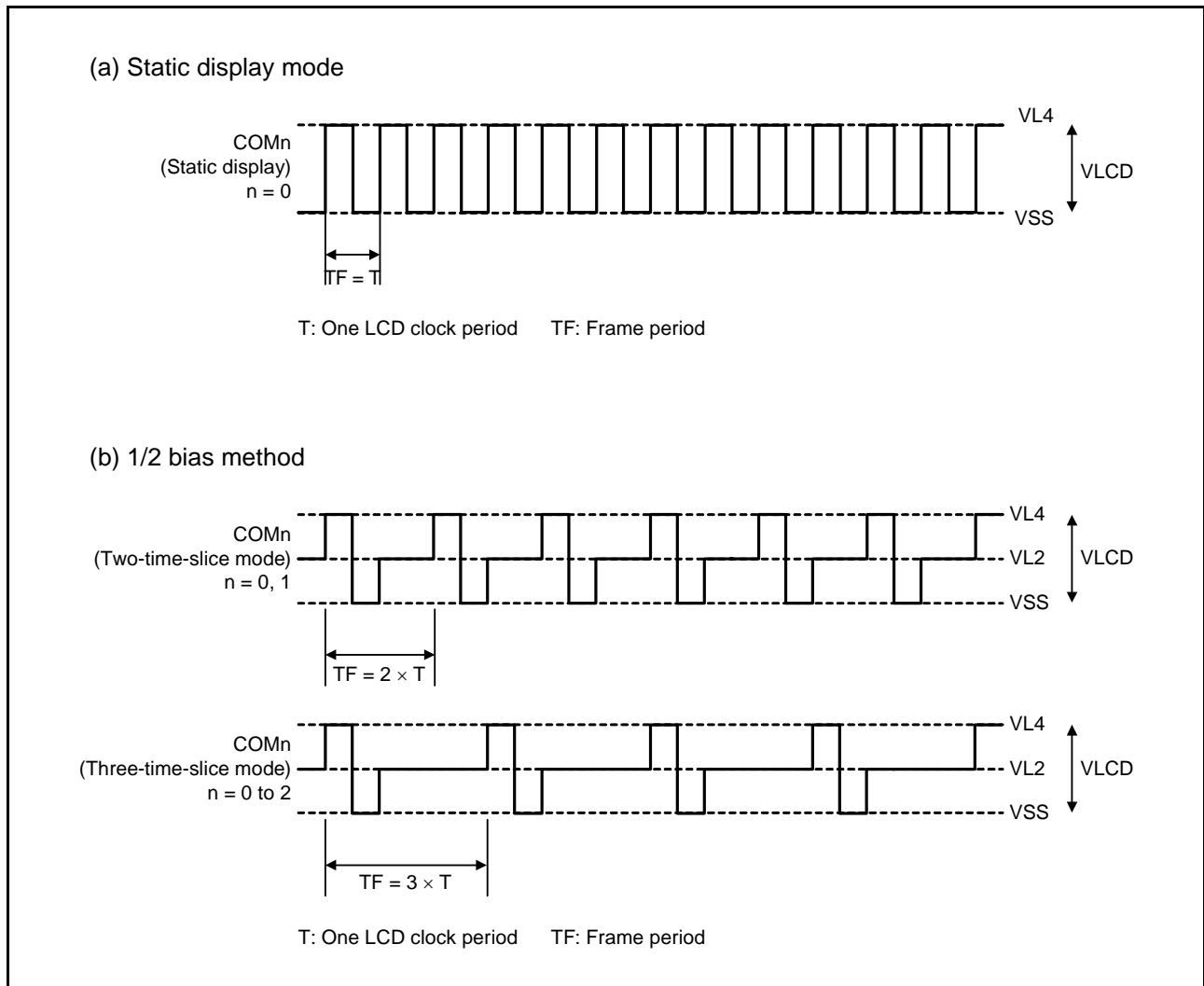


Figure 33.13 Common Signal Waveforms (1/2)

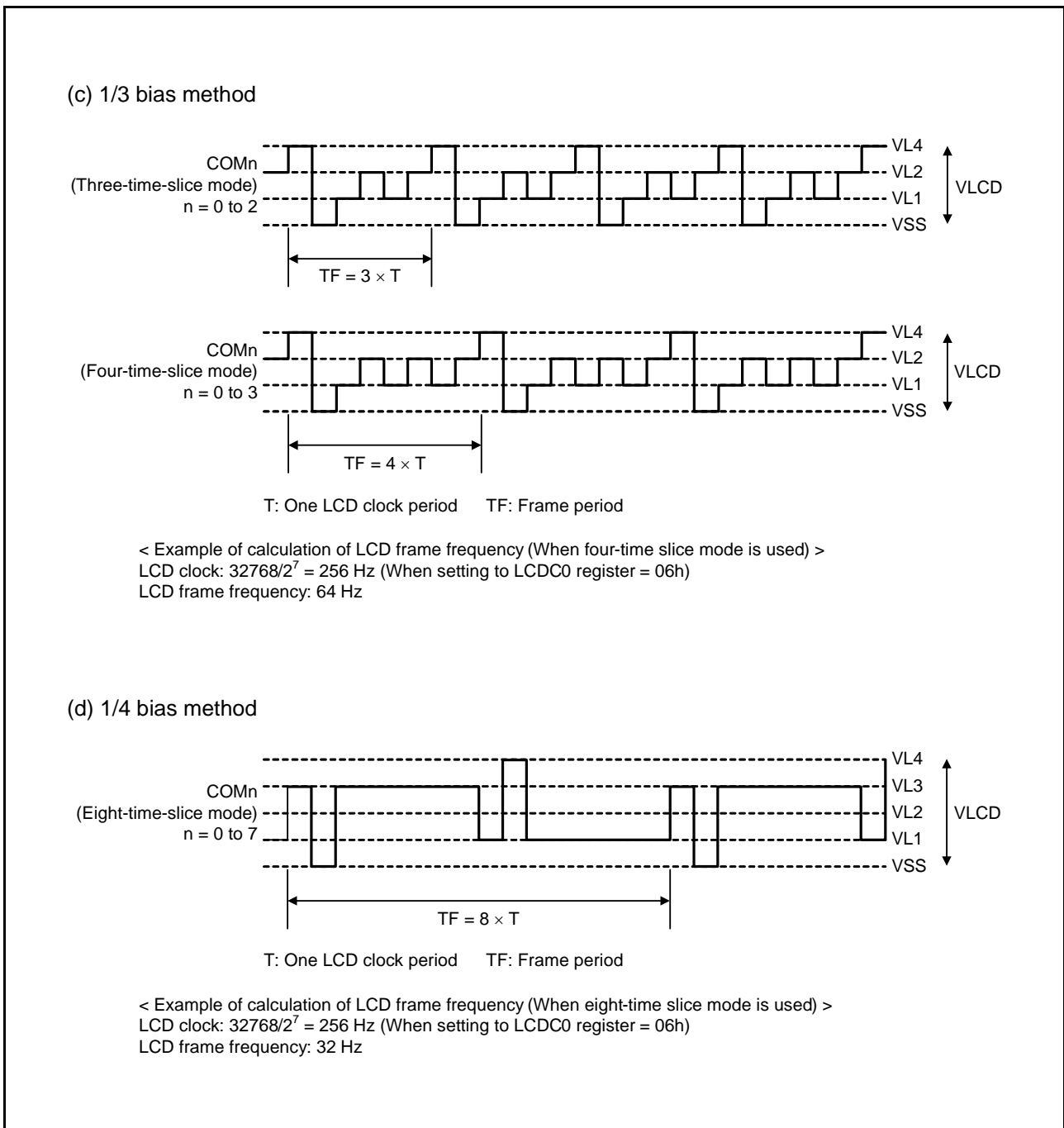


Figure 33.14 Common Signal Waveforms (2/2)

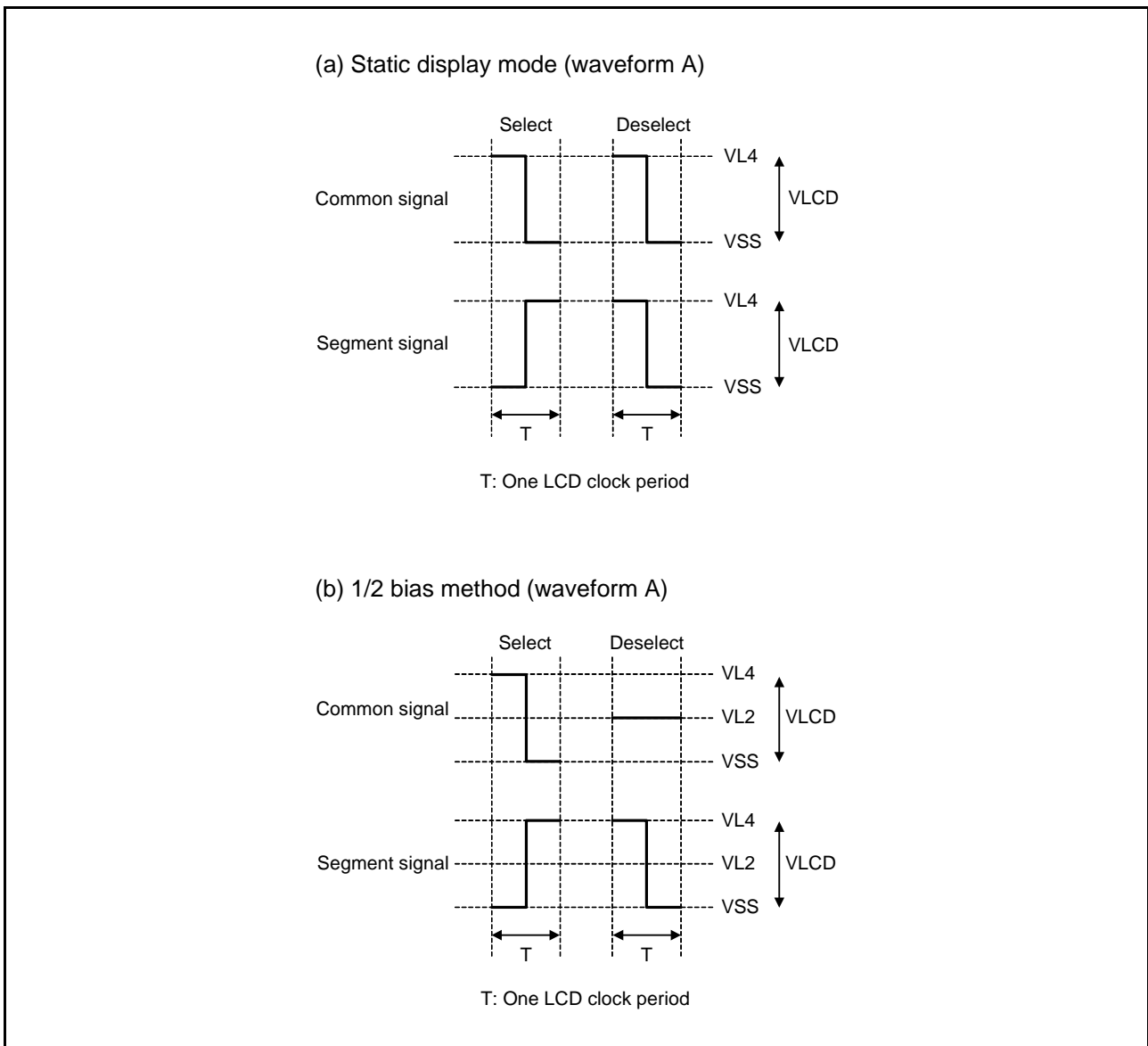


Figure 33.15 Voltages and Phases of Common and Segment Signals (1/3)

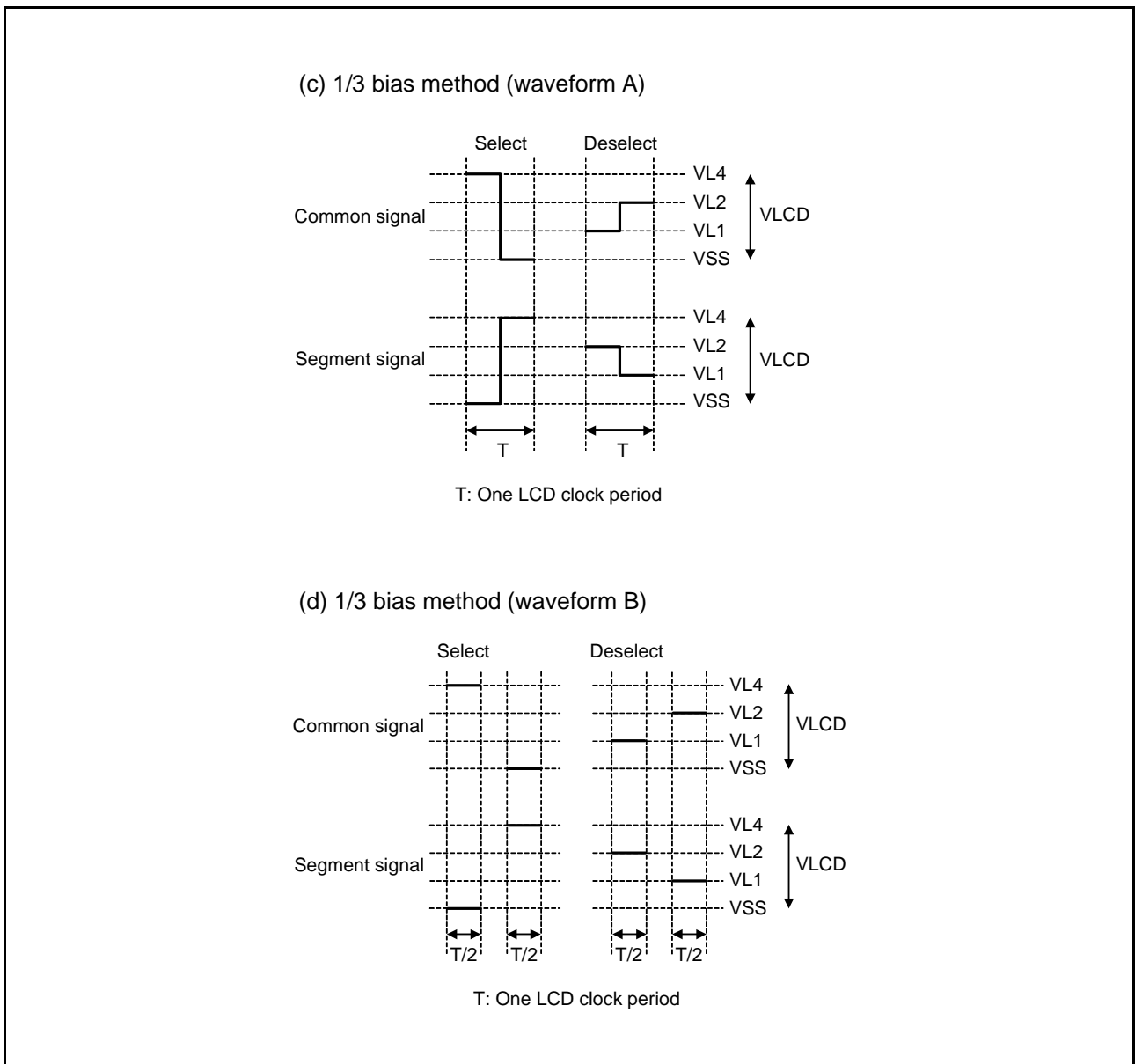


Figure 33.16 Voltages and Phases of Common and Segment Signals (2/3)

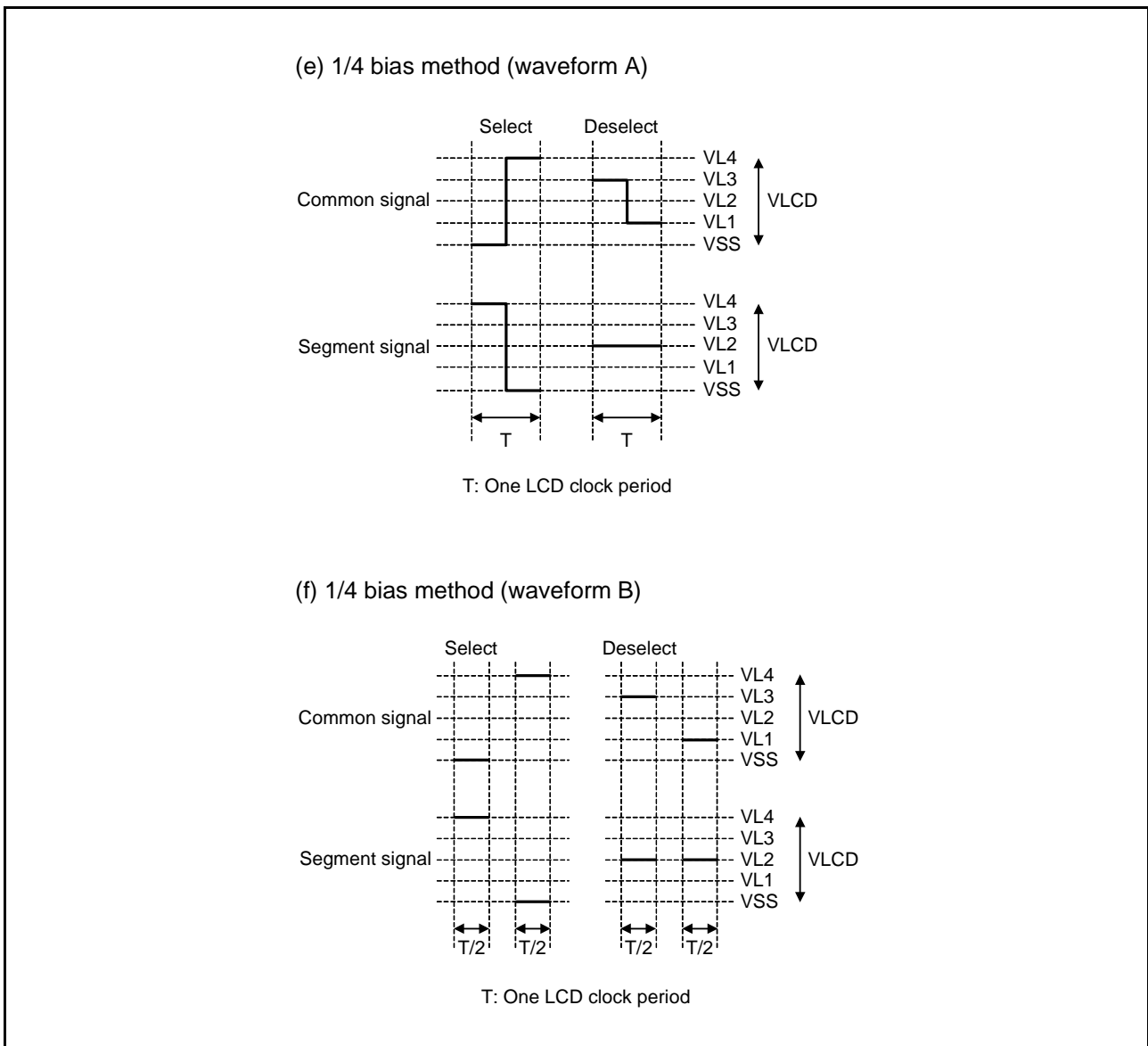


Figure 33.17 Voltages and Phases of Common and Segment Signals (3/3)



### 33.9 Display Modes

#### 33.9.1 Static Display Example

Figure 33.19 shows how the three-digit LCD panel having the display pattern shown in Figure 33.18 is connected to the segment signals (SEG00 to SEG23) and the common signal (COM0). This example displays data “12.3” in the LCD panel. The contents of the display data register correspond to this display.

The following description focuses on numeral “2.” (  $\frac{2}{.}$  ) displayed in the second digit. To display “2.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG08 to SEG15 pins at the select timing of the common signal COM0; see Figure 33.18 for the relationship between the segment signals and LCD segments.

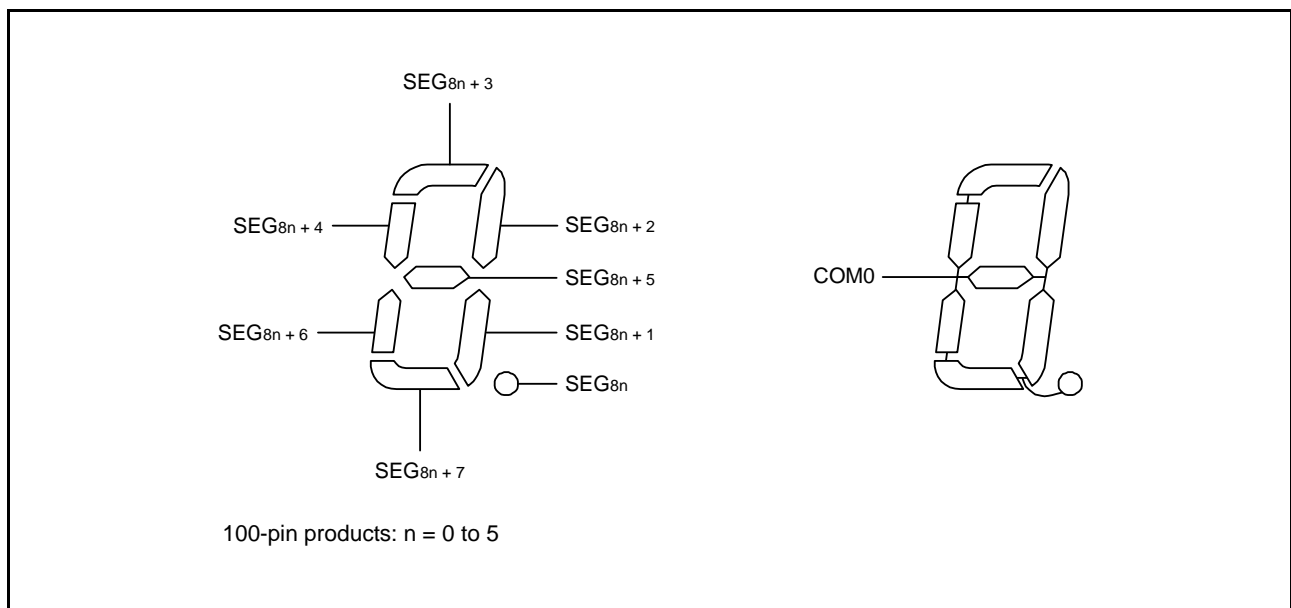
**Table 33.12 Example of Select (1) and Deselect (0) Data (COM0)**

Common \ Segment	SEG08	SEG09	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
COM0	1	0	1	1	0	1	1	1

According to Table 33.12, it is determined that the display data register location that corresponds to SEG15 must contain “xxx1b”.

Figure 33.20 shows the LCD drive waveforms of SEG11 and SEG12, and COM0. When the select voltage is applied to SEG11 at the timing of COM0, an alternate rectangle waveform, +VLCD/-VLCD, is generated to turn on the corresponding LCD segment.

COM1 to COM3 are supplied with the same waveform as for COM0. So, COM0 to COM3 may be connected together to increase the driving capacity.



**Figure 33.18 Static LCD Display Pattern and Electrode Connections**

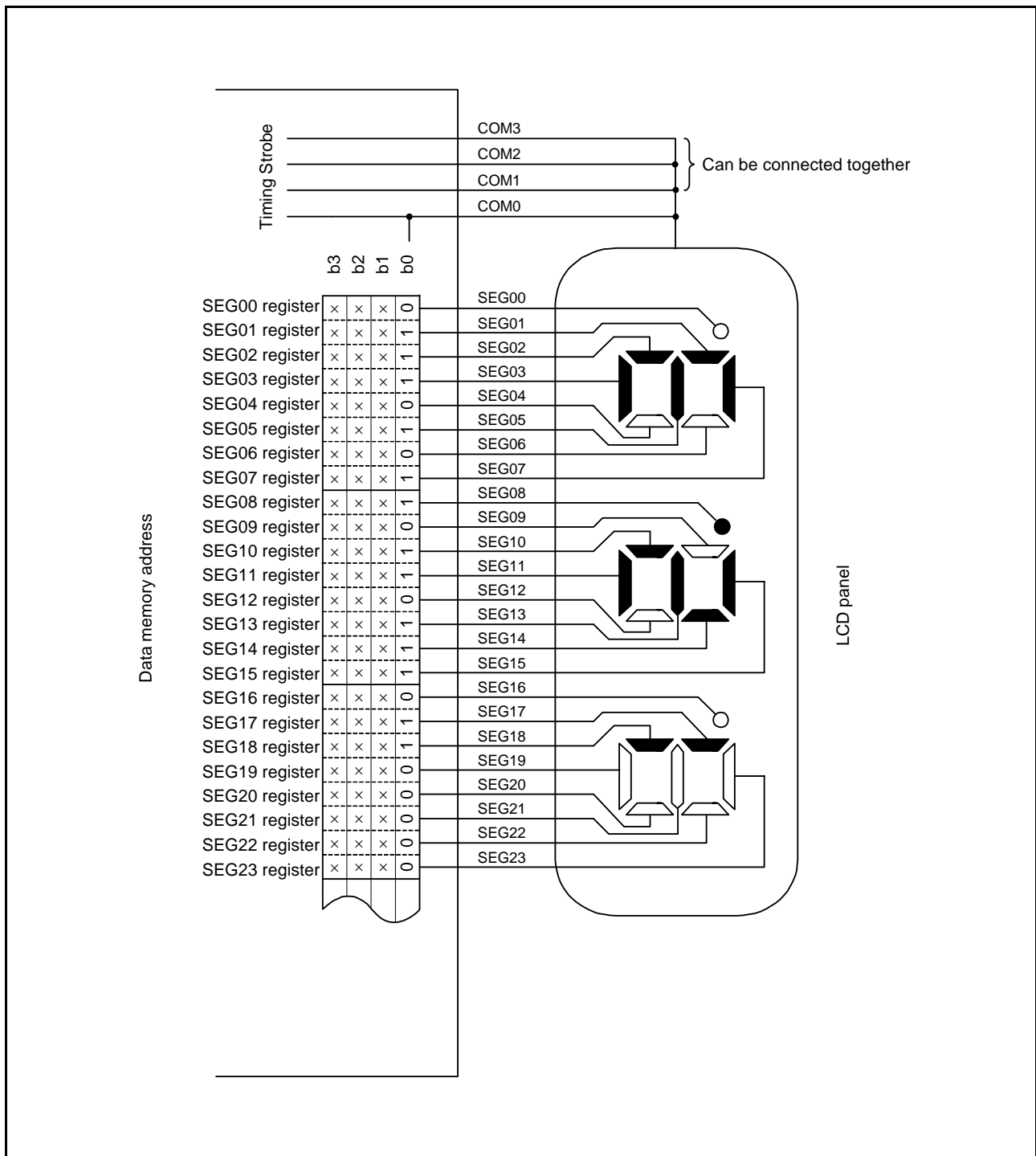


Figure 33.19 Example of Connecting Static LCD Panel

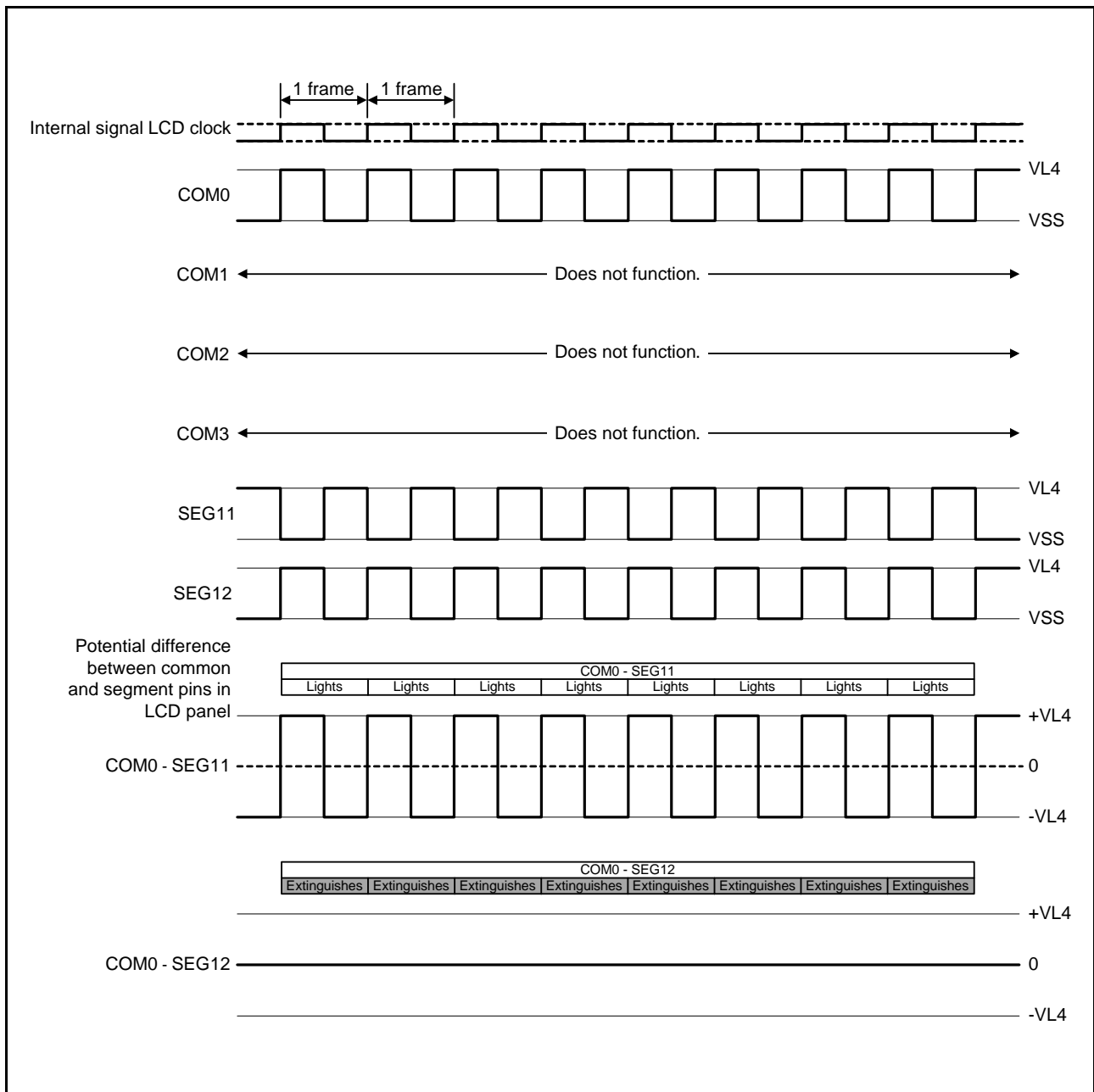


Figure 33.20 Static LCD Drive Waveform Examples for SEG11, SEG12, and COM0

### 33.9.2 Two-Time-Slice Display Example

Figure 33.22 shows how the 6-digit LCD panel having the display pattern shown in Figure 33.21 is connected to the segment signals (SEG00 to SEG23) and the common signals (COM0 and COM1). This example displays data “12345.6” in the LCD panel. The contents of the display data register correspond to this display.

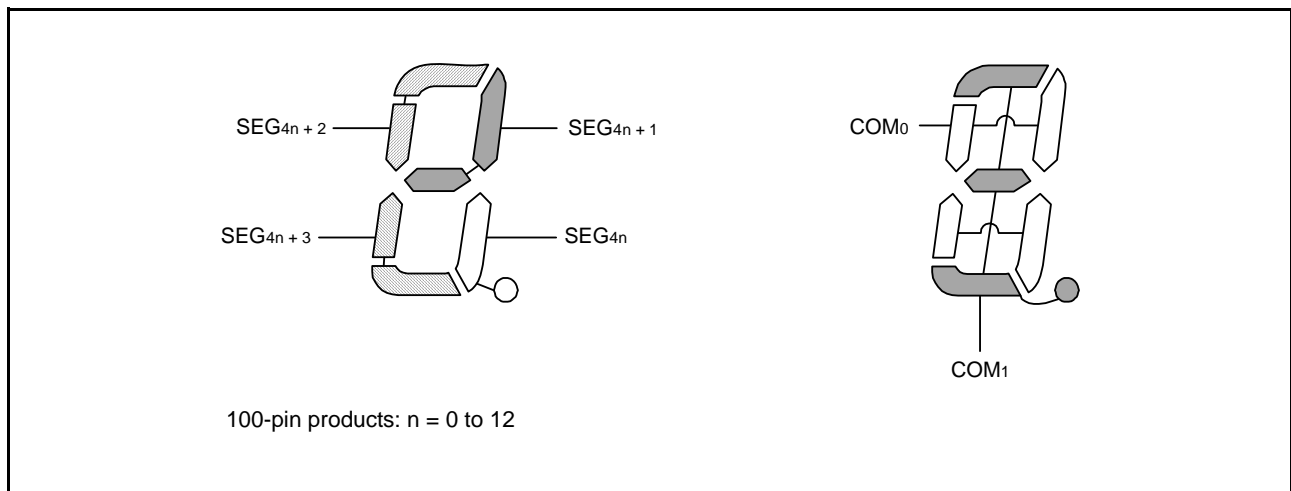
The following description focuses on numeral “3” ( 3 ) displayed in the fourth digit. To display “3” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG12 to SEG15 pins at the select timing of the common signals COM0 and COM1; see Figure 33.21 for the relationship between the segment signals and LCD segments.

**Table 33.13 Example of Select (1) and Deselect (0) Data (COM0 and COM1)**

Common \ Segment	SEG12	SEG13	SEG14	SEG15
COM0	1	1	0	0
COM1	0	1	1	1

According to Table 33.13, it is determined that the display data register location that corresponds to SEG15 must contain “xx10b”.

Figure 33.23 shows examples of LCD drive waveforms between the SEG15 signal and each common signal. When the select voltage is applied to SEG15 at the timing of COM1, an alternate rectangle waveform, +VLCD/−VLCD, is generated to turn on the corresponding LCD segment.



**Figure 33.21 Two-Time-Slice LCD Display Pattern and Electrode Connections**

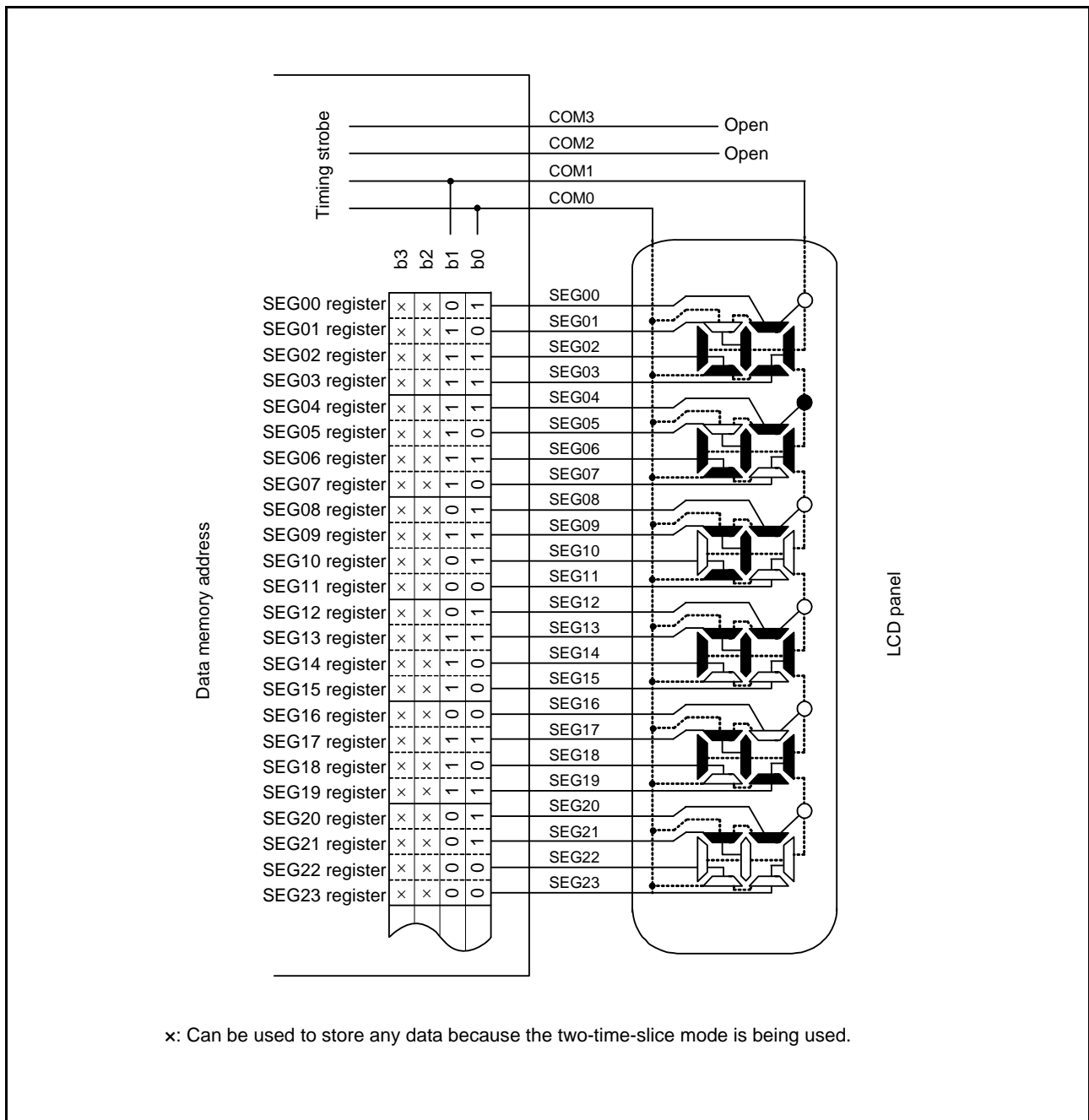


Figure 33.22 Example of Connecting Two-Time-Slice LCD Panel

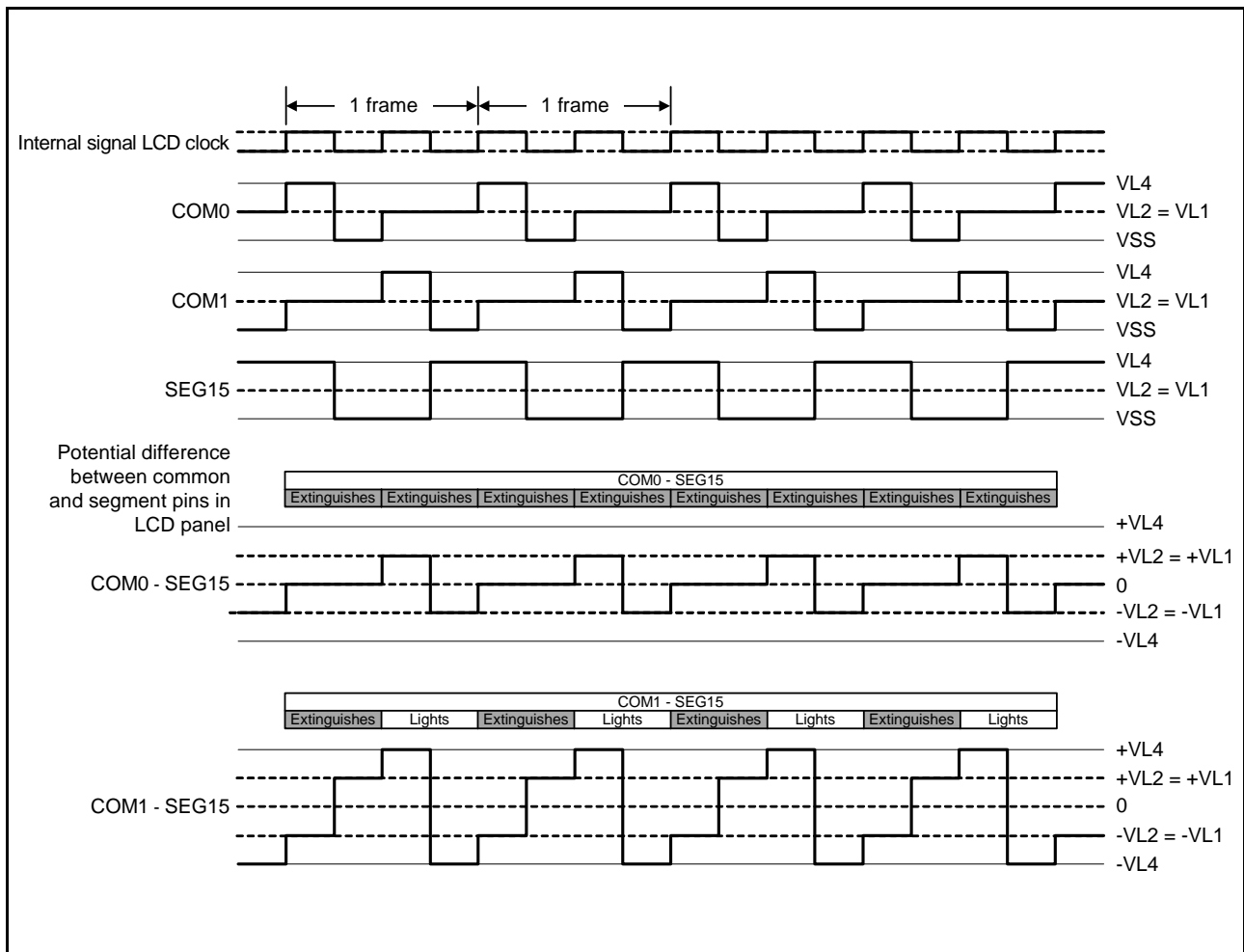


Figure 33.23 Two-Time-Slice LCD Drive Waveform Examples Between SEG15 and Each Common Signals (1/2 Bias Method)

### 33.9.3 Three-Time-Slice Display Example

Figure 33.25 shows how the 8-digit LCD panel having the display pattern shown in Figure 33.24 is connected to the segment signals (SEG00 to SEG23) and the common signals (COM0 to COM2). This example displays data “123456.78” in the LCD panel. The contents of the display data register correspond to this display.

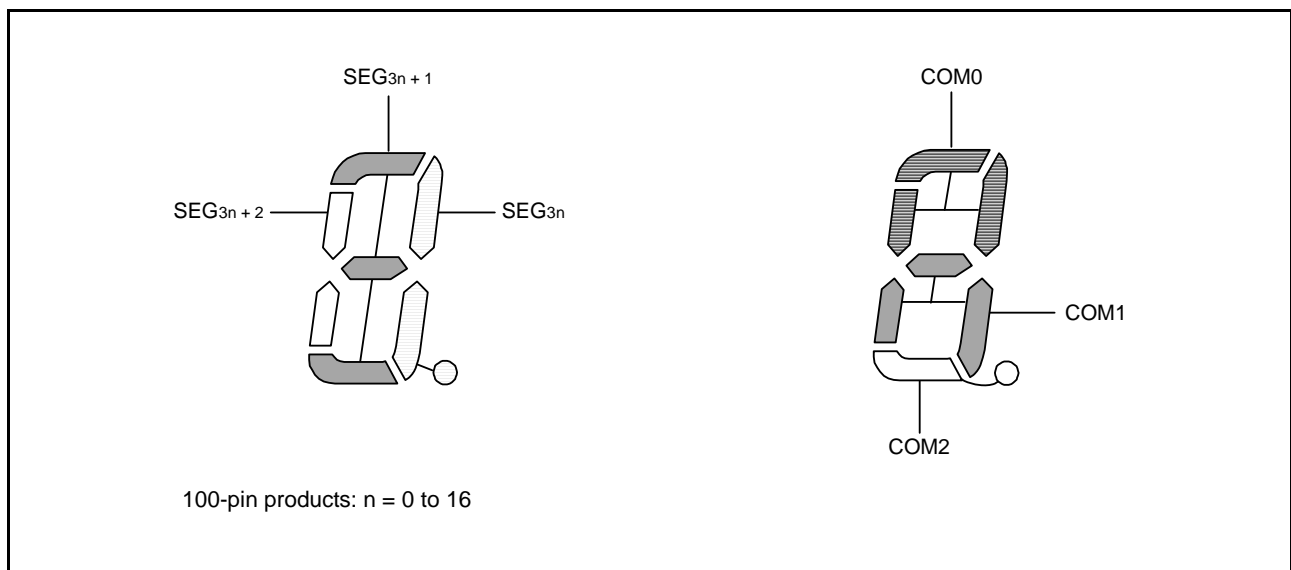
The following description focuses on numeral “6.” (6.) displayed in the third digit. To display “6.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG06 to SEG08 pins at the select timing of the common signals COM0 to COM2; see Figure 33.24 for the relationship between the segment signals and LCD segments.

**Table 33.14 Example of Select (1) and Deselect (0) Data (COM0 to COM2)**

Common	Segment	SEG06	SEG07	SEG08
	COM0		0	1
COM1		1	1	1
COM2		1	1	—

According to Table 33.14, it is determined that the display data register location that corresponds to SEG06 must contain “x110b”.

Figure 33.26 and Figure 33.27 show examples of LCD drive waveforms between the SEG06 signal and each common signal in the 1/2 and 1/3 bias methods, respectively. When the select voltage is applied to SEG06 at the timing of COM1 or COM2, an alternate rectangle waveform, +VLCD/–VLCD, is generated to turn on the corresponding LCD segment.



**Figure 33.24 Three-Time-Slice LCD Display Pattern and Electrode Connections**

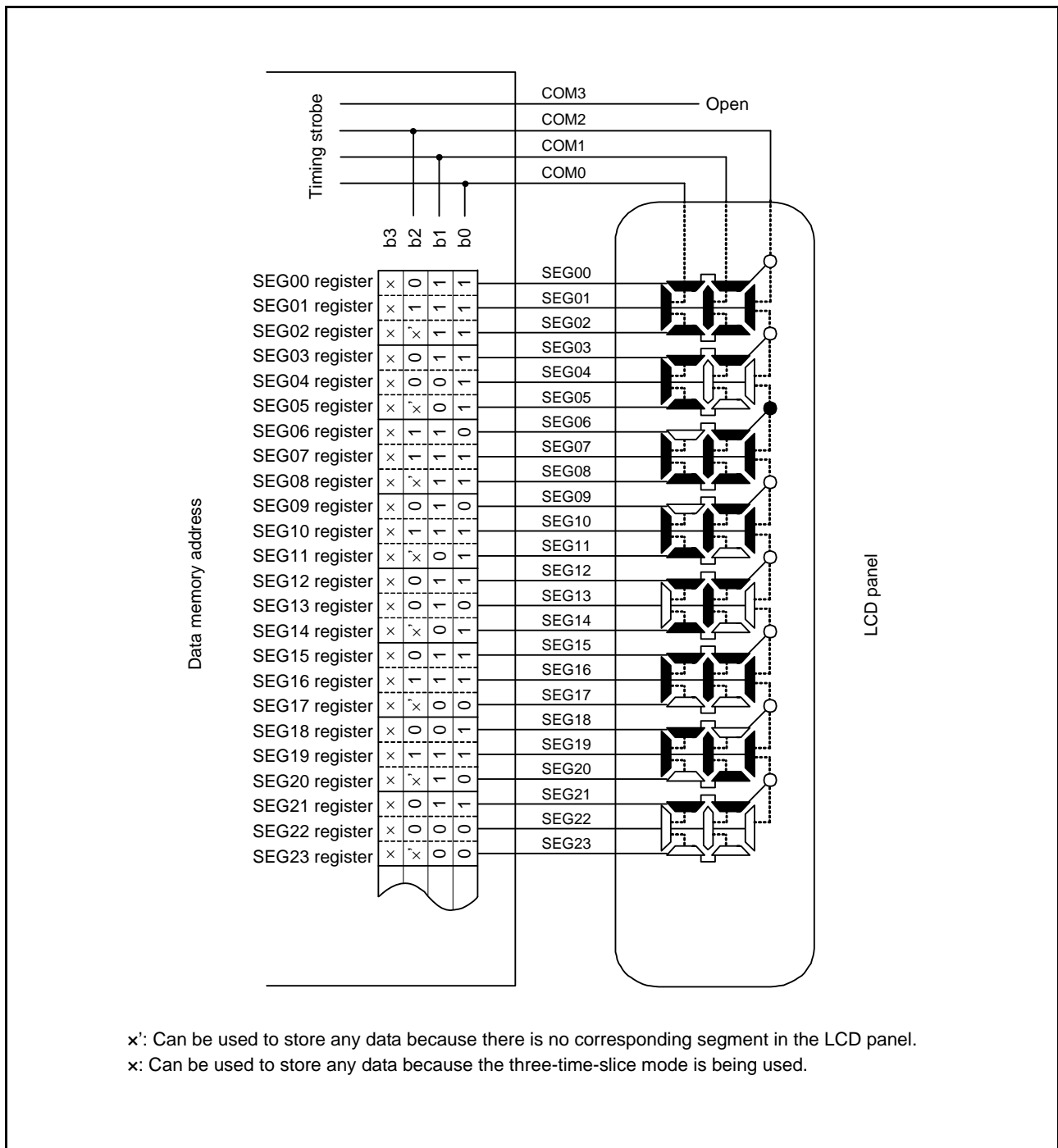


Figure 33.25 Example of Connecting Three-Time-Slice LCD Panel



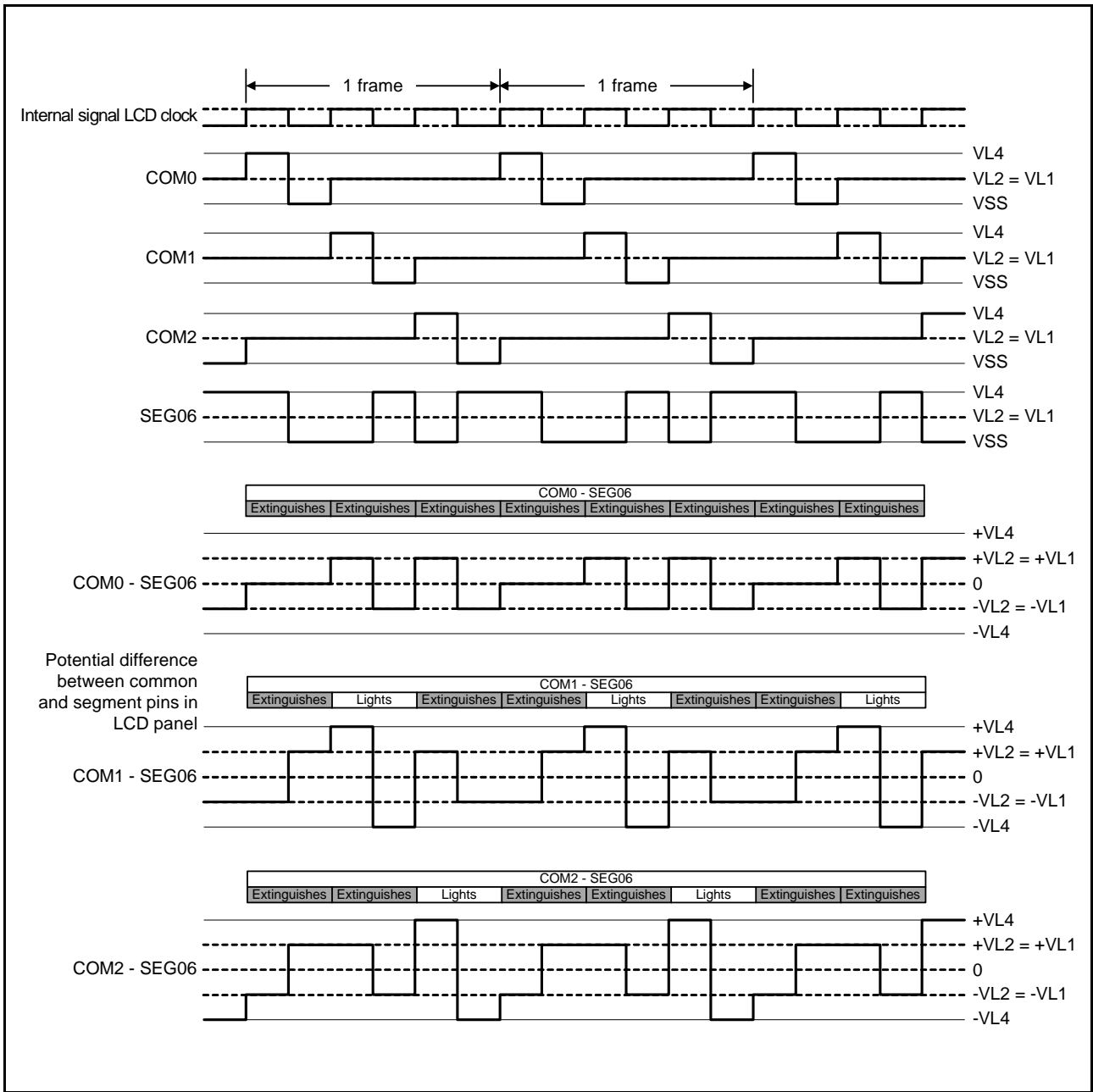


Figure 33.26 Three-Time-Slice LCD Drive Waveform Examples Between SEG06 and Each Common Signals (1/2 Bias Method)

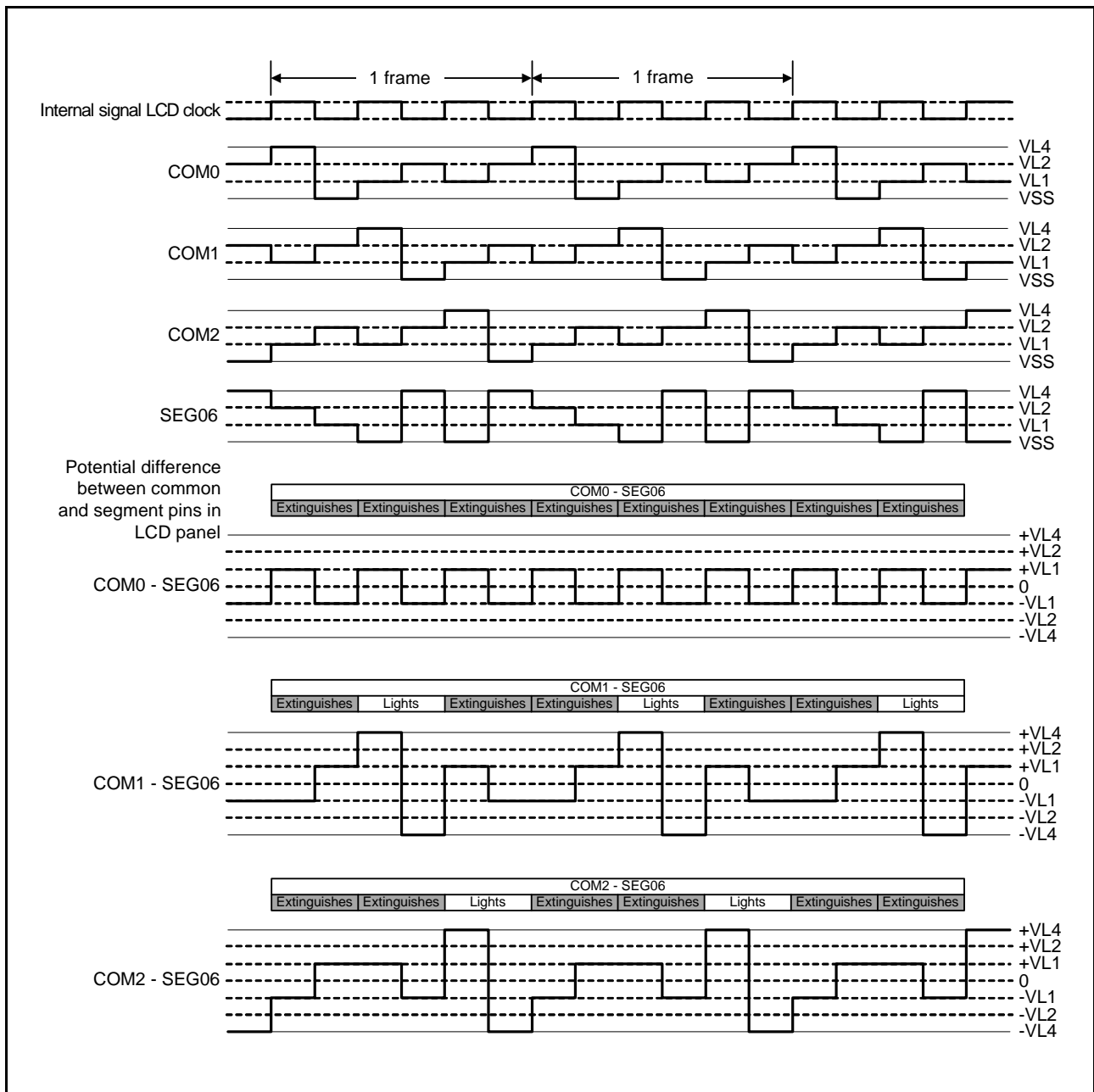


Figure 33.27 Three-Time-Slice LCD Drive Waveform Examples Between SEG06 and Each Common Signals (1/3 Bias Method)

### 33.9.4 Four-Time-Slice Display Example

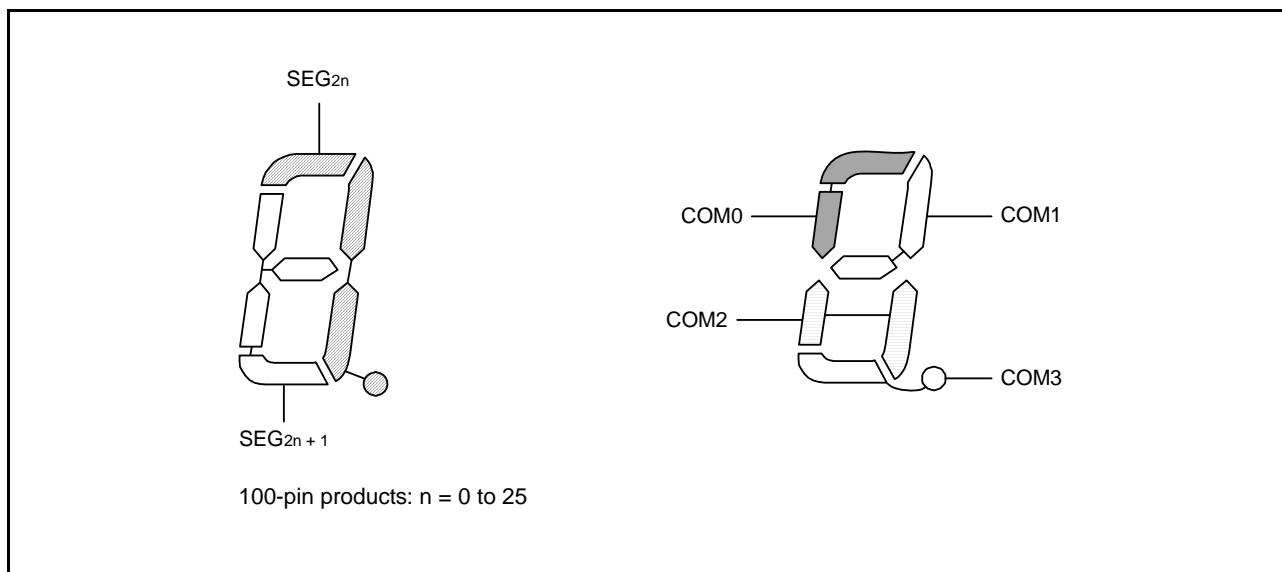
Figure 33.29 shows how the 12-digit LCD panel having the display pattern shown in Figure 33.28 is connected to the segment signals (SEG00 to SEG23) and the common signals (COM0 to COM3). This example displays data “123456.789012” in the LCD panel. The contents of the display data register correspond to this display. The following description focuses on numeral “6.” ( 6. ) displayed in the seventh digit. To display “6.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG12 and SEG13 pins at the select timing of the common signals COM0 to COM3; see Figure 33.28 for the relationship between the segment signals and LCD segments.

**Table 33.15 Example of Select (1) and Deselect (0) Data (COM0 to COM3)**

Common	Segment	SEG12	SEG13
COM0		1	1
COM1		0	1
COM2		1	1
COM3		1	1

According to Table 33.15, it is determined that the display data register location that corresponds to SEG12 must contain “1101b”.

Figure 33.30 shows examples of LCD drive waveforms between the SEG12 signal and each common signal. When the select voltage is applied to SEG12 at the timing of COM0, an alternate rectangle waveform, +VLCD/-VLCD, is generated to turn on the corresponding LCD segment.



**Figure 33.28 Four-Time-Slice LCD Display Pattern and Electrode Connections**

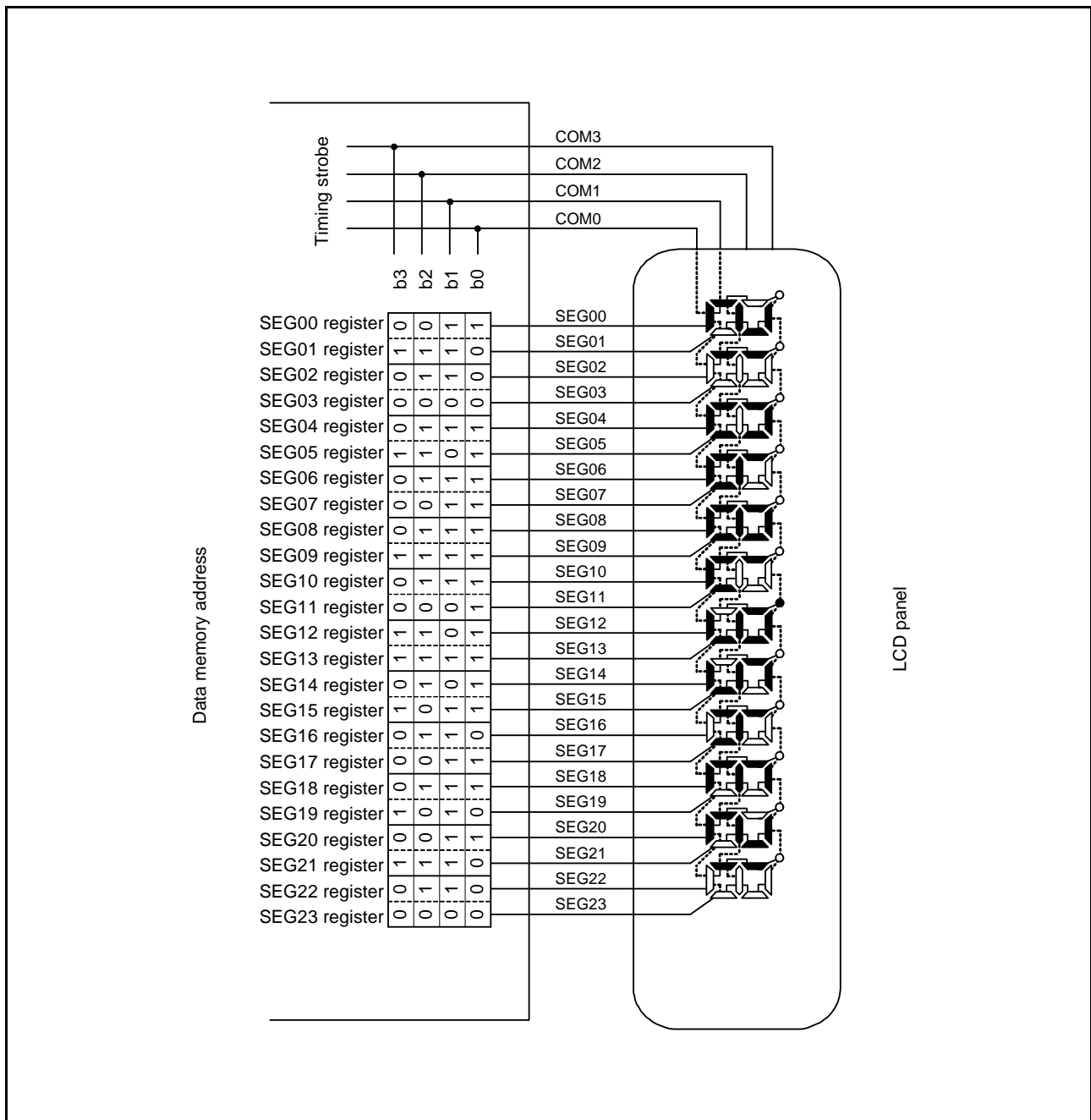


Figure 33.29 Example of Connecting Four-Time-Slice LCD Panel

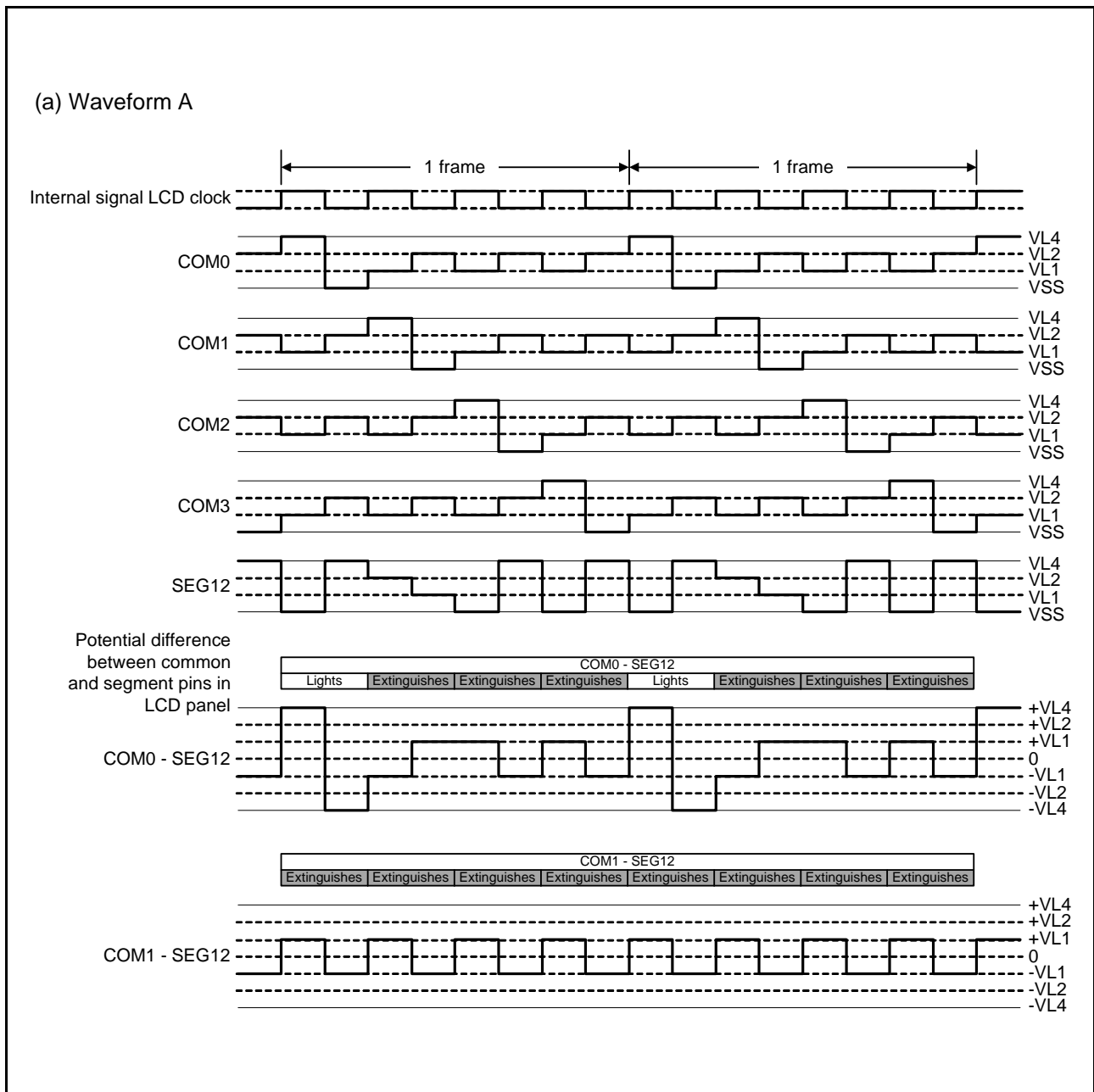


Figure 33.30 Four-Time-Slice LCD Drive Waveform Examples Between SEG12 and Each Common Signals (1/3 Bias Method)

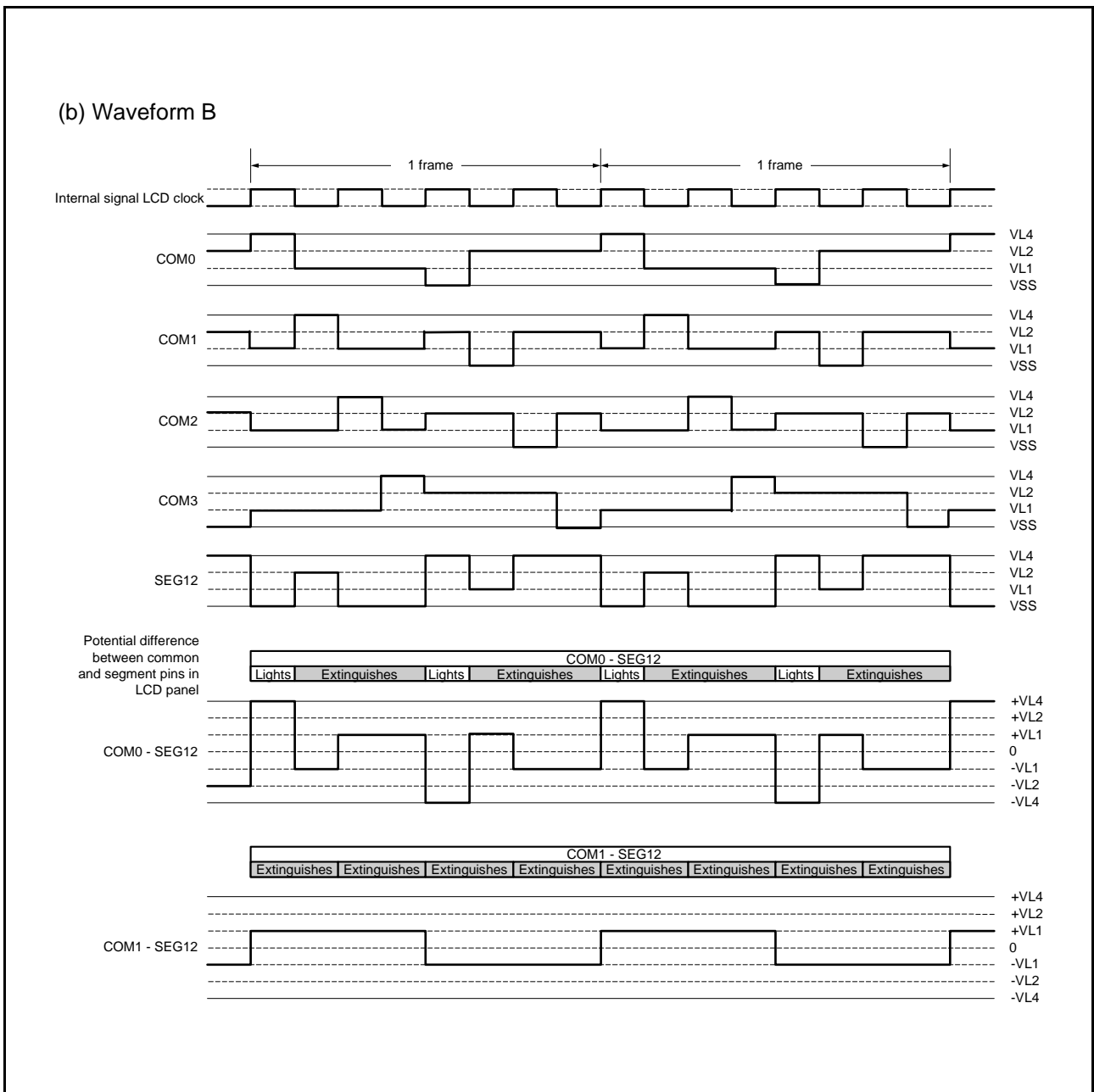


Figure 33.31 Four-Time-Slice LCD Drive Waveform Examples Between SEG12 and Each Common Signals (1/3 Bias Method)

### 33.9.5 Eight-Time-Slice Display Example

Figure 33.33 shows how the 15 × 8 dot LCD panel having the display pattern shown in Figure 33.32 is connected to the segment signals (SEG04 to SEG18) and the common signals (COM0 to COM7). This example displays data “123” in the LCD panel. The contents of the display data register correspond to this display.

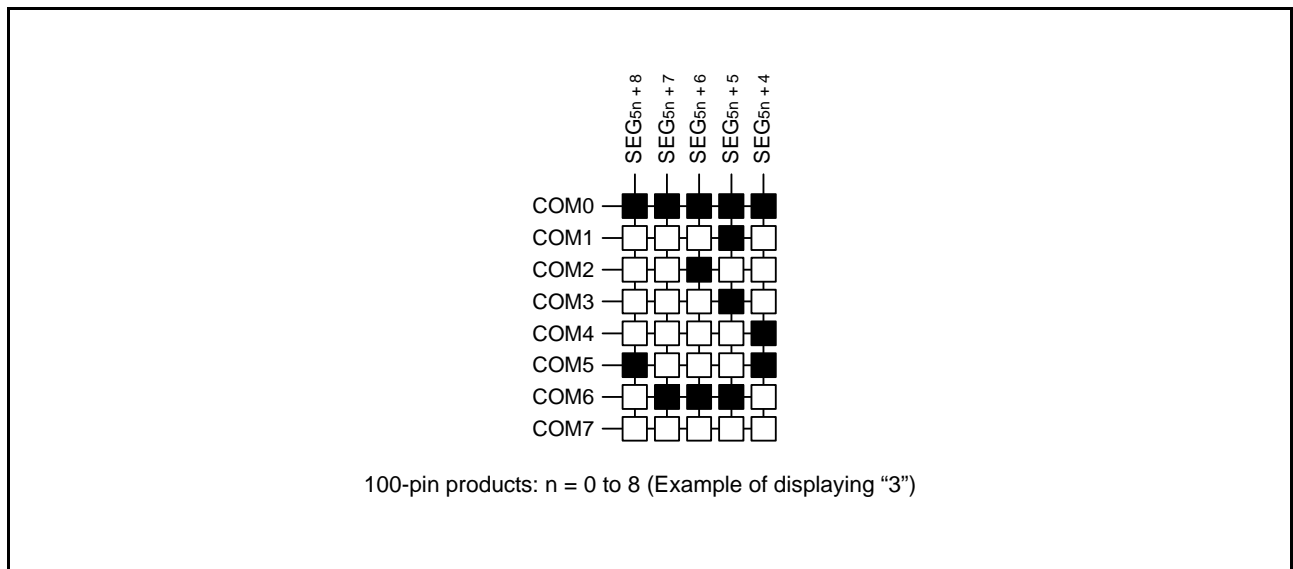
The following description focuses on numeral “3” (三) displayed in the first digit. To display “3” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG04 to SEG08 pins at the select timing of the common signals COM0 to COM7; see Figure 33.32 for the relationship between the segment signals and LCD segments.

**Table 33.16 Example of Select (1) and Deselect (0) Data (COM0 to COM7)**

Common \ Segment	SEG04	SEG05	SEG06	SEG07	SEG08
COM0	1	1	1	1	1
COM1	0	1	0	0	0
COM2	0	0	1	0	0
COM3	0	1	0	0	0
COM4	1	0	0	0	0
COM5	1	0	0	0	1
COM6	0	1	1	1	0
COM7	0	0	0	0	0

According to Table 33.16, it is determined that the display data register location that corresponds to SEG04 must contain “00110001b”.

Figure 33.34 and Figure 33.35 show examples of LCD drive waveforms between the SEG04 signal and each common signal. When the select voltage is applied to SEG04 at the timing of COM0, a waveform is generated to turn on the corresponding LCD segment.



**Figure 33.32 Eight-Time-Slice LCD Display Pattern and Electrode Connections**

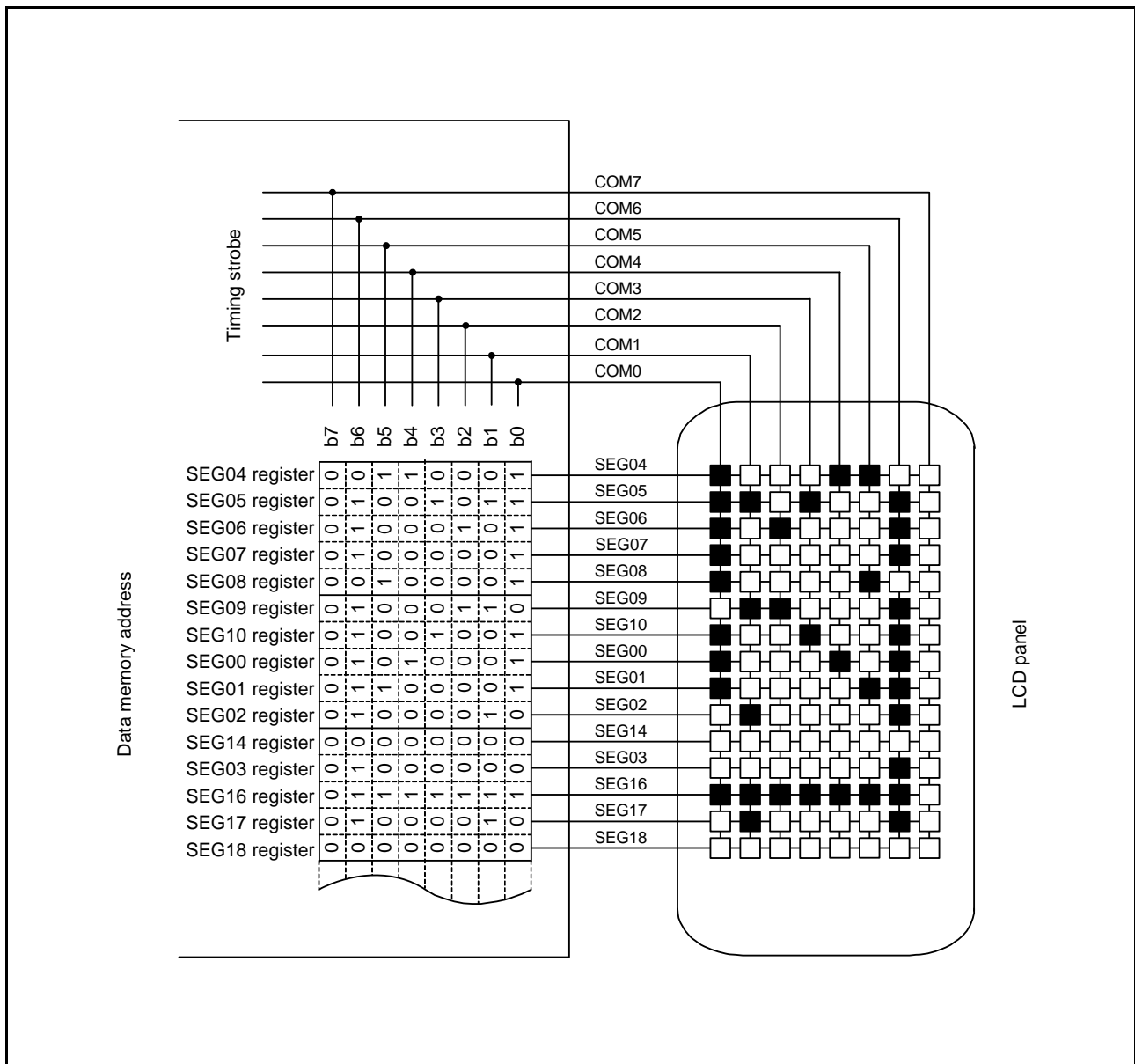


Figure 33.33 Example of Connecting Eight-Time-Slice LCD Panel



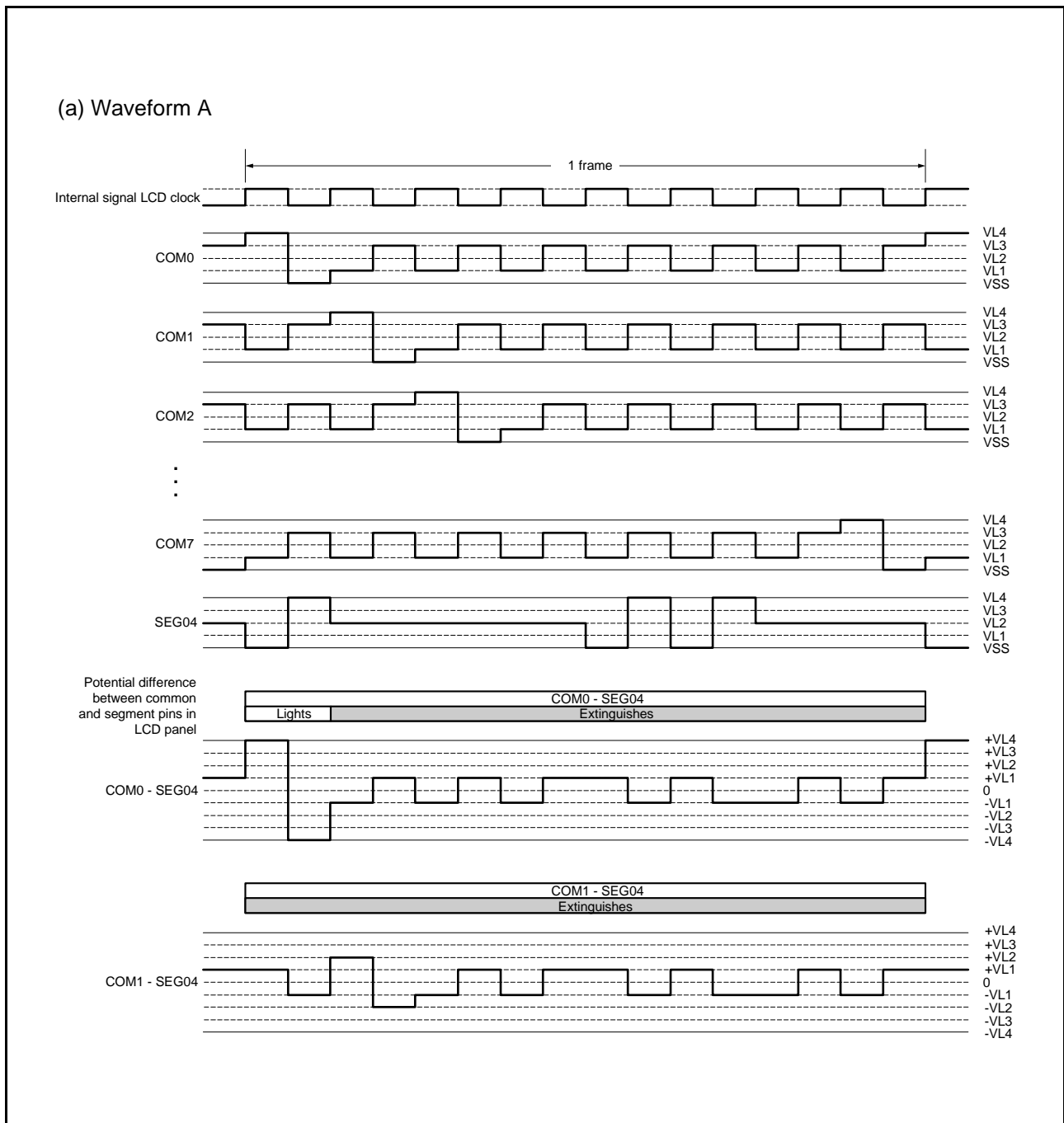


Figure 33.34 Eight-Time-Slice LCD Drive Waveform Examples Between SEG04 and Each Common Signals (1/4 Bias Method) (1/2)

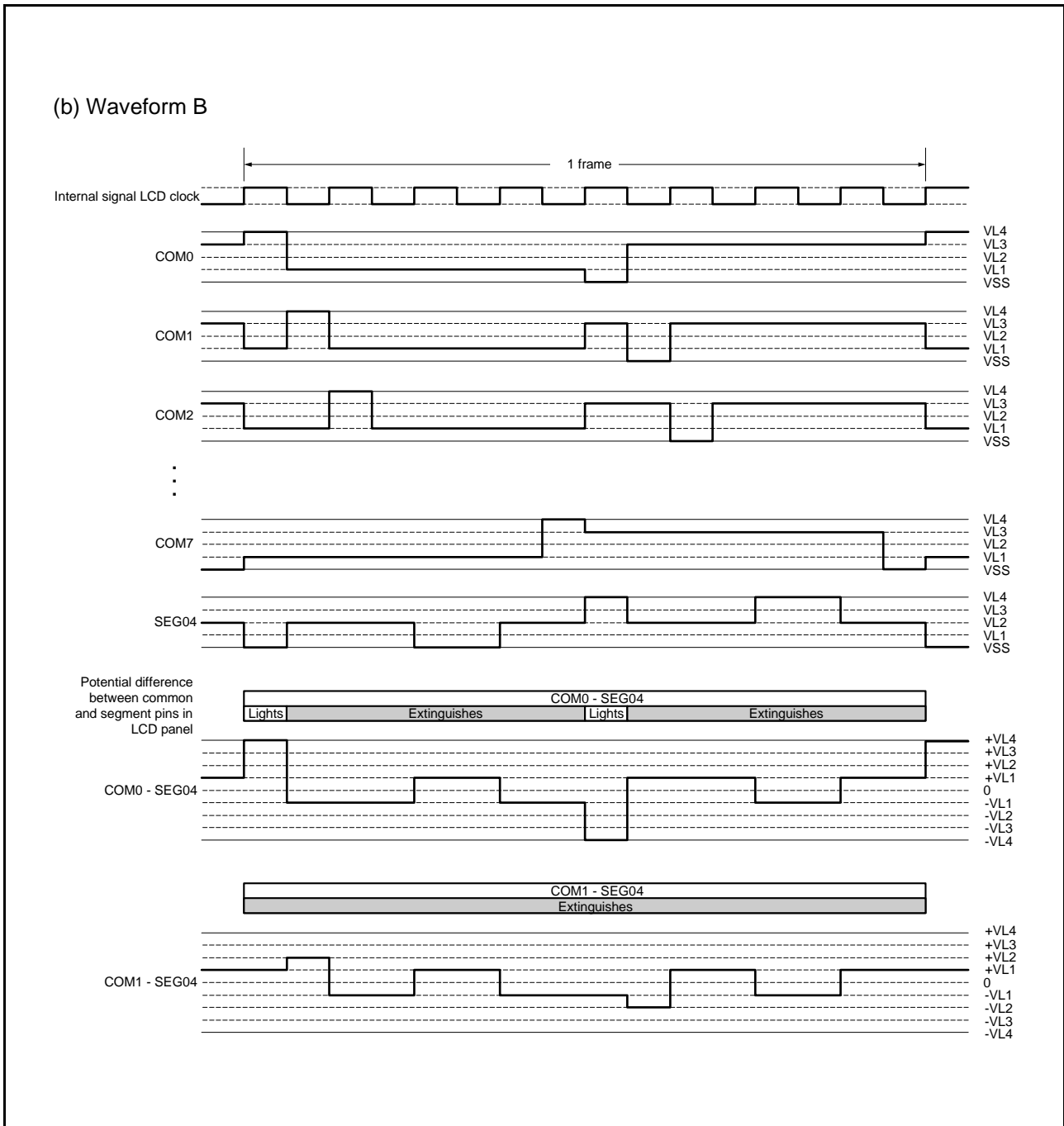


Figure 33.35 Eight-Time-Slice LCD Drive Waveform Examples Between SEG04 and Each Common Signals (1/4 Bias Method) (1/2)

## 33.10 Usage Notes

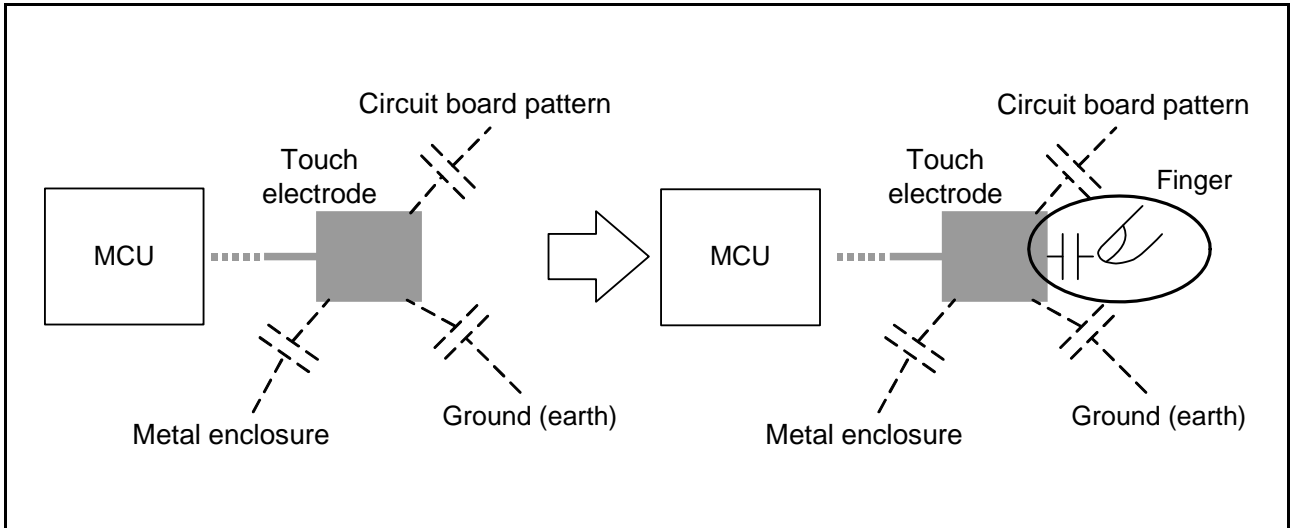
### 33.10.1 Module Stop Function Setting

LCD operation can be disabled or enabled using module stop control register D (MSTPCRD) and the LCD source clock control register (LCDSCLKCR2). The value after reset is for the LCD to be halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

### 34. Capacitive Touch Sensing Unit (CTSU)

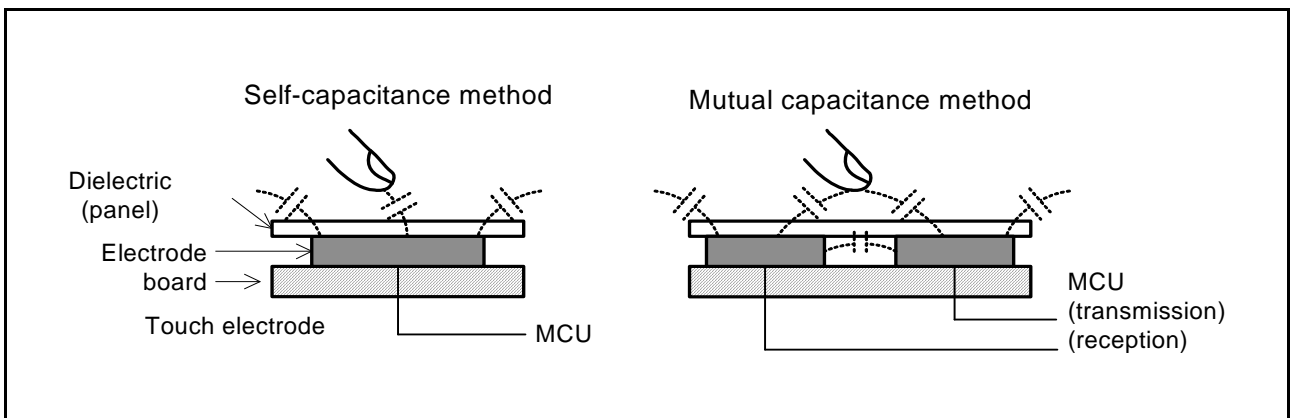
The capacitive touch sensing unit (CTSU) measures the electrostatic capacitance of the touch sensor. Changes in the electrostatic capacitance are determined by software, which enables the CTSU to detect whether a finger is in contact with the touch sensor. The electrode surface of the touch sensor is usually enclosed with a dielectric so that a finger does not come in contact with the electrode.

As shown in Figure 34.1, electrostatic capacitance (parasitic capacitance) exists between the electrode and the surrounding conductors. Because the human body is an electrical conductor, when a finger is placed close to the electrode, the value of electrostatic capacitance increases.



**Figure 34.1** Increased Electrostatic Capacitance Due to Presence of Finger

Electrostatic capacitance is detected by the following methods: Self-capacitance and mutual capacitance. In the self-capacitance method, the CTSU detects electrostatic capacitance generated between a finger and a single electrode. In the mutual capacitance method, two electrodes are used as a transmit electrode and a receive electrode, and the CTSU detects the change in the electrostatic capacitance generated between the two when a finger is placed close to them.



**Figure 34.2** Self-Capacitance Method and Mutual Capacitance Method

Electrostatic capacitance is measured by counting a clock signal whose frequency changes according to the amount of charged/discharged current, for a specified period.

For details on the measurement principles of the CTSU, refer to section 34.3.1, Principles of Measurement Operation.

In this section, “PCLK” is used to refer to PCLKB.

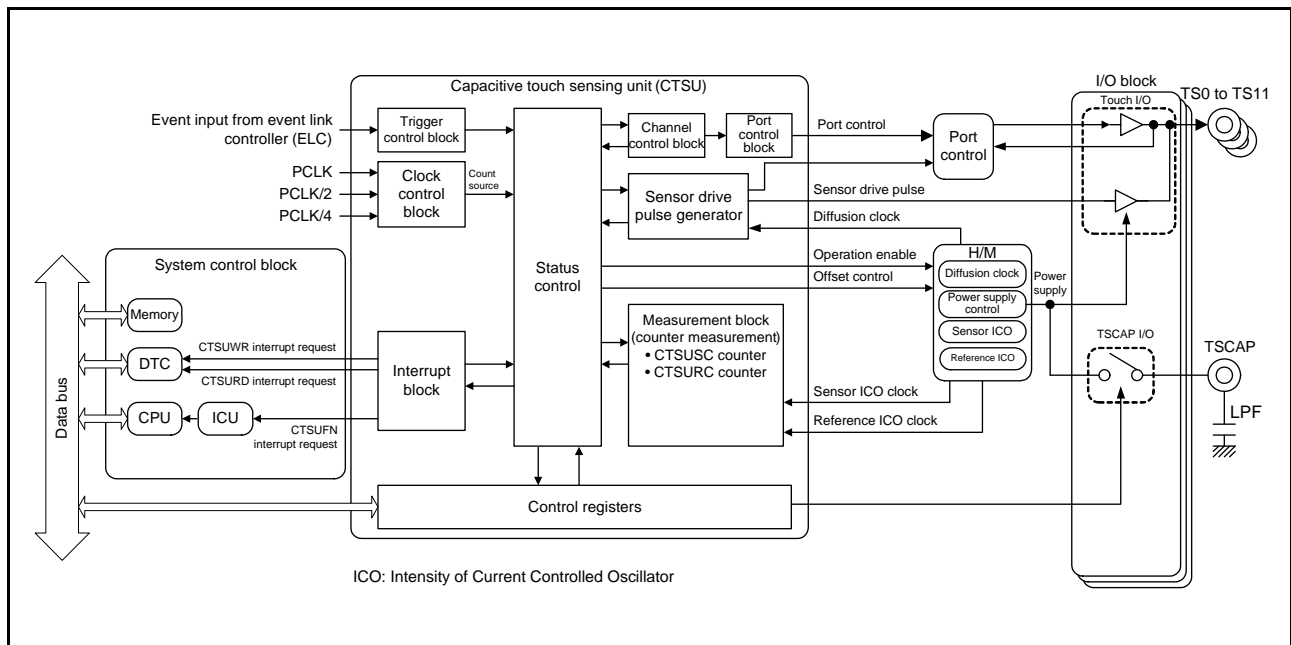
### 34.1 Overview

Table 34.1 lists the specifications of the CTSU, and Figure 34.3 shows a block diagram of the CTSU.

**Table 34.1 CTSU Specifications**

Item	Description	
Operating clock	PCLK, PCLK/2, or PCLK/4	
Pins	TS0 to TS11	Electrostatic capacitance measurement pins (12 channels)
	TSCAP	LPF (low-pass filter) connection pin
Measurement modes	Self-capacitance single scan mode	Electrostatic capacitance on a channel is measured by the self-capacitance method.
	Self-capacitance multi-scan mode	Electrostatic capacitance on multiple channels is measured successively by the self-capacitance method.
	Mutual capacitance simple scan mode	<ul style="list-style-type: none"> <li>Simple method for measurement by switching the impedance of the transmit electrode</li> <li>Electrostatic capacitance on multiple channels is measured successively.</li> </ul>
	Mutual capacitance full scan mode	Electrostatic capacitance on multiple channels is measured successively by mutual capacitance.
Noise prevention	Synchronous noise prevention, high-pass noise prevention	
Measurement start conditions	<ul style="list-style-type: none"> <li>Software trigger</li> <li>External trigger (event input from the event link controller (ELC))</li> </ul>	

As shown in Figure 34.3, the CTSU consists of the status control block, trigger control block, clock control block, channel control block, port control block, sensor drive pulse generator, measurement block, interrupt block, and control registers.



**Figure 34.3 CTSU Block Diagram**

**Table 34.2 CTSU Pin Configuration**

Pin Name	I/O	Function
TS0 to TS11	Output	Electrostatic capacitive measurement pins (touch pins)
TSCAP	Output	LPF connection pin

## 34.2 Register Descriptions

### 34.2.1 CTSU Control Register 0 (CTSUCR0)

Address(es): 000A 0900h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CTSUI NIT	CTSUI OC	CTSUS NZ	CTSUC AP	CTSUS TRT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUSTRT	CTSUS Measurement Operation Start	This bit specifies whether CTSU operation starts or stops. 0: Measurement operation stops 1: Measurement operation starts	R/W
b1	CTSUCAP	CTSUS Measurement Operation Start Trigger Select	This bit specifies the measurement start condition. 0: Software trigger 1: External trigger	R/W
b2	CTSUSNZ	CTSUS Wait State Power-Saving Enable	This bit sets the power-saving function during a wait state. 0: Power-saving function during wait state is disabled 1: Power-saving function during wait state is enabled	R/W
b3	CTSUIOC	CTSUS Transmit Pin Control	0: Low-level output from transmit channel non-measurement pin 1: High-level output from transmit channel non-measurement pin	R/W
b4	CTSUIINIT	CTSUS Control Block Initialization	Writing 1 to this bit initializes the CTSU control block and registers.*1 This bit is read as 0.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The CTSUSC, CTSURC, CTSUMCH0, CTSUMCH1, and CTSUST registers are initialized.

The CTSUCAP, CTSUSNZ, and CTSUIOC bits should be set when the CTSUSTRT bit = 0. These bits can be set at the same time as starting measurement operation.

#### CTSUSTRT Bit (CTSUS Measurement Operation Start)

This bit specifies whether CTSUS operation starts or stops.

When the CTSUCAP bit = 0, measurement is started by writing 1 to the CTSUSTRT bit (software trigger), and the CTSUSTRT bit is cleared to 0 by hardware when measurement is stopped.

When the CTSUCAP bit = 1, the CTSUS waits for an external trigger by writing 1 to the CTSUSTRT bit, and measurement is started at the rising edge of the external trigger. When measurement is stopped, the CTSUS waits for the next external trigger and operation is continued.

Table 34.3 lists the CTSUS states.

**Table 34.3 CTSUS States**

CTSUSTRT Bit	CTSUCAP Bit	CTSUS State
0	0	Stopped
0	1	Stopped
1	0	During measurement
1	1	During measurement/wait for an external trigger*1

Note 1. The state can be read from the CTSUST.CTSUSTC[2:0] counters.  
 During measurement: CTSUST.CTSUSTC[2:0] counters ≠ 000b  
 Wait for an external trigger: CTSUST.CTSUSTC[2:0] counters = 000b

If the CTSUSTRT bit is set to 1 by software when the CTSUSTRT bit = 1, writing is ignored and operation is continued. To forcibly stop operation set by software (forced stop) when the CTSUSTRT bit = 1, set the CTSUSTRT bit and 0 to the CTSUINIT bit to 1 at the same time.

#### CTSUCAP Bit (CTSU Measurement Operation Start Trigger Select)

This bit specifies the measurement start condition. For details, see the description of the CTSUSTRT bit.

#### CTSUSNZ Bit (CTSU Wait State Power-Saving Enable)

This bit enables or disables power-saving operation during a wait state. This bit can also be used to suspend the CTSU power supply, which decreases power consumption during the wait state.

In the suspended state, the CTSU power supply is turned off while the external TSCAP is still charged after the CTSU power supply has been turned on and the TSCAP has been charged.

**Table 34.4 CTSU Power Supply State Control**

CTSUCR1.CTSUPON Bit	CTSUSNZ Bit	CTSUCAP Bit	CTSUSTRT Bit	CTSU Power Supply State
0	0	0	0	Stopped
1	0	—	—	Operating
1	1	0	0	Suspended

Note: Settings other than the above are prohibited.

To start measurement from the suspended state, set the CTSUSNZ bit to 0 and then set the CTSUSTRT bit to 1. To set the suspended state after measurement is stopped, set the CTSUSNZ bit to 1.

#### CTSUIOC Bit (CTSU Transmit Pin Control)

This bit specifies the output state (high-level output/low-level output) of the non-measurement pin in mutual capacitance simple scan mode (CTSUCR1.CTSUMD[1:0] bits = 10b). This bit is enabled only in mutual capacitance simple scan mode.

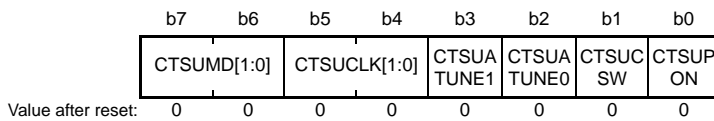
#### CTSUINIT Bit (CTSU Control Block Initialization)

The internal control registers can be initialized by writing 1 to this bit. To forcibly stop the current operation, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 at the same time. In this case, the operation is stopped and the internal control registers are initialized.

Do not write 1 to the CTSUINIT bit at the same time as the CTSUSTRT bit = 1 (CTSU operation starts).

### 34.2.2 CTSU Control Register 1 (CTSUCR1)

Address(es): 000A 0901h



Bit	Symbol	Bit Name	Description	R/W
b0	CTSUPON	CTSUS Power Supply Enable	This bit controls the CTSUS power supply. 0: Powered off 1: Powered on	R/W
b1	CTSUCSW	CTSUS LPF Capacitance Charging Control	This bit controls charging of the LPF capacitor connected to the TSCAP pin. 0: Capacitance switch turned off 1: Capacitance switch turned on	R/W
b2	CTSUA TUNE0	CTSUS Power Supply Operating Mode Setting	VCC ≥ 2.4 V 0: Normal operating mode 1: Low-voltage operating mode VCC < 2.4 V 0: Setting prohibited 1: Low-voltage operating mode	R/W
b3	CTSUA TUNE1	CTSUS Power Supply Capacity Adjustment	0: Normal output 1: High-current output	R/W
b5, b4	CTSUCLK[1:0]	CTSUS Operating Clock Select	These bits select the operating clock. b5 b4 0 0: PCLK 0 1: PCLK/2 (PCLK divided by 2) 1 0: PCLK/4 (PCLK divided by 4) 1 1: Setting prohibited	R/W
b7, b6	CTSUMD[1:0]	CTSUS Measurement Mode Select	These bits select the measurement mode. b7 b6 0 0: Self-capacitance single scan mode 0 1: Self-capacitance multi-scan mode 1 0: Mutual capacitance simple scan mode 1 1: Mutual capacitance full scan mode	R/W

The CTSUCR1 register should be set when the CTSUCR0.CTSUSTRT bit = 0.

#### CTSUPON Bit (CTSUS Power Supply Enable)

This bit controls power supply to the CTSUS. Set the CTSUPON and CTSUCSW bits to the same value at the same time.

#### CTSUCSW Bit (CTSUS LPF Capacitance Charging Control)

This bit controls charging of the LPF capacitor connected to the TSCAP pin (turning on/off of the capacitance switch). After the capacitance switch is turned on, wait until the capacitance connected to the TSCAP pin is charged for the specified time before starting measurement (CTSUCR0.CTSUSTRT = 1). Prior to measurement, use an I/O port to output a low level to the TSCAP pin, and discharge the LPF capacitance that has been already charged. Set the CTSUPON and CTSUCSW bits to the same value at the same time.

#### CTSUA TUNE0 Bit (CTSUS Power Supply Operating Mode Setting)

This bit sets the power supply operating mode. Set this bit according to the lower limit of the VCC for operating the CTSUS. As an example, when performing touch measurement in a system (the VCC voltage range is 2 to 3 V) where the VCC varies depending on battery operation, set this bit to 1 regardless of the initial VCC voltage.



**CTSUSOFF Bit (CTSUS High-Pass Noise Reduction Function Off Setting)**

This bit turns on or off the function for reducing high-pass noise. Set this bit to 1 when turning off the high-pass noise reduction function.

**CTSUSOFF Bit (CTSUS High-Pass Noise Reduction Function Off Setting)**

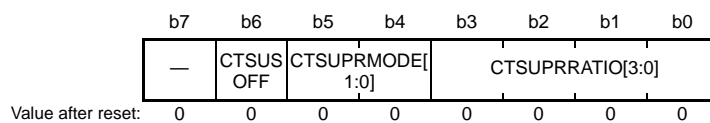
This bit turns on/off the spectrum diffusion function for reducing high-pass noise.

**CTSUSOFF Bit (CTSUS High-Pass Noise Reduction Function Off Setting)**

This bit turns on/off the spectrum diffusion function for reducing high-pass noise.

**34.2.3 CTSUS Synchronous Noise Reduction Setting Register (CTSUSDPRES)**

Address(es): 000A 0902h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CTSUPRRATIO[3:0]	CTSUS Measurement Time and Pulse Count Adjustment	These bits set the measurement time and the number of measurement pulses. Recommended setting value: 3 (0011b)	R/W
b5, b4	CTSUPRMODE[1:0]	CTSUS Base Period and Pulse Count Setting	These bits set the number of base pulses. b5 b4 0 0: 510 pulses 0 1: 126 pulses 1 0: 62 pulses (recommended setting value) 1 1: Setting prohibited	R/W
b6	CTSUSOFF	CTSUS High-Pass Noise Reduction Function Off Setting	This bit turns on/off the spectrum diffusion function for reducing high-pass noise. 0: High-pass noise reduction function turned on 1: High-pass noise reduction function turned off	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The CTSUSDPRES register should be set when the CTSUCR0.CTUSSTRT bit = 0.

**CTSUPRRATIO[3:0] Bits (CTSUS Measurement Time and Pulse Count Adjustment)**

These bits are used to determine the measurement time and the number of measurement pulses. These are calculated by the following formula. The number of base pulses is determined by setting the CTSUPRMODE[1:0] bits.

Number of measurement pulses = number of base pulses × (CTSUPRRATIO[3:0] bits + 1)

Measurement time = (number of base pulses × (CTSUPRRATIO[3:0] bits + 1) + (number of base pulses - 2) × 0.25) × base clock cycle

Note: For details on the base clock cycle, refer to section 34.2.15, CTSUS Sensor Offset Register 1 (CTSUSO1).

**CTSUPRMODE[1:0] Bits (CTSUS Base Period and Pulse Count Setting)**

These bits select the number of base pulses during measurement.

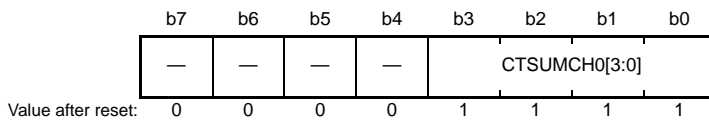
**CTSUSOFF Bit (CTSUS High-Pass Noise Reduction Function Off Setting)**

This bit turns on or off the function for reducing high-pass noise. Set this bit to 1 when turning off the high-pass noise reduction function.



### 34.2.5 CTSU Measurement Channel Register 0 (CTSUMCH0)

Address(es): 000A 0904h



Bit	Symbol	Bit Name	Description	R/W																																																																																																																																							
b3 to b0	CTSUMCH0[3:0]	CTSU Measurement Channel 0	<p>In self-capacitance single scan, these bits set a channel to be measured.</p> <table style="margin-left: 20px; border: none;"> <tr> <td style="padding-right: 10px;">b3</td> <td style="padding-right: 10px;">b2</td> <td style="padding-right: 10px;">b1</td> <td style="padding-right: 10px;">b0</td> <td></td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>TS0</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>TS1</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>TS2</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>TS3</td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>0</td><td>TS4</td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>1</td><td>TS5</td> </tr> <tr> <td>0</td><td>1</td><td>1</td><td>0</td><td>TS6</td> </tr> <tr> <td>0</td><td>1</td><td>1</td><td>1</td><td>TS7</td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>0</td><td>TS8</td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>1</td><td>TS9</td> </tr> <tr> <td>1</td><td>0</td><td>1</td><td>0</td><td>TS10</td> </tr> <tr> <td>1</td><td>0</td><td>1</td><td>1</td><td>TS11</td> </tr> </table> <p>Other than above: Starting measurement operation (CTSUCR0.CTSUSTRT = 1) is prohibited after these bits are set.</p> <p>In other measurement modes, these bits indicate the channel that is currently being measured.</p> <table style="margin-left: 20px; border: none;"> <tr> <td style="padding-right: 10px;">b3</td> <td style="padding-right: 10px;">b2</td> <td style="padding-right: 10px;">b1</td> <td style="padding-right: 10px;">b0</td> <td></td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>TS0</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>TS1</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>TS2</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>TS3</td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>0</td><td>TS4</td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>1</td><td>TS5</td> </tr> <tr> <td>0</td><td>1</td><td>1</td><td>0</td><td>TS6</td> </tr> <tr> <td>0</td><td>1</td><td>1</td><td>1</td><td>TS7</td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>0</td><td>TS8</td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>1</td><td>TS9</td> </tr> <tr> <td>1</td><td>0</td><td>1</td><td>0</td><td>TS10</td> </tr> <tr> <td>1</td><td>0</td><td>1</td><td>1</td><td>TS11</td> </tr> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>Measurement is being stopped</td> </tr> </table>	b3	b2	b1	b0		0	0	0	0	TS0	0	0	0	1	TS1	0	0	1	0	TS2	0	0	1	1	TS3	0	1	0	0	TS4	0	1	0	1	TS5	0	1	1	0	TS6	0	1	1	1	TS7	1	0	0	0	TS8	1	0	0	1	TS9	1	0	1	0	TS10	1	0	1	1	TS11	b3	b2	b1	b0		0	0	0	0	TS0	0	0	0	1	TS1	0	0	1	0	TS2	0	0	1	1	TS3	0	1	0	0	TS4	0	1	0	1	TS5	0	1	1	0	TS6	0	1	1	1	TS7	1	0	0	0	TS8	1	0	0	1	TS9	1	0	1	0	TS10	1	0	1	1	TS11	1	1	1	1	Measurement is being stopped	R/W*1
b3	b2	b1	b0																																																																																																																																								
0	0	0	0	TS0																																																																																																																																							
0	0	0	1	TS1																																																																																																																																							
0	0	1	0	TS2																																																																																																																																							
0	0	1	1	TS3																																																																																																																																							
0	1	0	0	TS4																																																																																																																																							
0	1	0	1	TS5																																																																																																																																							
0	1	1	0	TS6																																																																																																																																							
0	1	1	1	TS7																																																																																																																																							
1	0	0	0	TS8																																																																																																																																							
1	0	0	1	TS9																																																																																																																																							
1	0	1	0	TS10																																																																																																																																							
1	0	1	1	TS11																																																																																																																																							
b3	b2	b1	b0																																																																																																																																								
0	0	0	0	TS0																																																																																																																																							
0	0	0	1	TS1																																																																																																																																							
0	0	1	0	TS2																																																																																																																																							
0	0	1	1	TS3																																																																																																																																							
0	1	0	0	TS4																																																																																																																																							
0	1	0	1	TS5																																																																																																																																							
0	1	1	0	TS6																																																																																																																																							
0	1	1	1	TS7																																																																																																																																							
1	0	0	0	TS8																																																																																																																																							
1	0	0	1	TS9																																																																																																																																							
1	0	1	0	TS10																																																																																																																																							
1	0	1	1	TS11																																																																																																																																							
1	1	1	1	Measurement is being stopped																																																																																																																																							
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W																																																																																																																																							

Note 1. Writing to these bits is enabled only in self-capacitance single scan mode (CTSUCR1.CTSUMD[1:0] bits = 00b).

The CTSUMCH0 register should be set when CTSUCR0.CTSUSTRT bit = 0.

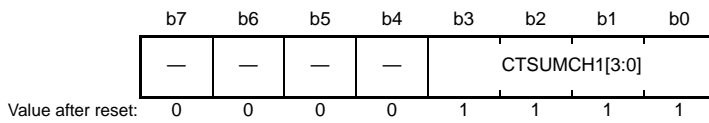
#### CTSUMCH0[3:0] Bits (CTSU Measurement Channel 0)

These bits set the channel to be measured in self-capacitance single scan mode, and indicate the receive channel that is being measured in other modes.

Set only enabled channels (0000b to 1011b) when setting channels in self-capacitance single scan mode. In other modes, writing to these bits has no effect.

### 34.2.6 CTSU Measurement Channel Register 1 (CTSUSMCH1)

Address(es): 000A 0905h



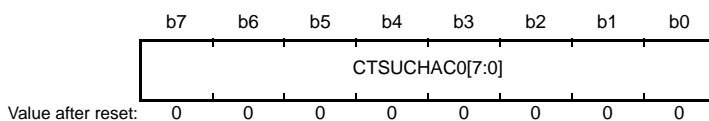
Bit	Symbol	Bit Name	Description	R/W																																										
b3 to b0	CTSUSMCH1[3:0]	CTSUS Measurement Channel 1	These bits indicate the transmit channel that is currently being measured. <table border="0"> <tr> <td>b3</td> <td>b0</td> <td></td> </tr> <tr> <td>0 0 0 0:</td> <td>0:</td> <td>TS0</td> </tr> <tr> <td>0 0 0 1:</td> <td>1:</td> <td>TS1</td> </tr> <tr> <td>0 0 1 0:</td> <td>0:</td> <td>TS2</td> </tr> <tr> <td>0 0 1 1:</td> <td>1:</td> <td>TS3</td> </tr> <tr> <td>0 1 0 0:</td> <td>0:</td> <td>TS4</td> </tr> <tr> <td>0 1 0 1:</td> <td>1:</td> <td>TS5</td> </tr> <tr> <td>0 1 1 0:</td> <td>0:</td> <td>TS6</td> </tr> <tr> <td>0 1 1 1:</td> <td>1:</td> <td>TS7</td> </tr> <tr> <td>1 0 0 0:</td> <td>0:</td> <td>TS8</td> </tr> <tr> <td>1 0 0 1:</td> <td>1:</td> <td>TS9</td> </tr> <tr> <td>1 0 1 0:</td> <td>0:</td> <td>TS10</td> </tr> <tr> <td>1 0 1 1:</td> <td>1:</td> <td>TS11</td> </tr> <tr> <td>1 1 1 1:</td> <td>1:</td> <td>Measurement is being stopped</td> </tr> </table>	b3	b0		0 0 0 0:	0:	TS0	0 0 0 1:	1:	TS1	0 0 1 0:	0:	TS2	0 0 1 1:	1:	TS3	0 1 0 0:	0:	TS4	0 1 0 1:	1:	TS5	0 1 1 0:	0:	TS6	0 1 1 1:	1:	TS7	1 0 0 0:	0:	TS8	1 0 0 1:	1:	TS9	1 0 1 0:	0:	TS10	1 0 1 1:	1:	TS11	1 1 1 1:	1:	Measurement is being stopped	R/W
b3	b0																																													
0 0 0 0:	0:	TS0																																												
0 0 0 1:	1:	TS1																																												
0 0 1 0:	0:	TS2																																												
0 0 1 1:	1:	TS3																																												
0 1 0 0:	0:	TS4																																												
0 1 0 1:	1:	TS5																																												
0 1 1 0:	0:	TS6																																												
0 1 1 1:	1:	TS7																																												
1 0 0 0:	0:	TS8																																												
1 0 0 1:	1:	TS9																																												
1 0 1 0:	0:	TS10																																												
1 0 1 1:	1:	TS11																																												
1 1 1 1:	1:	Measurement is being stopped																																												
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W																																										

#### CTSUSMCH1[3:0] Bits (CTSUS Measurement Channel 1)

These bits indicate the transmit channel that is being measured in mutual capacitance simple scan mode and full scan mode. The value of these bits is 1111b while measurement is stopped or in self-capacitance single scan mode and multi-scan mode.

### 34.2.7 CTSU Channel Enable Control Register 0 (CTSUSCHAC0)

Address(es): 000A 0906h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CTSUSCHAC0[7:0]	CTSUS Channel Enable Control 0	These bits set whether the TS pins are to be measured. 0: Not measurement target 1: Measurement target  These bits specify the TS0 to TS7 pins.	R/W

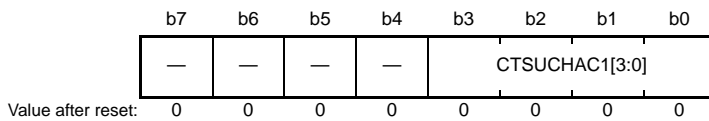
The CTSUSCHAC0 register should be set when the CTSUCR0.CTUSSTRT bit = 0.

#### CTSUSCHAC0[7:0] Bits (CTSUS Channel Enable Control 0)

These bits set the pins (for receive and transmit) whose electrostatic capacitance is to be measured. CTSUSCHAC0[0] corresponds to TS0 and CTSUSCHAC0[7] corresponds to TS7.

### 34.2.8 CTSU Channel Enable Control Register 1 (CTSUCHAC1)

Address(es): 000A 0907h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CTSUCHAC1[3:0]	CTSU Channel Enable Control 1	These bits set whether the TS pins are to be measured. 0: Not measurement target 1: Measurement target  These bits specify the TS8 to TS11 pins.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The CTSUCHAC1 register should be set when the CTSUCR0.CTSUSTRT bit = 0.

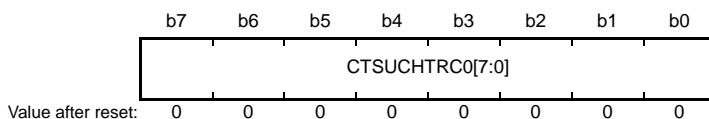
#### CTSUCHAC1[3:0] Bits (CTSU Channel Enable Control 1)

These bits set the pins (for receive and transmit) whose electrostatic capacitance is to be measured.

CTSUCHAC1[0] corresponds to TS8 and CTSUCHAC1[3] corresponds to TS11.

### 34.2.9 CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0)

Address(es): 000A 090Bh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CTSUCHTRC0[7:0]	CTSU Channel Transmit/Receive Control 0	These bits allocate reception or transmission to the TS pins in mutual capacitance simple scan mode and full scan mode. 0: Reception 1: Transmission  These bits specify the TS0 to TS7 pins.	R/W

The CTSUCHTRC0 register should be set when the CTSUCR0.CTSUSTRT bit = 0.

#### CTSUCHTRC0[7:0] Bits (CTSU Channel Transmit/Receive Control 0)

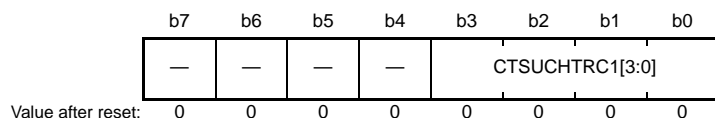
These bits allocate reception or transmission to the TS pins in mutual capacitance simple scan mode and full scan mode.

The setting of these bits is ignored in self-capacitance single scan mode and multi-scan mode.

CTSUCHTRC0[0] corresponds to TS0 and CTSUCHTRC0[7] corresponds to TS7.

### 34.2.10 CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1)

Address(es): 000A 090Ch



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CTSUCHTRC1[3:0]	CTSUS Channel Transmit/Receive Control 1	These bits allocate reception or transmission to the TS pins in mutual capacitance simple scan mode and full scan mode. 0: Reception 1: Transmission  These bits specify the TS8 to TS11 pins.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

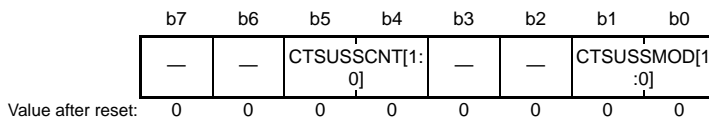
The CTSUCHTRC1 register should be set when the CTSUCR0.CTSUSTRTR bit = 0.

#### CTSUCHTRC1[3:0] Bits (CTSUS Channel Transmit/Receive Control 1)

These bits allocate reception or transmission to the TS pins in mutual capacitance simple scan mode and full scan mode. The setting of these bits is ignored in self-capacitance single scan mode and multi-scan mode. CTSUCHTRC1[0] corresponds to TS8 and CTSUCHTRC1[3] corresponds to TS11.

### 34.2.11 CTSU High-Pass Noise Reduction Control Register (CTSUDCLKC)

Address(es): 000A 0910h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	CTSUSSMOD[1:0]	CTSU Diffusion Clock Mode Select	These bits should be set to 00b.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	CTSUSSCNT[1:0]	CTSU Diffusion Clock Control	These bits should be set to 11b.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The CTSUDCLKC register should be set when the CTSUCR0.CTSUSTRT bit = 0.

#### CTSUSSMOD[1:0] Bits (CTSU Diffusion Clock Mode Select)

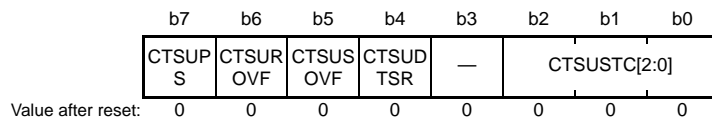
These bits set the mode of the spectrum diffusion clock for high-pass noise reduction. When using the high-pass function, the value of these bits should be fixed to 00b. If these bits are not set, the effect of high-pass noise reduction cannot be correctly obtained.

#### CTSUSSCNT[1:0] Bits (CTSU Diffusion Clock Control)

These bits adjust the spectrum diffusion amount to reduce high-pass noise. When using the high-pass noise reduction function, the value of these bits should be fixed to 11b. If these bits are not set, touch measurement may not be correctly performed.

### 34.2.12 CTSU Status Register (CTSUST)

Address(es): 000A 0911h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSUSTC[2:0]	CTSUS Measurement Status Counter	These counters indicate the current measurement status. b2 b0 0 0 0: Status 0 0 0 1: Status 1 0 1 0: Status 2 0 1 1: Status 3 1 0 0: Status 4 1 0 1: Status 5	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	CTSUDTSR	CTSUS Data Transfer Status Flag	This flag indicates whether the measurement result stored in the sensor counter and the reference counter has been read. 0: Measurement result has been read 1: Measurement result has not been read	R
b5	CTSUSOVF	CTSUS Sensor Counter Overflow Flag	This flag indicates whether the sensor counter has overflowed. 0: No overflow 1: An overflow	R/W
b6	CTSUROVF	CTSUR Reference Counter Overflow Flag	This flag indicates whether the reference counter has overflowed. 0: No overflow 1: An overflow	R/W
b7	CTSUPS	CTSUS Mutual Capacitance Status Flag	This flag indicates the measurement status in mutual capacitance full scan mode. 0: First measurement 1: Second measurement	R

When using the CTSUCR0.CTSUINIT bit to clear an overflow flag, make sure that the CTSUCR0.CTSUSTRT bit = 0.

#### CTSUSTC[2:0] Counters (CTSUS Measurement Status Counter)

These counters indicate the current measurement status. For details on each status, refer to section 34.3.2.2, Status Counter.

#### CTSUDTSR Flag (CTSUS Data Transfer Status Flag)

This flag indicates whether the measurement result stored in the sensor counter and the reference counter has been read. This flag is set to 1 when measurement is completed; 0 when the reference counter is read by software or the DTC. This flag is also cleared using the CTSUCR0.CTSUINIT bit.

#### CTSUSOVF Flag (CTSUS Sensor Counter Overflow Flag)

This flag indicates whether the sensor counter has overflowed. FFFFh can be read as the measurement result (CTSUSC counter) when an overflow has occurred.

Even if an overflow occurs, measurement processing is continued until the set period.

No interrupt is generated even when an overflow occurs. To determine the channel on which the overflow has occurred, read the measurement result of each channel after measurement is completed (after a measurement end interrupt is generated).



This flag is cleared when 0 is written after 1 is read by software. This flag is also cleared using the CTSUCR0.CTSUINIT bit.

**CTSUROVF Flag (CTSU Reference Counter Overflow Flag)**

This flag indicates whether the reference counter has overflowed. FFFFh can be read as the measurement result (CTSURC counter) when an overflow has occurred.

Even if an overflow occurs, measurement processing is continued until the set period.

No interrupt is generated even when an overflow occurs. To determine the channel on which the overflow has occurred, read the measurement result of each channel after measurement is completed (after a measurement end interrupt is generated).

This flag is cleared when 0 is written after 1 is read by software. This flag is also cleared using the CTSUCR0.CTSUINIT bit.

**CTSUPS Flag (CTSU Mutual Capacitance Status Flag)**

This flag indicates whether the measurement is the first or second of two measurements for each channel in mutual capacitance full scan mode (CTSUCR1.CTSUMD[1:0] bits = 11b).

This flag indicates 0 while measurement is stopped or in other measurement modes.

### 34.2.13 CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register (CTSUSSC)

Address(es): 000A 0912h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
—	—	—	—	CTSUSSDIV[3:0]				—	—	—	—	—	—	—	—	—
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
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	CTSUSSDIV[3:0]	CTSUS Spectrum Diffusion Frequency Division Setting	These bits specify the spectrum diffusion frequency division setting according to the base clock frequency division setting.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### CTSUSSDIV[3:0] Bits (CTSUS Spectrum Diffusion Frequency Division Setting)

These bits specify the spectrum diffusion frequency division setting according to the base clock frequency division setting. See the relationship between base clock frequencies and CTSUSSDIV[3:0] bit settings in Table 34.5, for setting the value of these bits.

**Table 34.5 Relationship between Base Clock Frequencies and CTSUSSDIV[3:0] Bit Settings**

Base Clock Frequency fb (MHz)	CTSUSSDIV[3:0] Bit Setting
$4.00 \leq fb$	0000b
$2.00 \leq fb < 4.00$	0001b
$1.33 \leq fb < 2.00$	0010b
$1.00 \leq fb < 1.33$	0011b
$0.80 \leq fb < 1.00$	0100b
$0.67 \leq fb < 0.80$	0101b
$0.57 \leq fb < 0.67$	0110b
$0.50 \leq fb < 0.57$	0111b
$0.44 \leq fb < 0.50$	1000b
$0.40 \leq fb < 0.44$	1001b
$0.36 \leq fb < 0.40$	1010b
$0.33 \leq fb < 0.36$	1011b
$0.31 \leq fb < 0.33$	1100b
$0.29 \leq fb < 0.31$	1101b
$0.27 \leq fb < 0.29$	1110b
$fb \leq 0.25$	1111b

### 34.2.14 CTSU Sensor Offset Register 0 (CTSUSO0)

Address(es): 000A 0914h



Bit	Symbol	Bit Name	Description	R/W
b9 to b0	CTSUSO[9:0]	CTSU Sensor Offset Adjustment	These bits adjust the electronic capacitance while the electrode is not being touched. b9 b0 0 0 0 0 0 0 0 0 0: Current offset amount is 0 0 0 0 0 0 0 0 0 1: Current offset amount is 1 0 0 0 0 0 0 0 0 1 0: Current offset amount is 2 : 1 1 1 1 1 1 1 1 1 0: Current offset amount is 1022 1 1 1 1 1 1 1 1 1 1: Current offset amount is maximum	R/W
b15 to b10	CTSUSNUM[5:0]	CTSU Measurement Count Setting	These bits set the number of measurements.	R/W

#### CTSUSO[9:0] Bits (CTSU Sensor Offset Adjustment)

These control bits adjust the input current offset of the sensor ICO. These bits are used to offset the sensor ICO input current generated from electrostatic capacitance while the electrode is being not touched during touch measurement, thus preventing overflow of the CTSU sensor counter.

Make settings for the TS pin that is to be measured next after a CTSUWR interrupt is generated.

#### CTSUSNUM[5:0] Bits (CTSU Measurement Count Setting)

These bits set how many times the number of measurement pulses specified by the CTSUSDPRS.CTUPRRATIO[3:0] and CTSUSDPRS.CTUPRMODE[1:0] bits is repeated in the measurement time. The number of measurement pulses is repeated (CTSUSNUM[5:0] bits + 1) times.

Make settings for the TS pin that is to be measured next after a CTSUWR interrupt is generated.

### 34.2.15 CTSU Sensor Offset Register 1 (CTSUSO1)

Address(es): 000A 0916h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CTSURICOA[7:0]	CTSUS Reference ICO Current Adjustment	These bits adjust the input current of the reference ICO. b7 b0 0 0 0 0 0 0 0 0: Input current amount 0 0 0 0 0 0 0 0 1: Input current amount 1 0 0 0 0 0 1 0 0: Input current amount 2 : : 1 1 1 1 1 1 1 0: Input current amount 254 1 1 1 1 1 1 1 1: Input current amount maximum	R/W
b12 to b8	CTSUSDPA[4:0]	CTSUS Base Clock Setting	These bits are used to generate the base clock. b12 b8 0 0 0 0 0: Operating clock divided by 2*1 0 0 0 0 1: Operating clock divided by 4 0 0 0 1 0: Operating clock divided by 6 0 0 0 1 1: Operating clock divided by 8 0 0 1 0 0: Operating clock divided by 10 0 0 1 0 1: Operating clock divided by 12 0 0 1 1 0: Operating clock divided by 14 0 0 1 1 1: Operating clock divided by 16 0 1 0 0 0: Operating clock divided by 18 0 1 0 0 1: Operating clock divided by 20 0 1 0 1 0: Operating clock divided by 22 0 1 0 1 1: Operating clock divided by 24 0 1 1 0 0: Operating clock divided by 26 0 1 1 0 1: Operating clock divided by 28 0 1 1 1 0: Operating clock divided by 30 0 1 1 1 1: Operating clock divided by 32 1 0 0 0 0: Operating clock divided by 34 1 0 0 0 1: Operating clock divided by 36 1 0 0 1 0: Operating clock divided by 38 1 0 0 1 1: Operating clock divided by 40 1 0 1 0 0: Operating clock divided by 42 1 0 1 0 1: Operating clock divided by 44 1 0 1 1 0: Operating clock divided by 46 1 0 1 1 1: Operating clock divided by 48 1 1 0 0 0: Operating clock divided by 50 1 1 0 0 1: Operating clock divided by 52 1 1 0 1 0: Operating clock divided by 54 1 1 0 1 1: Operating clock divided by 56 1 1 1 0 0: Operating clock divided by 58 1 1 1 0 1: Operating clock divided by 60 1 1 1 1 0: Operating clock divided by 62 1 1 1 1 1: Operating clock divided by 64	R/W
b14, b13	CTSUICOG[1:0]	CTSUS ICO Gain Adjustment	These bits adjust the output frequency gain of the sensor ICO and the reference ICO. b14 b13 0 0: 100% gain 0 1: 66% gain 1 0: 50% gain 1 1: 40% gain	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The CTSUSDPA[4:0] bits should not be set to 00000b while the high-pass noise reduction function is turned off (CTSUSDPRS.CTSUSOFF bit = 1) in mutual capacitance full scan mode (CTSUCR1.CTSUMD[1:0] bits = 11b).

Write first to the CTSUSSC register, then CTSUSO0 register, and then CTSUSO1 register after a CTSUWR interrupt is

generated. Write operation to the CTSUSO1 register causes a transition to Status 3. Thus, set all the bits in a single setting when writing to the CTSUSO1 register.

#### CTSURICOA[7:0] Bits (CTSU Reference ICO Current Adjustment)

These bits adjust the oscillation frequency using the input current of the reference ICO.

#### CTSUSDPA[4:0] Bits (CTSU Base Clock Setting)

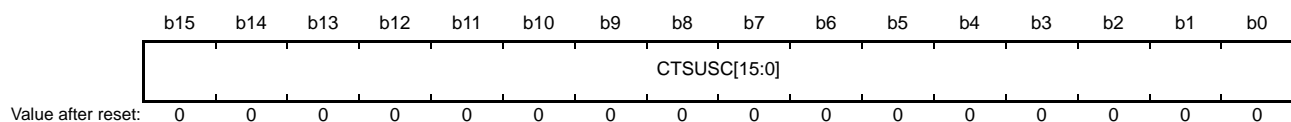
These bits are used to generate a base clock used as the source for the sensor drive pulse by dividing the operating clock. Set these bits so that the base clock is 4 MHz or lower. For details on the setting procedure, refer to section 34.3.2.1, Initial Setting Flowchart.

#### CTSUICOG[1:0] Bits (CTSU ICO Gain Adjustment)

These bits adjust the output frequency gain of the sensor ICO and the reference ICO. Normally, the value of these bits should be set to 00b for the maximum gain. If changes in the capacitance between when the electrode is not touched and touched greatly exceed the dynamic range of the sensor ICO, set the gain adjustment bits to adjust the gain appropriately.

### 34.2.16 CTSU Sensor Counter (CTSUSC)

Address(es): 000A 0918h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CTSUSC[15:0]	CTSU Sensor Counter	These bits indicate the measurement result of the CTSU. These bits indicate FFFFh when an overflow occurs.	R/W

Read first from the CTSUSC counter and then the CTSURC counter after a CTSURD interrupt is generated.

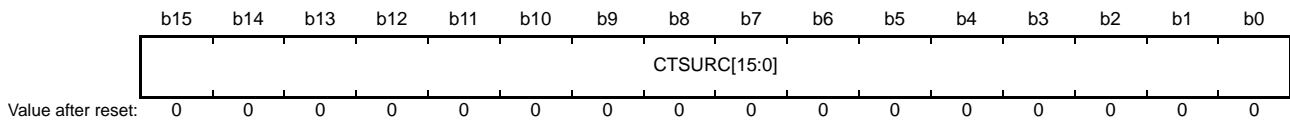
#### CTSUSC[15:0] Bits (CTSU Sensor Counter)

These bits are configured as an increment counter that counts the sensor ICO clock.

Read these bits after a CTSURD interrupt is generated. After the CTSURC counter is read, these bits are cleared immediately before the CTSUST.CTSUSTC[2:0] counter value changes to Status 4 in the next measurement. These bits are also cleared using the CTSUCR0.CTSUINIT bit.

### 34.2.17 CTSU Reference Counter (CTSURC)

Address(es): 000A 091Ah



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CTSURC[15:0]	CTSU Reference Counter	These bits indicate the measurement result of the reference ICO. These bits indicate FFFFh when an overflow occurs.	R/W

Read first from the CTSUSC counter and then the CTSURC counter after a CTSURD interrupt is generated. Even when the stabilization time specified for Status 3 has elapsed, if the CTSURC counter is not read, Status 3 continues until the counter is read.

#### CTSURC[15:0] Bits (CTSU Reference Counter)

These bits are configured as an increment counter that counts the reference ICO clock.

The reference ICO is used to optimize touch measurement performed using the sensor ICO. There is some deviation depending on the internal sensor ICO and the reference ICO in the CTSU, but the same ICOs have almost the same characteristics, and the dynamic range and the current to frequency characteristics are almost the same. The range of current amount that can be set by the reference ICO current adjustment bits is about the same as the ICO dynamic range, the current amount input to the sensor ICO must be within this dynamic range. First, use the reference ICO to check the differences between the ICOs and measure the current to oscillation frequency characteristics. Since the reference ICO oscillation frequency can be obtained from the reference ICO counter, the ICO oscillation frequency (counter value/ measurement time) for the input current amount can be measured by setting the value in the reference ICO current adjustment bits and measuring the reference ICO counter. The reference ICO counter value measured using the maximum value of the reference ICO current adjustment bits is the maximum value of the ICO dynamic range. Therefore, the current amount of the sensor ICO needs to be offset by setting the offset adjustment bits so that the sensor ICO counter value does not exceed this value.

Read the CTSURC[15:0] bits after a CTSURD interrupt is generated. After these bits are read, they are cleared immediately before the CTSUST.CTSUSTC[2:0] counter value changes to Status 4 in the next measurement. These bits are also cleared using the CTSUCR0.CTSUINIT bit.

### 34.2.18 CTSU Error Status Register (CTSUERRS)

Address(es): 000A 091Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUI COMP	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	CTSUICOMP	TSCAP Voltage Error Monitor	This bit monitors the error status of the TSCAP voltage. 0: Normal TSCAP voltage 1: Abnormal TSCAP voltage	R/W

#### CTSUICOMP Bit (TSCAP Voltage Error Monitor)

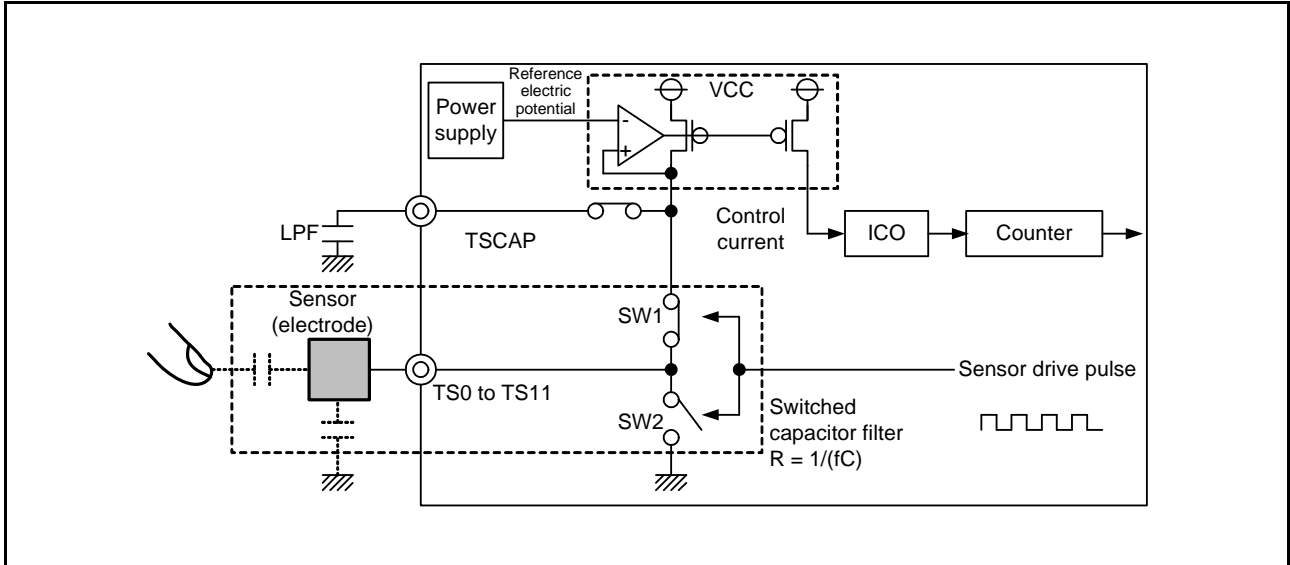
If the offset current amount set by the CTSUSO1 register exceeds the sensor ICO input current during touch measurement, the TSCAP voltage becomes abnormal and touch measurement cannot be correctly performed. This bit monitors the TSCAP voltage and it is set to 1 if the voltage becomes abnormal. If the TSCAP voltage becomes abnormal, the sensor ICO counter value will be undefined, but touch measurement is normally completed, so it difficult to detect an abnormality by reading the sensor ICO counter value. If the CTSU reference ICO current adjustment bits (CTSURICOA[7:0]) in the CTSUSO1 register are set to a value other than 0, check this bit when touch measurement is completed.

This bit is cleared by writing 0 to the CTSUCR1.CTSUPON bit and turning off the power supply.

### 34.3 Operation

#### 34.3.1 Principles of Measurement Operation

Figure 34.4 shows the measurement circuit.

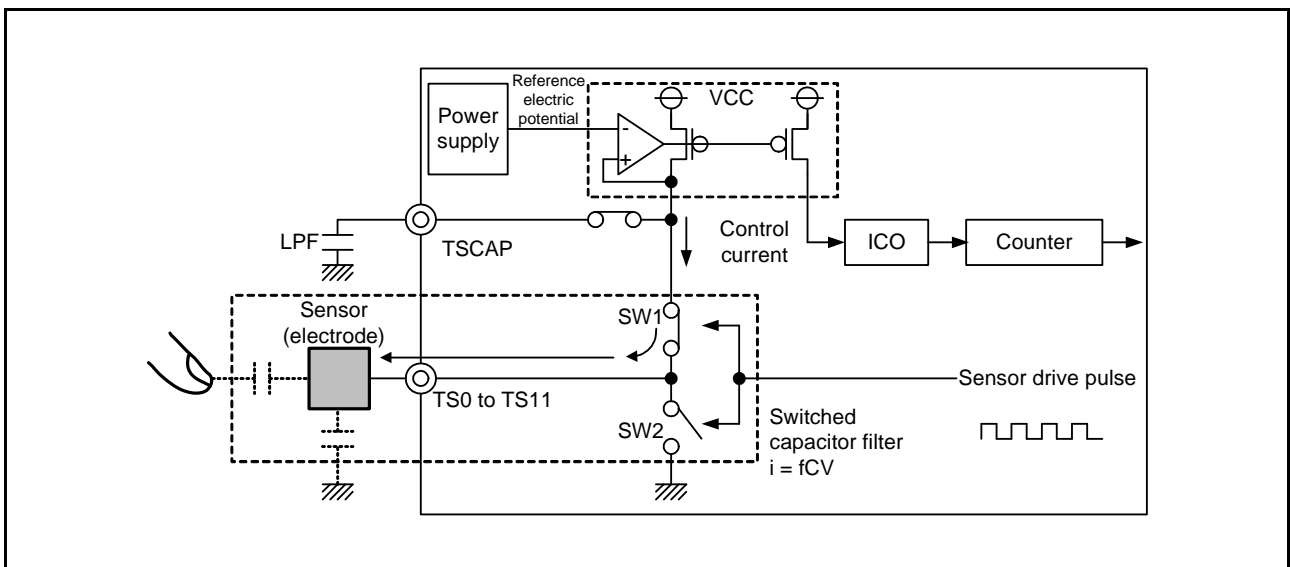


**Figure 34.4 Measurement Circuit**

The electrostatic capacitance measurement operation principles of the CTSU current frequency conversion method are explained using Figure 34.5 to Figure 34.7.

- (1) The electrostatic capacitance of the electrode is charged by turning SW1 on and SW2 off (Figure 34.5).
- (2) The charged capacitance is discharged by turning SW1 off and SW2 on (Figure 34.6).

Current flows to the switched capacitor filter by switching between charging and discharging in steps (1) and (2). At this time, the value of electrostatic capacitance varies depending on whether a finger is in close proximity, so the flowing current changes. A clock is generated by supplying the control current, which is proportional to the amount of the current flowing through the switched capacitor filter, from the circuit that generates the TSCAP power supply to the ICO. The counter is used to measure the clock frequency which changes depending on whether a finger is in close proximity, and the value read from the counter is used by software to determine contact with a finger (Figure 34.7).



**Figure 34.5 Charging Operation**



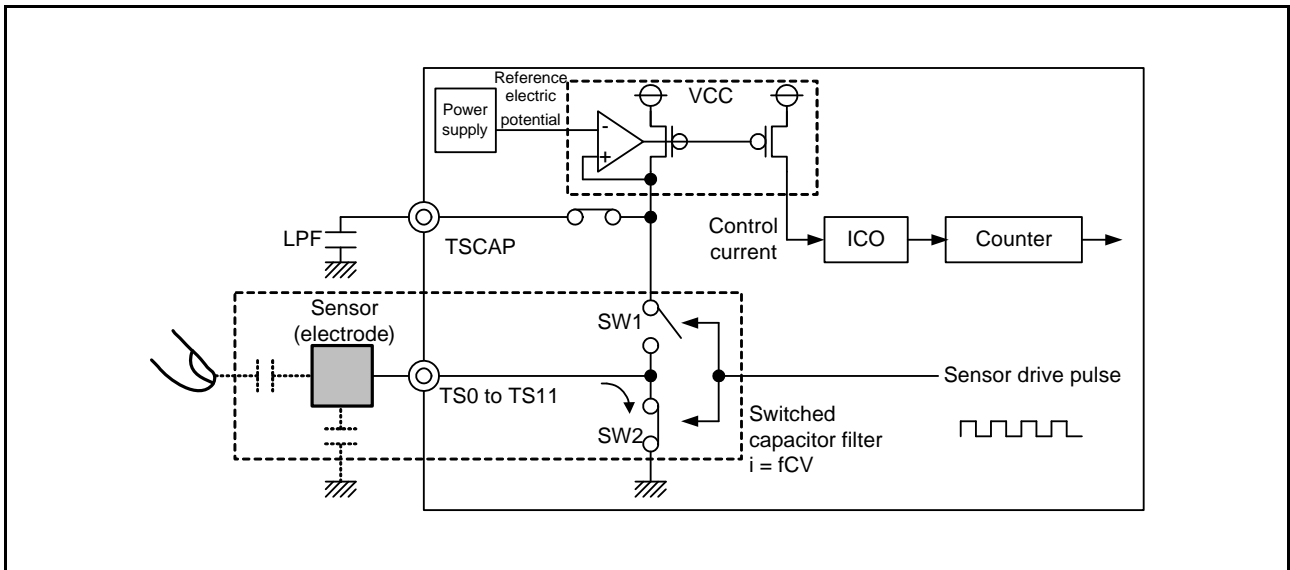


Figure 34.6 Discharging Operation

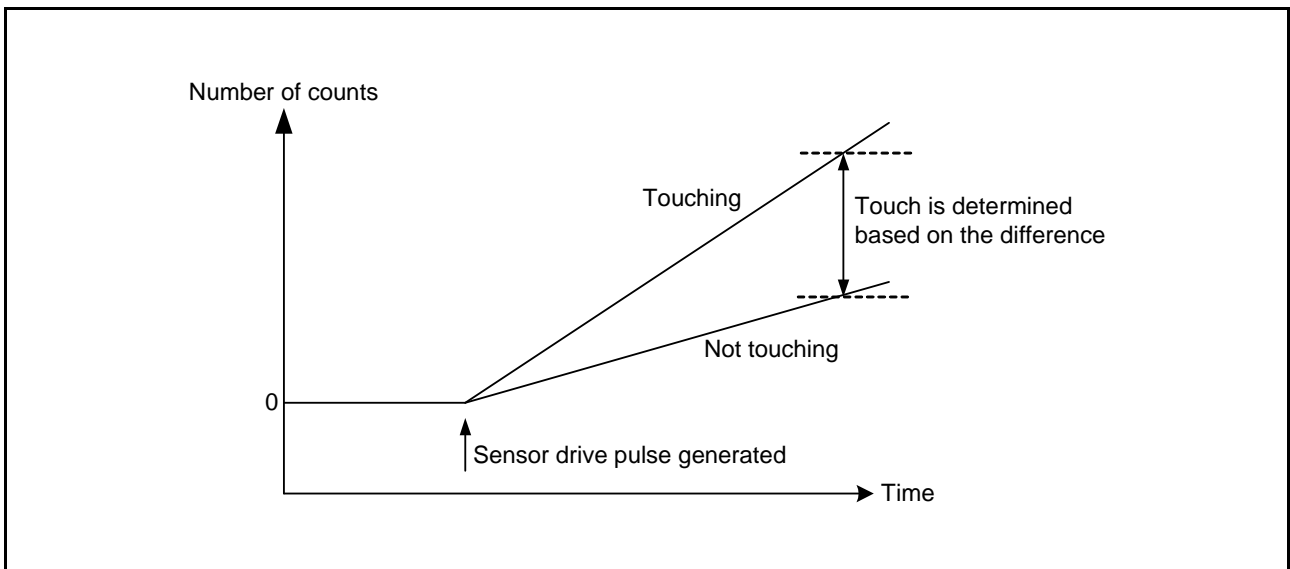
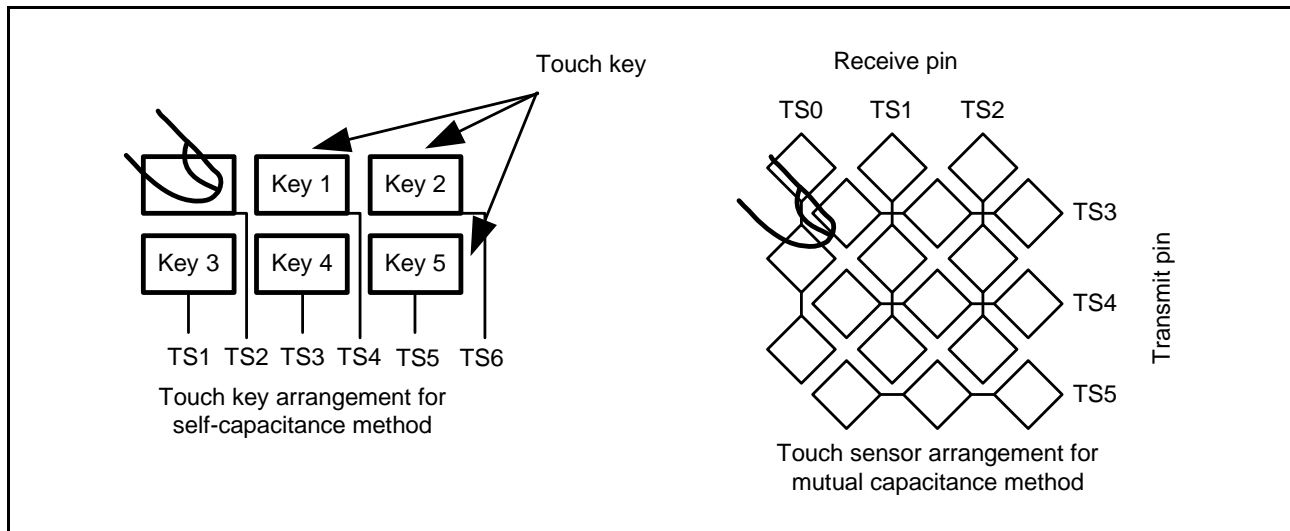


Figure 34.7 Change in Measured Value When Finger is Touching and Not Touching

### 34.3.2 Measurement Modes

The CTSU supports self-capacitance and mutual capacitance methods. Figure 34.8 illustrates these methods.



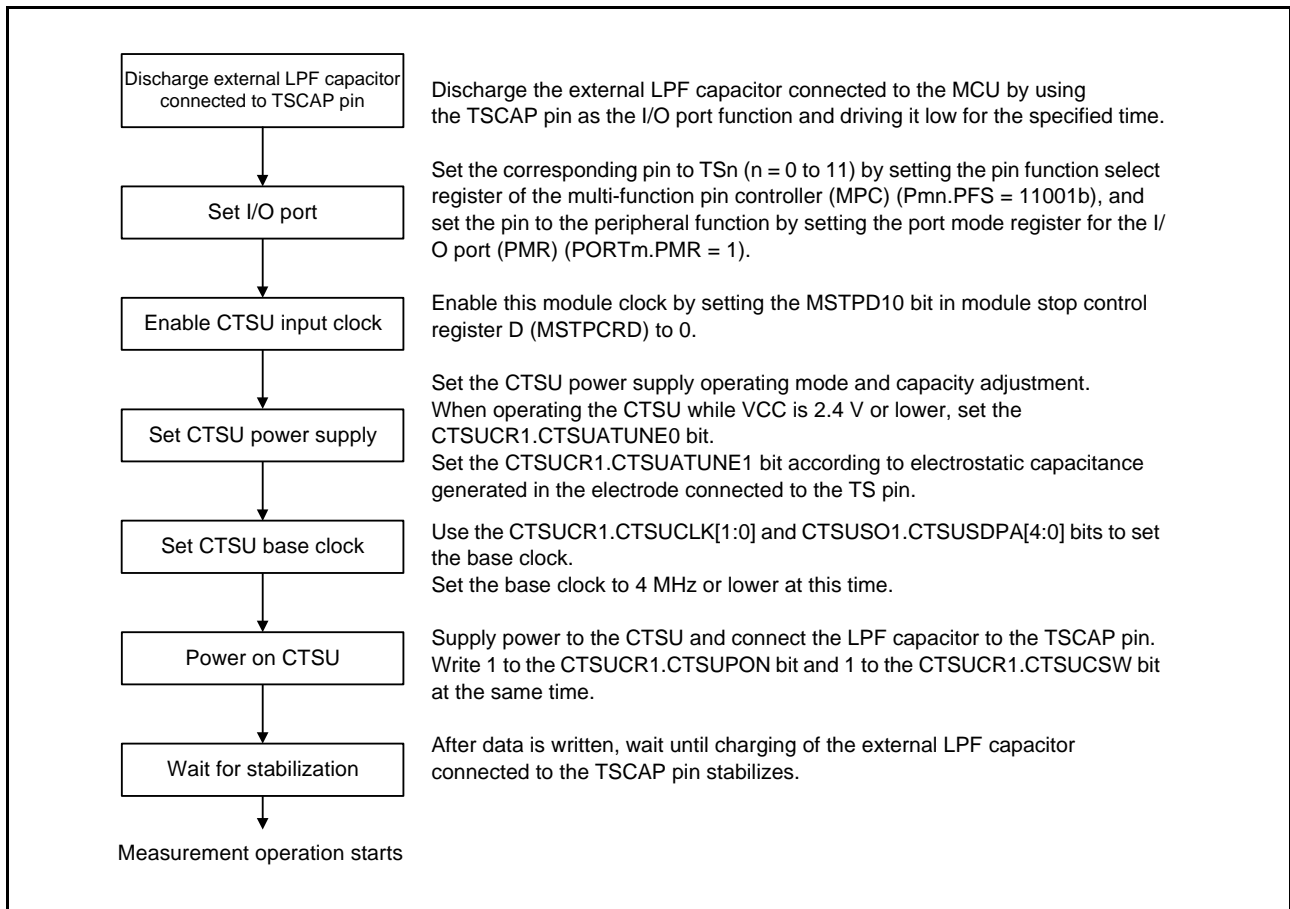
**Figure 34.8 Overview of Self-Capacitance Method and Mutual Capacitance Method**

In self-capacitance method, a single touch pin is allocated to a single touch key to measure individual electrostatic capacitance when a finger is in close proximity. In this method, single scan and multi-scan can be used as measurement modes.

In mutual capacitance method, the capacitance between two opposite electrodes (transmit and receive pins) is measured. In this method, simple scan and full scan can be used as measurement modes.

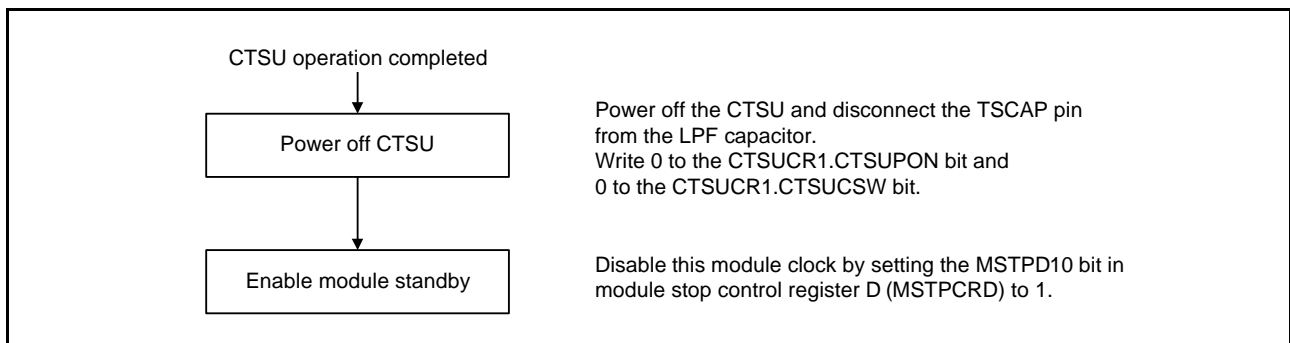
### 34.3.2.1 Initial Setting Flowchart

Figure 34.9 shows the flowchart for CTSU initial setting.



**Figure 34.9 CTSU Initial Setting Flowchart**

Figure 34.10 shows the flowchart for stopping CTSU operation and setting to the standby state.

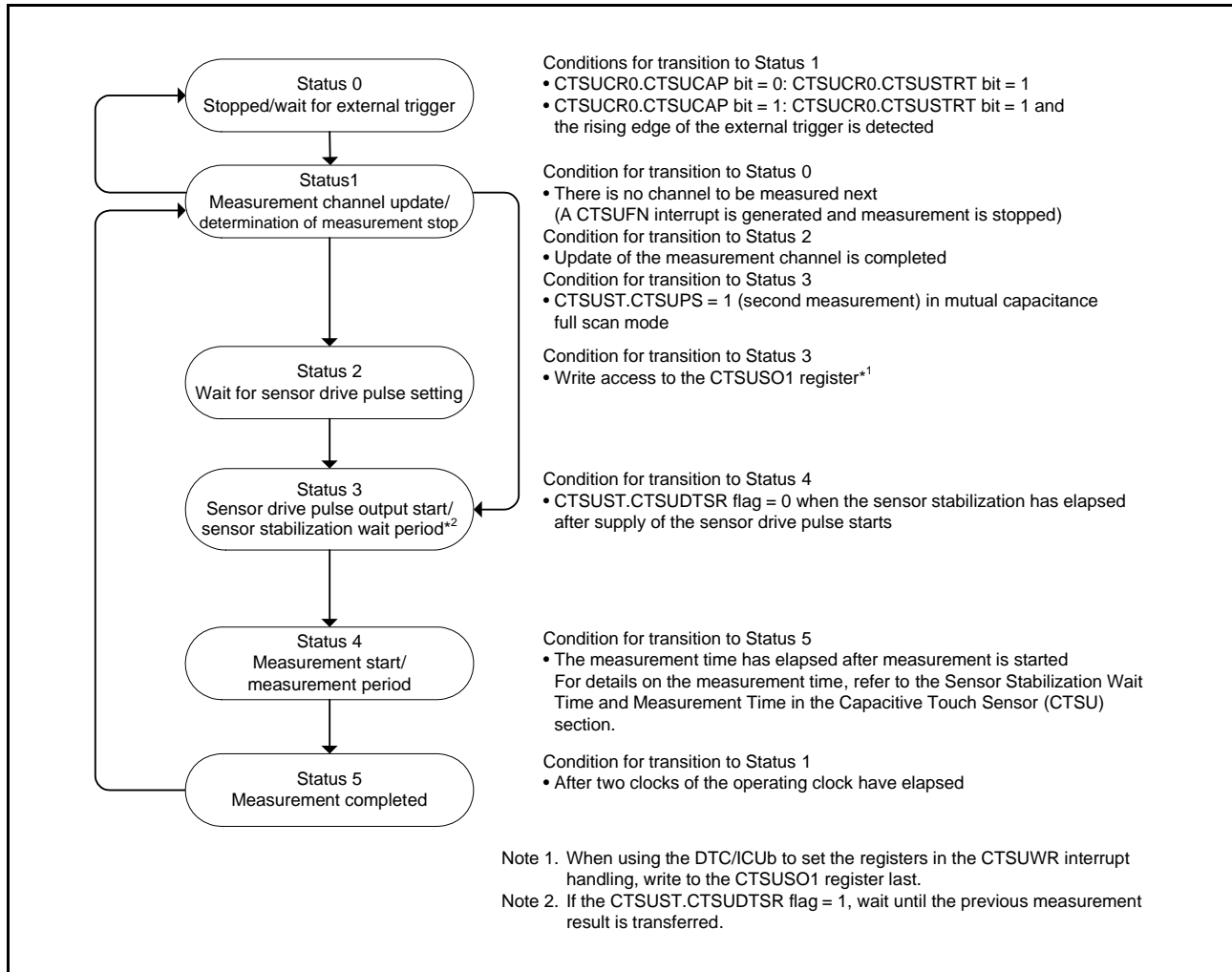


**Figure 34.10 CTSU Stopping Flowchart**

When restarting operation after it has been stopped, follow the initial setting flowchart shown in Figure 34.9.

### 34.3.2.2 Status Counter

The measurement status counter of the CTSU status register (CTSUST) indicates the current measurement status. The measurement status is common to all four modes. Figure 34.11 shows status operation transitions.



**Figure 34.11 Status Operation Transitions**

The status of the status counter transitions to Status 0 when all of the specified measurement channels are measured.

The CTSUCR0.CTSUSTRT bit is cleared to 0 by hardware when a software trigger is used. When an external trigger is used, the value 1 is retained, and the CTSU waits for the next trigger.

When operation is forcibly stopped (by writing 0 to the CTSUCR0.CTSUSTRT bit and 1 to CTSUCR0.CTSUINIT bit at the same time) during measurement or the wait state for the trigger, the status transitions to Status 0 and measurement is stopped forcibly.

If there is no channel to be measured by setting the CTSUMCH0, CTSUCHAC0, CTSUCHAC1, CTSUCHTRC0, and CTSUCHTRC1 registers, a CTSUFN interrupt is generated immediately after a transition to Status 1, and then the status transitions to Status 0. The following are the cases when there is no channel to be measured.

- A measurement target channel is not specified by the CTSUCHAC0 and CTSUCHAC1 registers.
- In self-capacitance single scan mode, the channel specified in the CTSUMCH0 register is not a measurement target in the CTSUCHAC0 and CTSUCHAC1 registers.
- In mutual capacitance simple scan and full scan modes, there is no transmit channel or receive channel to be measured by combining the CTSUCHAC0, CTSUCHAC1, CTSUCHTRC0, and CTSUCHTRC1 registers.

### 34.3.2.3 Self-Capacitance Single Scan Mode Operation

In self-capacitance single scan mode, electrostatic capacitance on a channel is measured. Figure 34.12 shows the software flowchart and an operation example, and Figure 34.13 shows the timing chart.

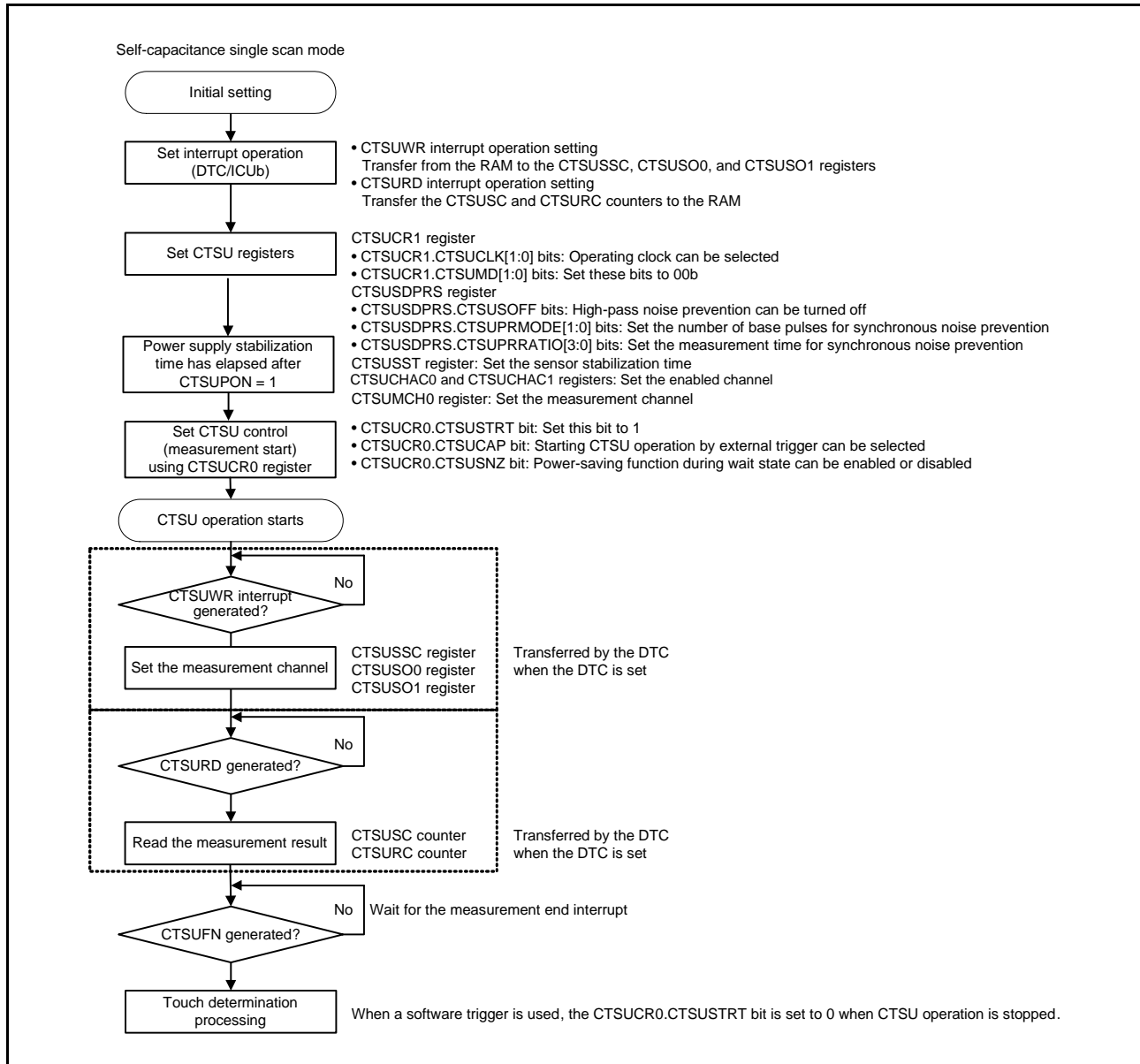
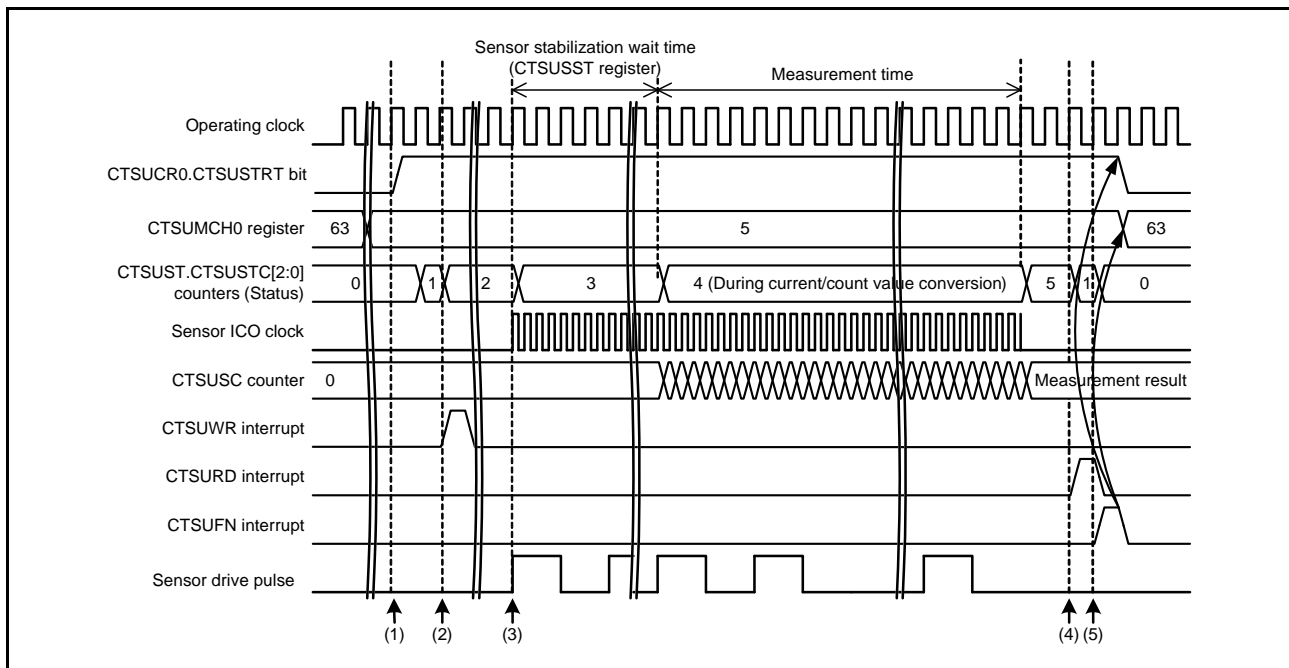


Figure 34.12 Software Flowchart and Operation Example of Self-Capacitance Single Scan Mode



**Figure 34.13 Timing Chart of Self-Capacitance Single Scan Mode (Measurement Start Condition is Software Trigger)**

The following describes operation shown in the timing chart in Figure 34.13.

- (1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.
- (2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.
- (3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.
- (4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is stopped, a measurement result read request (CTSURD) is output.
- (5) A measurement end interrupt (CTSUFN) is output and measurement is stopped (transition to Status 0).

Table 34.6 lists the touch pin states in self-capacitance single scan mode.

**Table 34.6 Touch Pin States in Self-Capacitance Single Scan Mode**

Status	Touch Pin		Remarks
	Measurement Channel	Non-Measurement Channel	
0	Low	Low	Normal I/O is driven low
1	Low	Low	Normal I/O is driven low
2	Low	Low	Normal I/O is driven low
3	Pulse	Low	Measurement channel: Touch I/O is pulse, normal I/O is Hi-Z Non-measurement channel: Normal I/O is driven low
4	Pulse	Low	Measurement channel: Touch I/O is pulse, normal I/O is Hi-Z Non-measurement channel: Normal I/O is driven low
5	Low	Low	Normal I/O is driven low The pin state in Status 4 changes to this state (all normal I/O being driven low) after one operating clock cycle.

### 34.3.2.4 Self-Capacitance Multi-Scan Mode Operation

In self-capacitance multi-scan mode, electrostatic capacitance on all channels that are specified as measurement targets by setting the CTSUCHAC0 and CTSUCHAC1 registers are measured sequentially in ascending order. Figure 34.14 shows the software flowchart and an operation example, and Figure 34.15 shows the timing chart.

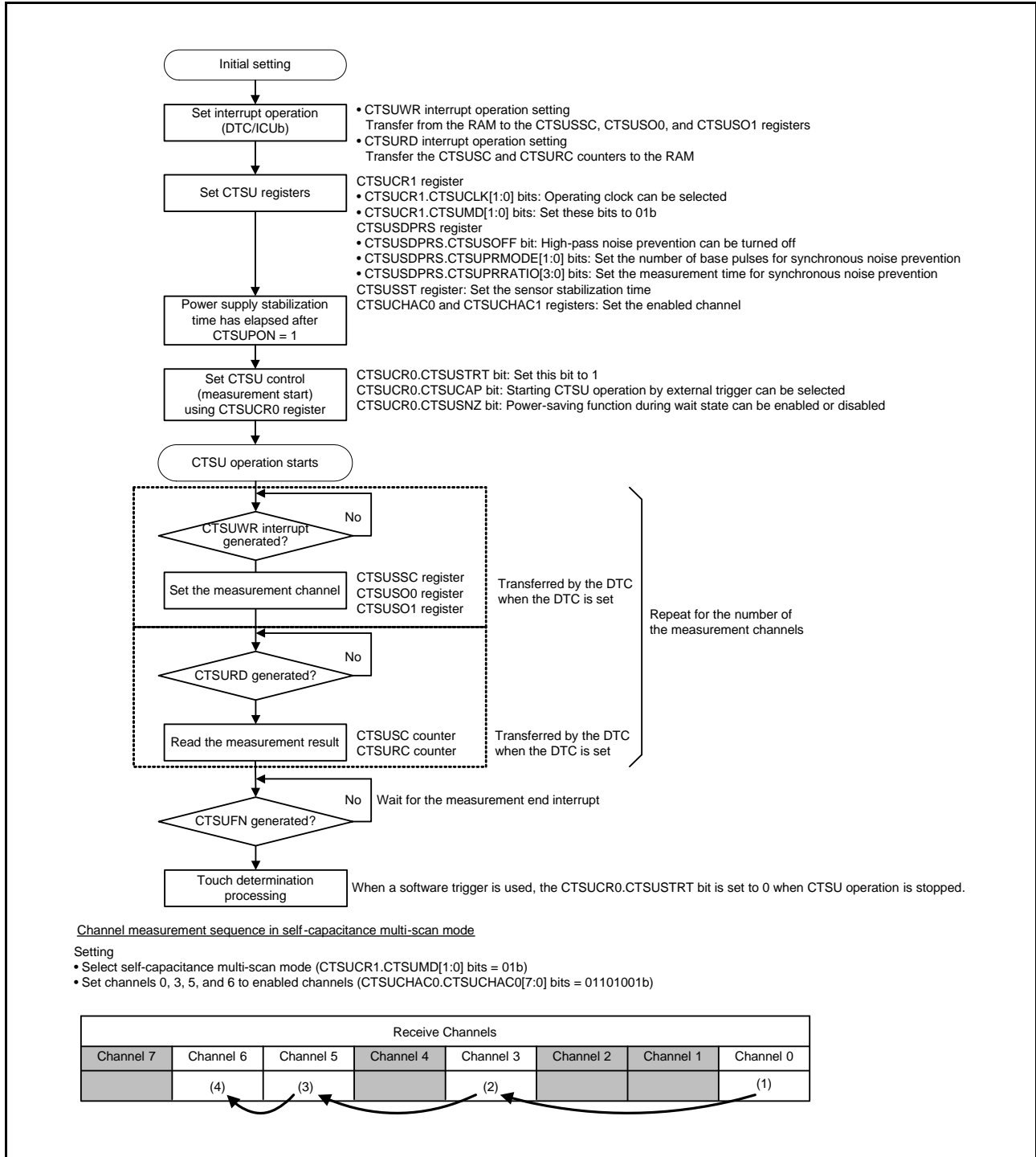
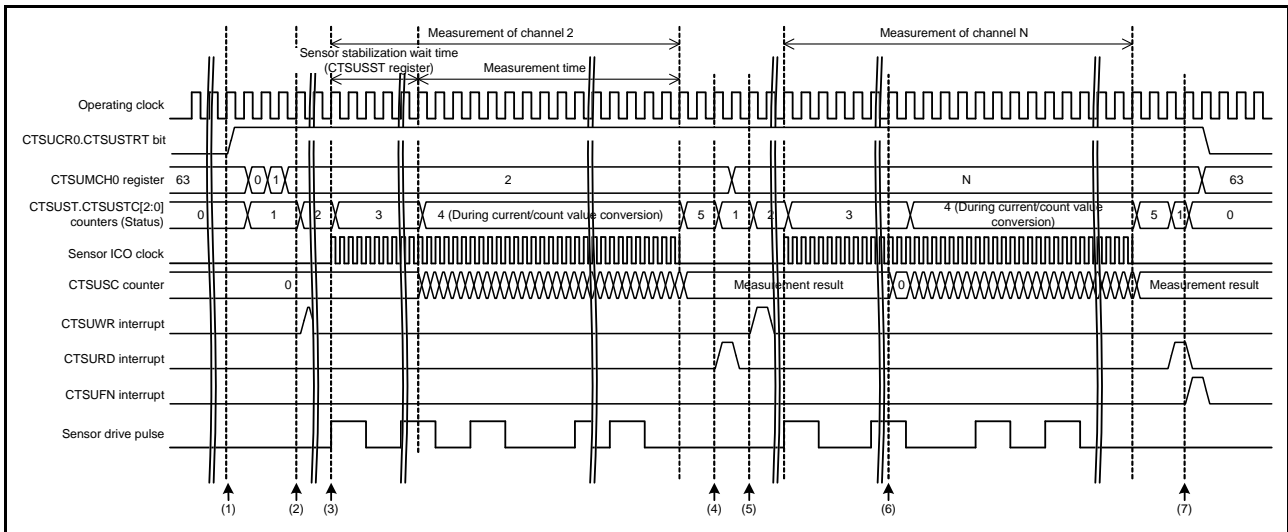


Figure 34.14 Software Flow and Operation Example of Self-Capacitance Multi-Scan Mode



**Figure 34.15 Timing Chart of Self-Capacitance Multi-Scan Mode (Measurement Start Condition is Software Trigger)**

The following describes operation shown in the timing chart in Figure 34.15.

- (1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.
- (2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.
- (3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.
- (4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is stopped, a measurement result read request (CTSURD) is output.
- (5) After a channel to be measured next is determined, a measurement channel setting request (CTSUWR) is output.
- (6) After the stabilization wait time has elapsed and when the previous measurement is read, the result is cleared and measurement is started.
- (7) Upon completion of all measurement channels, a measurement end interrupt (CTSUFN) is output and measurement is stopped (transition to Status 0).

Table 34.7 lists the touch pin states in self-capacitance multi-scan mode.

**Table 34.7 Touch Pin States in Self-Capacitance Multi-Scan Mode**

Status	Touch Pin		Remarks
	Measurement Channel	Non-Measurement Channel	
0	Low	Low	Normal I/O is driven low
1	Low	Low	Normal I/O is driven low
2	Low	Low	Normal I/O is driven low The pin changes to the pin state in Status 3 (measurement channel touch I/O starts being driven) after data is written to the CTSUSO1 register, and the status transitions to Status 3 after one base clock cycle.
3	Pulse	Low	Measurement channel: Touch I/O is pulse, normal I/O is Hi-Z Non-measurement channel: Normal I/O is driven low
4	Pulse	Low	Measurement channel: Touch I/O is pulse, normal I/O is Hi-Z Non-measurement channel: Normal I/O is driven low
5	Low	Low	Normal I/O is driven low The pin state in Status 4 pin changes to this state (all normal I/O being driven low) after one operating clock cycle.



### 34.3.2.5 Mutual Capacitance Simple Scan Mode Operation

In mutual capacitance simple scan mode, measurement is performed by switching the impedance of the transmit electrode, in comparison with mutual capacitance full scan mode.

Electrostatic capacitance is measured sequentially on channels set to transmission or reception specified by the CTSUCHTRC0 and CTSUCHTRC1 registers, and measurement targets specified by the CTSUCHAC0 and CTSUCHAC1 registers. Electrostatic capacitance is measured by combining signals from the measurement target pins that are allocated to transmission or reception. Figure 34.16 shows the software flowchart and an operation example, and Figure 34.17 shows the timing chart.

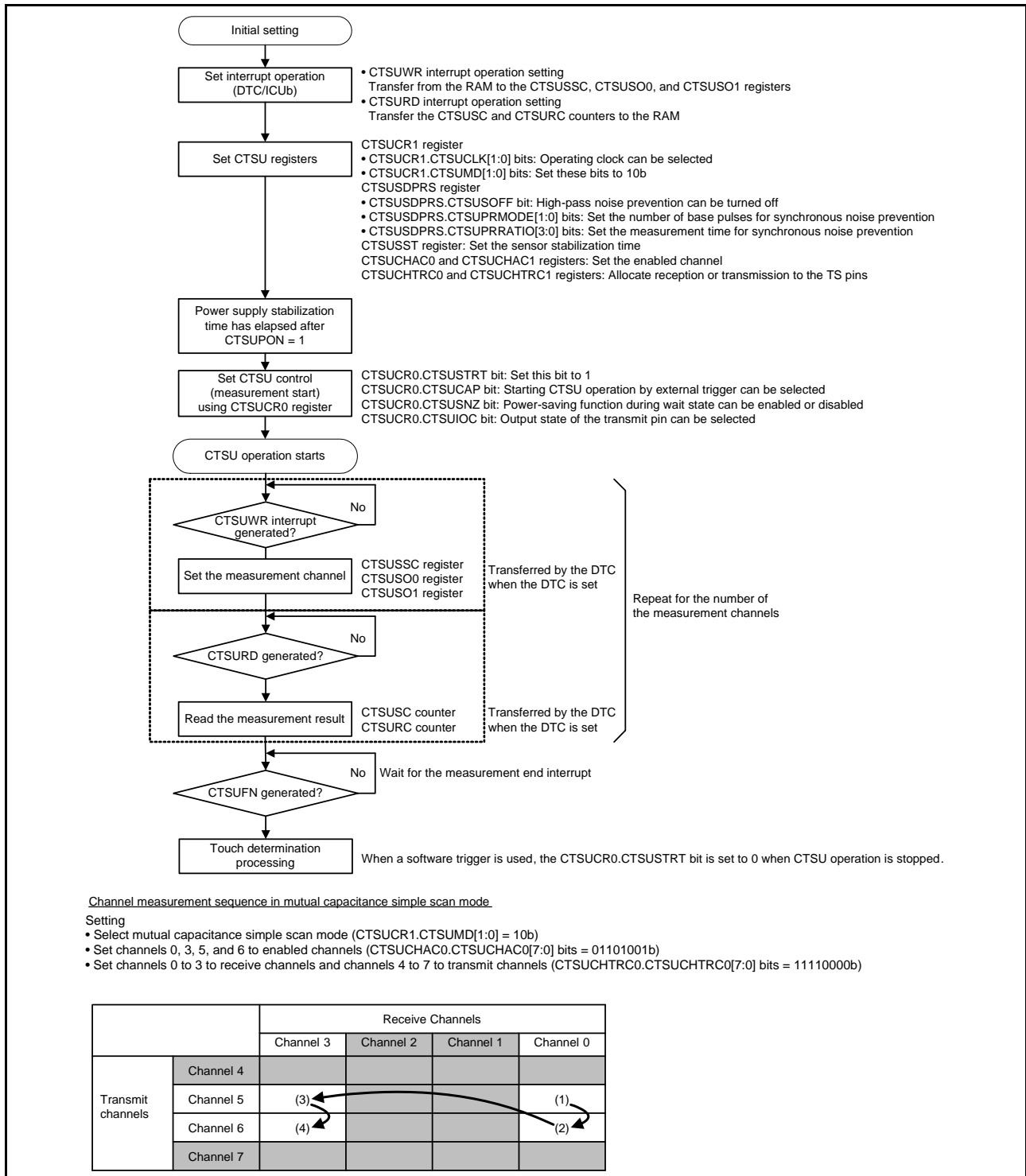
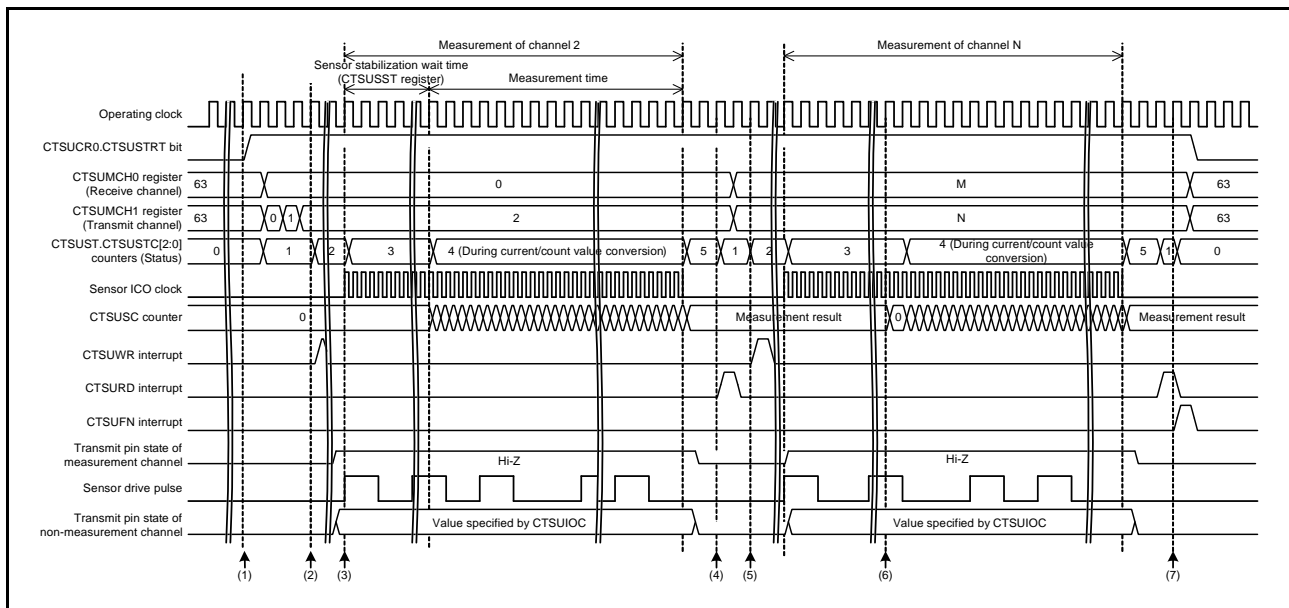


Figure 34.16 Software Flowchart and Operation Example of Mutual Capacitance Simple Scan Mode



**Figure 34.17 Timing Chart of Mutual Capacitance Simple Scan Mode (Measurement Start Condition is Software Trigger)**

The following describes operation shown in the timing chart in Figure 34.17.

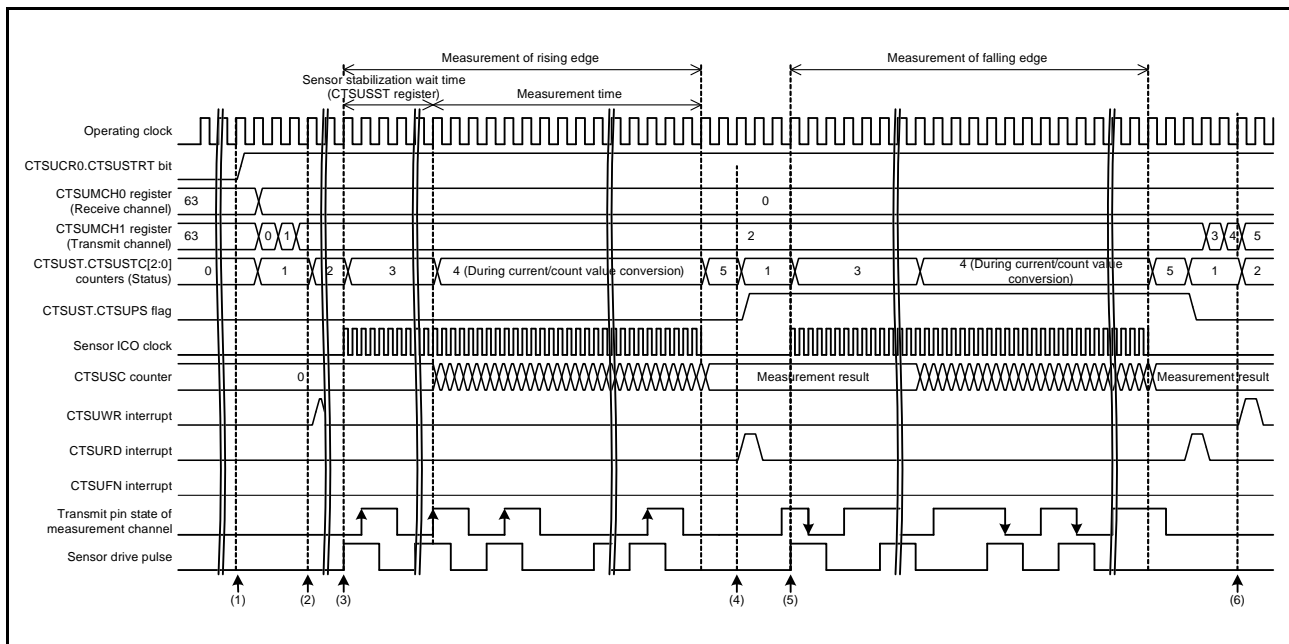
- (1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.
- (2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.
- (3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the transmit pin of the measurement channel changes to Hi-Z and the value set in the CTSUCR0.CTSUIOC bit is output from the transmit pin of the non-measurement channel. After a transition to Status 3, the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.
- (4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is stopped, a measurement result read request (CTSURD) is output.
- (5) After a channel to be measured next is determined, a measurement channel setting request (CTSUWR) is output.
- (6) After the stabilization wait time has elapsed and when the previous measurement is read, the result is cleared and measurement is started.
- (7) Upon completion of all measurement channels, a measurement end interrupt (CTSUFN) is output and measurement is stopped (transition to Status 0).

Table 34.8 lists the touch pin states in mutual capacitance simple scan mode.

**Table 34.8 Touch Pin States in Mutual Capacitance Simple Scan Mode**

Status	Touch Pin of Receive Channel		Touch Pin of Transmit Channel		Remarks
	Measurement Channel	Non-Measurement Channel	Measurement Channel	Non-Measurement Channel	
0	Low	Low	Low	Low	Normal I/O is driven low
1	Low	Low	Low	Low	Normal I/O is driven low
2	Low	Low	Low	Low	Normal I/O is driven low The pin changes to the pin state in Status 3 (measurement receive channel touch I/O starts being driven, measurement transmit channel is Hi-Z, non-measurement transmit channel is IOC value output) after data is written to the CTSUSO1 register, and the status transitions to Status 3 after one base clock cycle.
3	Pulse	Low	Hi-Z	CTSUIOC	Receive measurement channel: Touch I/O is pulse, normal I/O is Hi-Z Receive non-measurement channel: Normal I/O is driven low Transmit measurement channel: Hi-Z Transmit non-measurement channel: Normal I/O being driven according to the CTSUCR0.CTSUIOC bit value
4	Pulse	Low	Hi-Z	CTSUIOC	Receive measurement channel: Touch I/O is pulse, normal I/O is Hi-Z Receive non-measurement channel: Normal I/O is driven low Transmit measurement channel: Hi-Z Transmit non-measurement channel: Normal I/O is driven according to the CTSUCR0.CTSUIOC bit value
5	Low	Low	Low	Low	Normal I/O is driven low The pin state in Status 4 changes to this state (all normal I/O being driven low) after one operating clock cycle.





**Figure 34.19 Timing Chart of Mutual Capacitance Full Scan Mode (Measurement Start Condition is Software Trigger)**

The following describes operation shown in the timing chart in Figure 34.19.

- (1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.
- (2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.
- (3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate. At the same time, a pulse which is handled as the rising edge is output to the transmit pin on the measurement channel during the high-level period of the sensor drive pulse.
- (4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is stopped, a measurement result read request (CTSURD) is output.
- (5) The same channel is measured by outputting a pulse that is handled as the falling edge during the high-level period of the sensor drive pulse.
- (6) After the same channel is measured twice, a channel to be measured next is determined and measured in the similar way.
- (7) Upon completion of all measurement channels, a measurement end interrupt (CTSUFN) is output and measurement is stopped (transition to Status 0).

The mutual capacitance measurement status flag (CTSUST.CTSUPS bit) is changed at the timing at which Status 5 transitions to Status 1.

Table 34.9 lists the touch pin states in mutual capacitance full scan mode.

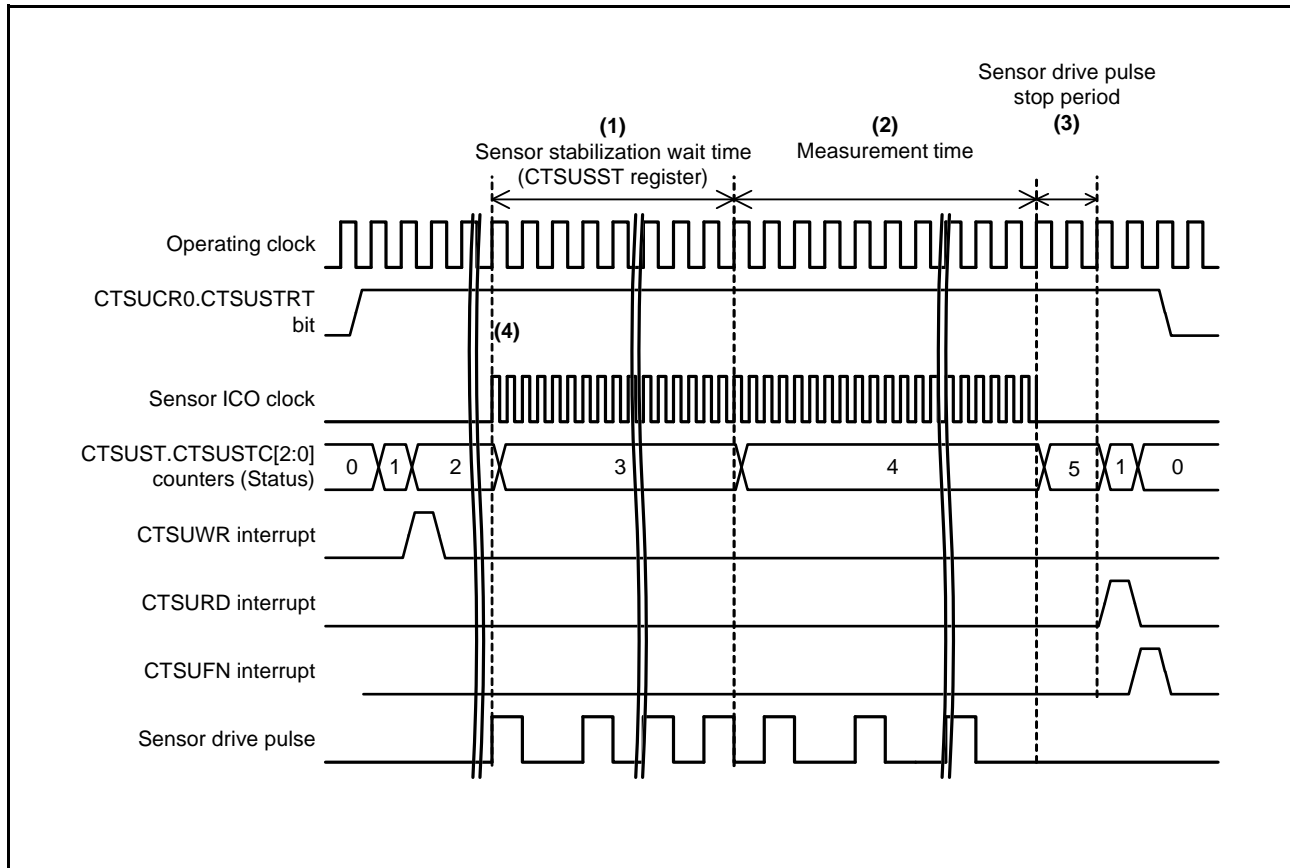
**Table 34.9 Touch Pin States in Mutual Capacitance Full Scan Mode**

Status	Touch Pin of Receive Channel		Touch Pin of Transmit Channel		Remarks
	Measurement Channel	Non-Measurement Channel	Measurement Channel	Non-Measurement Channel	
0	Low	Low	Low	Low	Normal I/O is driven low
1	Low	Low	Low/High	Low	Normal I/O is driven low However, at the second measurement, the pin changes to the pin state in Status 3 (measurement receive channel touch I/O starts being driven, measurement transmit channel is driven high) after two base clock cycles, and the status transitions to Status 3 after one base clock cycle.
2	Low	Low	Low	Low	Normal I/O is driven low The pin changes to the pin state in Status 3 (measurement receive channel touch I/O starts being driven) after data is written to the CTSUSO1 register, and the status transitions to Status 3 after one base clock cycle.
3	Pulse	Low	Pulse	Low	Receive measurement channel: Touch I/O is pulse, normal I/O is Hi-Z Receive non-measurement channel: Normal I/O is driven low Transmit measurement channel: Normal I/O is driven at the rising edge of the pulse at the first measurement Normal I/O is driven at the falling edge of the pulse at the second measurement Transmit non-measurement channel: Normal I/O is driven low
4	Pulse	Low	Pulse	Low	Same as above
5	Low	Low	Low	Low	Normal I/O is driven low The pin state in Status 4 changes to this state (all normal I/O being driven low) after one operating clock cycle.

### 34.3.3 Items Common to Multiple Modes

#### 34.3.3.1 Sensor Stabilization Wait Time and Measurement Time

Figure 34.20 shows the timing chart of the sensor stabilization wait time and measurement time.



**Figure 34.20 Sensor Stabilization Wait Time and Measurement Time**

- (1) In response to the CTSUWR interrupt request, output of the sensor drive pulse is started by write access to the CTSUSO1 register. Then, wait for the stabilization time set in the CTSUSST register.
- (2) When the sensor stabilization time has elapsed and the CTSUST.CTSUDTSR flag is set to 0, measurement is started at transition to Status 4. The measurement time is determined by setting the base clock cycle and the CTSUSDPRS.CTSUPRMODE[1:0], CTSUPRRATIO[3:0], and CTSUSO0.CTSUSNUM[5:0] bits. When the measurement time has elapsed, measurement of the corresponding channel is stopped.
- (3) After the measurement time has elapsed, the status transitions to Status 1 after two operating clock cycles and a CTSURD interrupt is generated, so read the data from the CTSUSC and CTSURC counters.  
At this time, the sensor drive pulse is output at the low level. When measurement of all specified channels is completed, the CTSUCR0.CTSUSTRT bit is set to 0.
- (4) The sensor ICO clock oscillates while the CTSUnSTC[2:0] counters = 011b (Status 3) or 100b (Status 4).

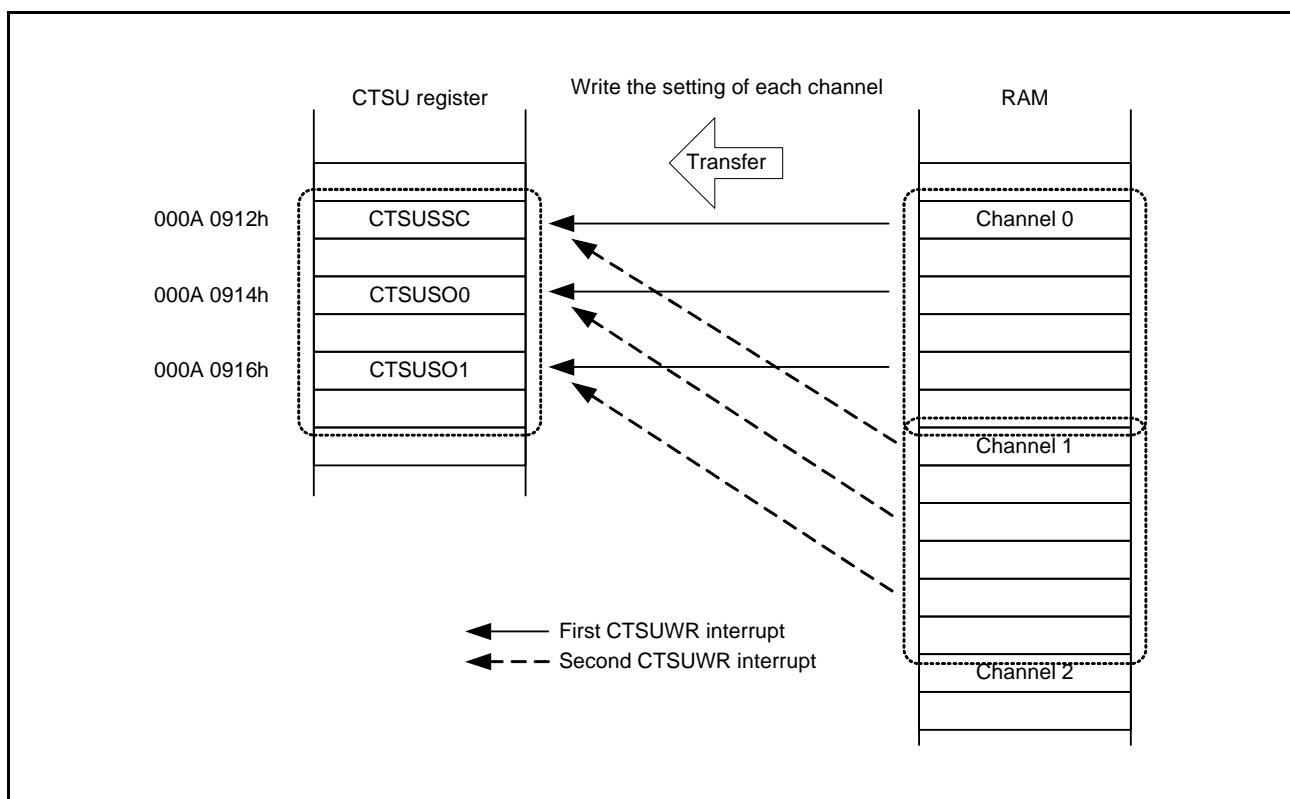
### 34.3.3.2 Interrupts

There are three types of interrupts for the CTSU.

- Write request interrupt of setting registers for each channel (CTSUWR)
- Measurement data transfer request interrupt (CTSURD)
- Measurement end interrupt (CTSUFN)

#### (1) Write request interrupt of setting registers for each channel (CTSUWR)

Store the setting data for each measurement channel in the RAM, and set the DTC or ICub transfer corresponding to the CTSUWR interrupt in advance. The CTSUWR interrupt is output at the timing at which Status 1 transitions to Status 2. Write the setting data of the corresponding channel from the RAM to the CTSUSSC, CTSUSO0, and CTSUSO1 registers (Figure 34.21). Since write access to the CTSUSO1 register controls a transition to the next status, be sure to set this register last.



**Figure 34.21 Example of DTC Transfer Operation Using CTSUWR Interrupt**

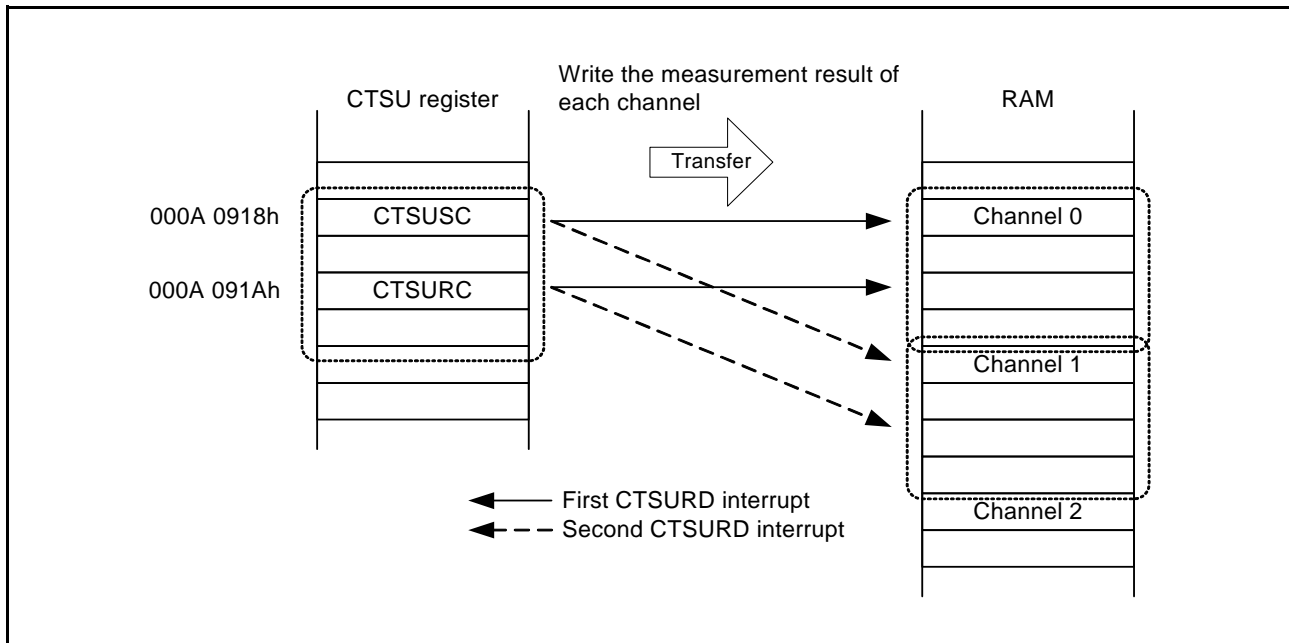
The registers (CTSUSSC, CTSUSO0, and CTSUSO1) to be set are allocated at sequential addresses. Set the operation at interrupt generation as shown below.

- Transfer destination address: Address of the CTSUSSC register
- Handling at the transfer destination address: Transfer 2-byte data three times by a single interrupt. (The address of the start byte is fixed.)
- Transfer source address: CTSUSO0 register data storage address for the minimum channel in the setting data stored in the RAM
- Handling at the transfer source address: Transfer 2-byte data three times by a single interrupt. (The address of the first byte is continued from the previous interrupt handling.)
- Number of transfers by an interrupt: Specify the number of measurements.



## (2) Measurement data transfer request interrupt (CTSURD)

Set DTC or ICUB transfer corresponding to the CTSURD interrupt in advance. The CTSURD interrupt is output at the timing at which Status 5 transitions to Status 1. Read the measurement result from the CTSUSC and CTSURC counters (Figure 34.22).



**Figure 34.22 Example of DTC Transfer Operation Using CTSURD Interrupt**

The measurement result registers (CTSUSC and CTSURC counters) used as transfer sources are allocated at sequential addresses. Set the operation at interrupt generation as shown below.

- Transfer source address: Address of the CTSUSC counter
- Handling at the transfer source address: Transfer 2-byte data twice by a single interrupt. (The start address is fixed.)
- Transfer destination address: CTSUSC counter data storage address for the minimum channel in the setting data stored in the RAM.
- Handling at the transfer destination address: Transfer 2-byte data twice by a single interrupt. (The start address is continued from the previous interrupt handling.)
- Number of transfers by an interrupt: Specify the number of measurements.

## (3) Measurement end interrupt (CTSUFN)

When all channels are measured, an interrupt is generated at the timing at which Status 1 transitions to Status 0. Use software to confirm the overflow flags (CTSUST.CTSUSOVF and CTSUROVF flags) and read the measurement results to determine whether or not the electrode is touched.

Interrupt requests are accepted or disabled in the interrupt control block.

## 34.4 Usage Notes

### 34.4.1 Measurement Result Data (CTSUSC and CTSURC Counters)

Read access during measurement is prohibited. If the measurement result data is accessed, an incorrect value may be read due to asynchronous operation.

### 34.4.2 Software Trigger

When 10b (PCLK/4) is selected by the CTSUCR1.CTSUCLK[1:0] bits, to restart measurement by writing 1 to the CTSUCR0.CTSUSTRT bit after measurement has been completed, wait for at least three cycles to elapse after an interrupt is generated, and then write to the CTSUCR0.CTSUSTRT bit.

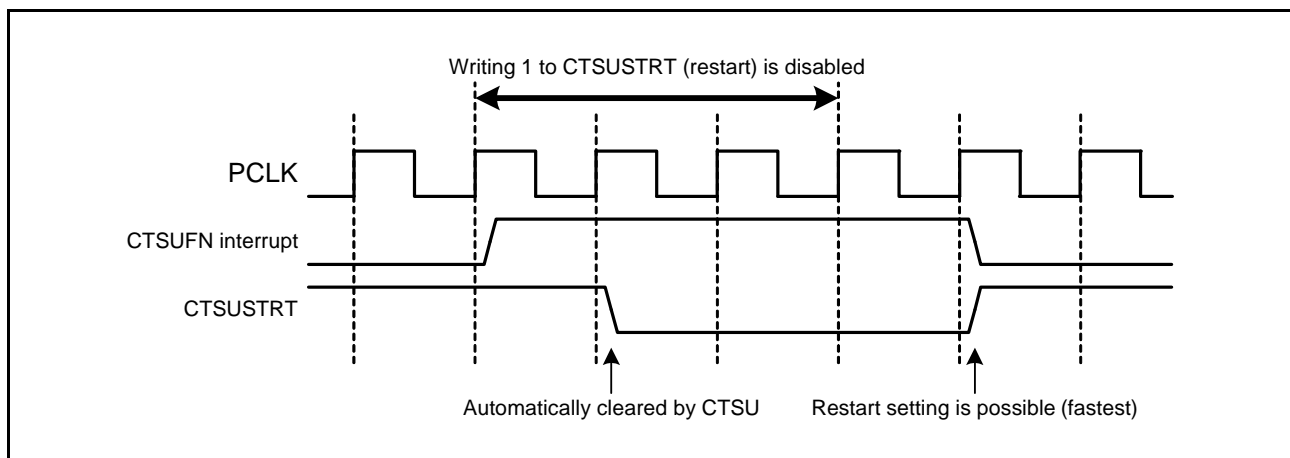


Figure 34.23 Notes on Restarting Measurement

### 34.4.3 External Trigger

- If an external trigger is input during the measurement time, measurement is not started. The next external event is enabled after one cycle of the operating clock when a CTSUFN interrupt is generated.
- To stop external trigger mode, write 0 to the CTSUCR0.CTSUSTRT bit and 0 to the CTSUCR0.CTSUINIT bit at the same time (forced stop).

### 34.4.4 Notes on Forcibly Stopping Operation

To forcibly stop the current operation, write 0 to the CTSUCR0.CTSUSTRT bit and 1 to the CTSUCR0.CTSUINIT bit at the same time. After this setting, the operation is stopped and the internal control registers are initialized.

When the CTSUCR0.CTSUINIT bit is used for initialization, the following registers are initialized in addition to the initialization of the internal measurement state.

- CTSUMCH0 register
- CTSUMCH1 register
- CTSUST register
- CTSUSC counter
- CTSURC counter

If operation is forcibly stopped, an interrupt request may be generated depending on the internal state. After operation is forcibly stopped, perform the processing for stopping/disabling the DTC or ICUb.

If DTC transfer is stopped in the mounted system for some reason, also perform the processing for forcibly stopping and initializing the CTSU.

#### 34.4.5 TSCAP Pin

The capacitor connected to the TSCAP pin should be fully discharged using I/O port control to output a low level, before turning on the switch (CTSUCR1.CTSUCSW bit = 1) to establish a connection.

#### 34.4.6 Notes during Measurement Operation (CTSUCR0.CTSUSTRT Bit = 1)

During measurement operation (CTSUCR0.CTSUSTRT bit = 1), do not make settings such as “stop the peripheral clock” or “change the port settings related to the touch pins (TS and TSCAP pins)” in the higher system.

If control settings non-compliant to these restrictions are made, after operation is forcibly stopped (CTSUCR0.CTSUSTRT bit = 0 and CTSUCR0.CTSUINIT bit = 1), write 0 to the CTSUCR1.CTSUPON bit and 0 to the CTSUCR1.CTSUCSW bit at the same time, and set the CTSUCR0.CTSUSNZ bit to 0. Then, restart from the initial setting flowchart shown in Figure 34.9.

## 35. 12-Bit A/D Converter (S12ADb)

In this section, “PCLK” is used to refer to PCLKB.

### 35.1 Overview

This MCU includes a 12-bit successive approximation A/D converter. Up to 17 channel analog inputs, temperature sensor outputs, or internal reference voltages can be selected.

The 12-bit A/D converter converts a maximum of 17 selected channels of analog inputs, temperature sensor outputs, or internal reference voltages into a 12-bit digital value through successive approximation.

The A/D converter has three operating modes: single scan mode in which the analog inputs of up to 17 arbitrarily selected channels are converted for only once in ascending channel order; and continuous scan mode in which the analog inputs of up to 17 arbitrarily selected channels are continuously converted in ascending channel order; and group scan mode in which up to 17 channels of the analog inputs are arbitrarily divided into two groups (group A and group B) and converted in ascending channel order in each group.

In group scan mode, the scan start conditions of group A and group B can be independently selected, thus allowing A/D conversion of group A and group B to be started independently.

In double trigger mode, one arbitrarily selected analog input channel is converted in single scan mode or group scan mode (group A), and the resulting data of A/D conversion started by the first and second triggers are stored into separate registers (duplication of A/D conversion data).

A/D conversion of the temperature sensor output or the internal reference voltage is accomplished independently.

The VREFH0 or AVCC0 pin can be selected as the reference power supply pin for high-electric potential. The VREFL0 or AVSS0 pin can be selected as the reference power supply ground pin for low-electric potential.

Table 35.1 lists the specifications of the 12-bit A/D converter and Table 35.2 lists the functions of the 12-bit A/D converter. Figure 35.1 shows a block diagram of the 12-bit A/D converter.

**Table 35.1 Specifications of 12-Bit A/D Converter (1/2)**

Item	Specifications
Number of units	One unit
Input channels	Up to 17 channels
Extended analog inputs	Temperature sensor output, internal reference voltage
A/D conversion method	Successive approximation
Resolution	12 bits
Conversion time	1.0 $\mu$ s per channel (when A/D conversion clock ADCLK = 32 MHz)
A/D conversion clock	Peripheral module clock PCLK*1 and A/D conversion clock ADCLK*1 can be set so that the frequency division ratio should be one of the following. PCLK to ADCLK frequency division ratio = 1:1, 1:2, 1:4, 1:8, 2:1, 4:1 ADCLK is set using the clock generation circuit.
Data registers	For analog input: 17 data registers For duplication of A/D conversion data in double trigger mode: One data register For temperature sensor: One data register For internal reference voltage: One data register The A/D conversion result is stored in 12-bit A/D data registers. In addition mode, A/D conversion results are added and stored in A/D data registers as 14-bit data. Duplication of A/D conversion data <ul style="list-style-type: none"> <li>• A/D conversion data of one selected analog input channel is stored into A/D data register y when conversion is started by the first trigger and into the duplication register when started by the second trigger.</li> <li>• Duplication is available only in double trigger mode in single scan mode or group scan mode.</li> </ul>

**Table 35.1 Specifications of 12-Bit A/D Converter (2/2)**

Item	Specifications
Operating modes	<ul style="list-style-type: none"> <li>• Single scan mode: A/D conversion is performed for only once on the analog inputs of up to 17 arbitrarily selected channels. A/D conversion is performed only once on the temperature sensor output. A/D conversion is performed only once on the internal reference voltage.</li> <li>• Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs of up to 17 arbitrarily selected channels.*<sup>2</sup></li> <li>• Group scan mode: Up to 17 channels of analog inputs are divided into group A and group B and A/D conversion is performed only once on all the selected channels on a group basis. The scan start conditions of group A and group B can be independently selected, thus allowing A/D conversion of group A and group B to be started independently.</li> </ul>
Conditions of A/D conversion start	<ul style="list-style-type: none"> <li>• Software trigger</li> <li>• Synchronous trigger Trigger by MTU or ELC</li> <li>• Asynchronous trigger A/D conversion can be triggered from the ADTRG0# pin.</li> </ul>
Functions	<ul style="list-style-type: none"> <li>• Variable sampling state count</li> <li>• A/D-converted value addition mode</li> <li>• Double trigger mode (duplication of A/D conversion data)</li> </ul>
Interrupt sources	<ul style="list-style-type: none"> <li>• In the modes except double trigger mode and group scan mode, A/D scan end interrupt (S12ADI0) request can be generated on completion of single scan.</li> <li>• In double trigger mode, A/D scan end interrupt (S12ADI0) request can be generated on completion of double scan.</li> <li>• In group scan mode, A/D scan end interrupt (S12ADI0) request can be generated on completion of group A scan, whereas A/D scan end interrupt specially for group B (GBADI) request can be generated on completion of group B scan.</li> <li>• In group scan mode with double trigger mode, A/D scan end interrupt (S12ADI0) request can be generated on completion of double scan of group A, whereas A/D scan end interrupt specially for group B (GBADI) request can be generated on completion of group B scan.</li> <li>• S12ADI0 or GBADI interrupt can activate data transfer controller (DTC).</li> </ul>
Event linking	<ul style="list-style-type: none"> <li>• An ELC event can be generated on completion of scans except for group B scan in group scan mode.</li> <li>• A/D conversion can be started by the trigger from ELC.</li> </ul>
Reference voltage	<ul style="list-style-type: none"> <li>• VREFH0, AVCC0, or the internal reference voltage can be selected as the high-side reference voltage.</li> <li>• VREFL0 or AVSS0 can be selected as the low-side reference voltage.</li> </ul>
Low power consumption function	<ul style="list-style-type: none"> <li>• Module stop state can be specified.*<sup>3</sup></li> </ul>

Note 1. Peripheral module clock PCLK is set according to the setting of the SCKCR.PCKB[3:0] bits and A/D conversion clock ADCLK is set according to the setting of the SCKCR.PCKB[3:0] bits.

Note 2. Do not use continuous scan mode or group scan mode when temperature sensor output or internal reference voltage is selected.

Note 3. When the module stop state is canceled, A/D conversion can be started after 1  $\mu$ s has elapsed.

**Table 35.2 Functions of 12-Bit A/D Converter**

Item		Function	
Analog input channels		AN000 to AN015, AN021, temperature sensor output, internal reference voltage	
A/D conversion start conditions	Software	Software trigger	
	Asynchronous trigger	ADTRG0#	
	Synchronous trigger	TRGA compare match/input capture from MTU0	TRG0AN
		TRGB compare match/input capture from MTU0	TRG0BN
		TRGA compare match/input capture or MTU4.TCNT underflow (trough) in complementary PWM mode from MTU0 to MTU4	TRGAN
		TRGE compare match from MTU0	TRG0EN
		TRGF compare match from MTU0	TRG0FN
		MTU4.TADCORA and MTU4.TCNT compare match (interrupt skipping function 1)	TRG4AN
		MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1)	TRG4BN
		MTU4.TADCORA and MTU4.TCNT compare match and MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1)	TRG4ABN
Trigger from ELC	Enabled		
Interrupt		S12ADIO interrupt, GBADI interrupt	
Module stop function setting*1		MSTPCRA.MSTPA17 bit	

Note 1. For details, refer to section 11, Low Power Consumption.

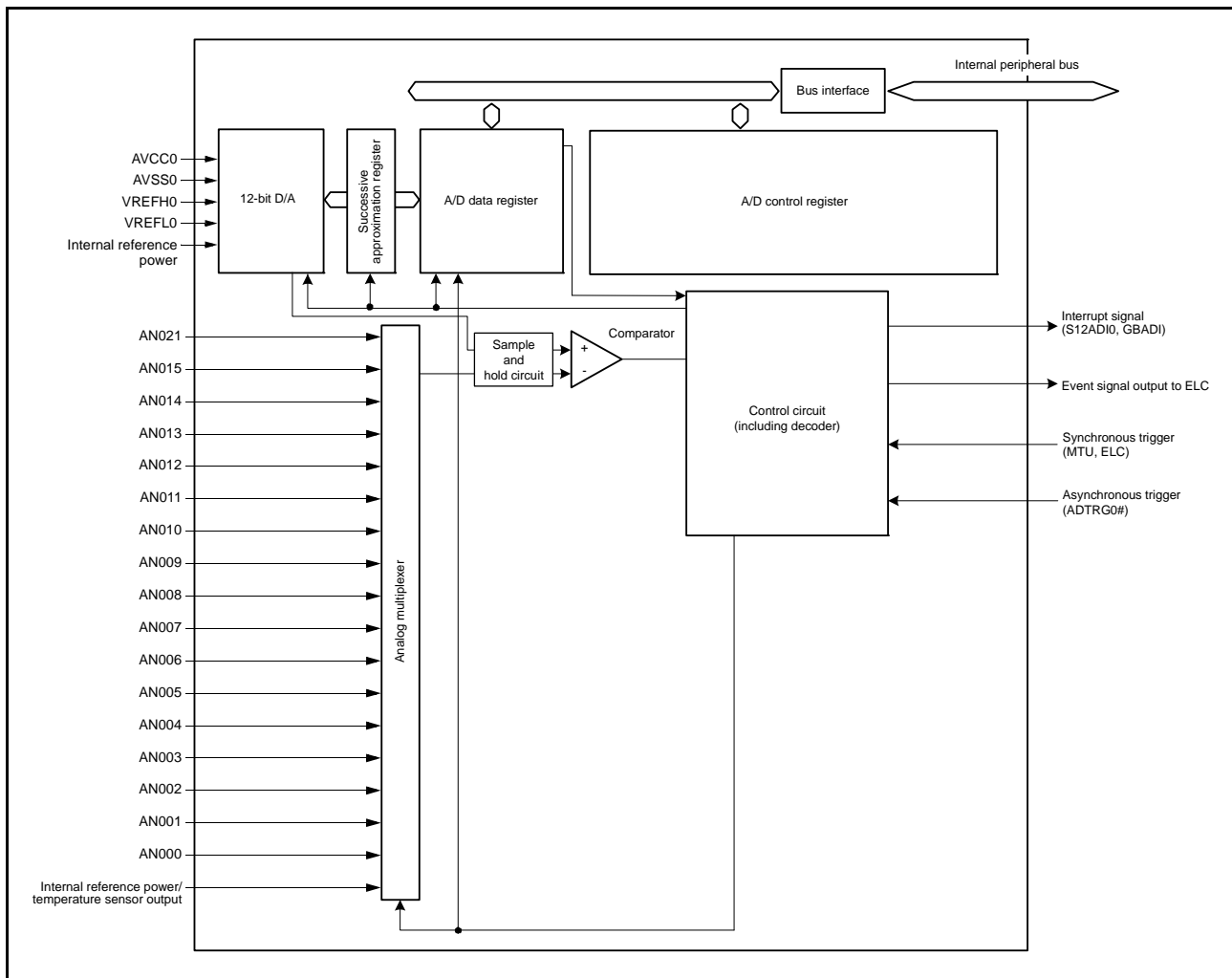


Figure 35.1 Block Diagram of 12-Bit A/D Converter

Table 35.3 lists the input pins of the 12-bit A/D converter.

Table 35.3 Input Pins of 12-Bit A/D Converter

Pin Name	Input	Function
AVCC0	Input	Analog block power supply pin
AVSS0	Input	Analog block ground pin
VREFH0	Input	Reference power supply pin
VREFL0	Input	Reference ground pin
AN000 to AN015, AN021	Input	Analog input pins
ADTRG0#	Input	External trigger input pin for starting A/D conversion

## 35.2 Register Descriptions

### 35.2.1 A/D Data Registers y (ADDRy) (y = 0 to 15, 21)

ADDRy are 16-bit read-only registers which store the A/D conversion results of channels AN000 to AN015, AN021. 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 right-alignment)

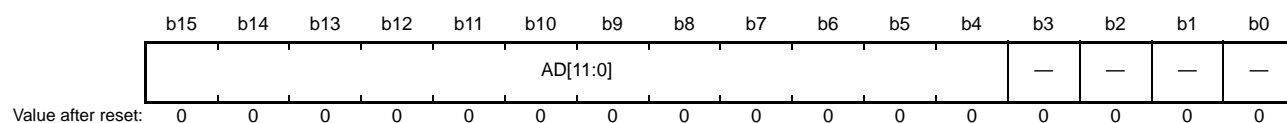
Address(es): ADDR0: 0008 9020h to ADDR15: 0008 903Eh, ADDR21: 0008 904Ah



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 left-alignment)

Address(es): ADDR0: 0008 9020h to ADDR15: 0008 903Eh, ADDR21: 0008 904Ah



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): ADDR0: 0008 9020h to ADDR15: 0008 903Eh, ADDR21: 0008 904Ah



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 left-aligned.

The following minimum and maximum values apply to channels on which A/D-converted value addition mode is selected.

First conversion:  $0000h \leq ADDR_y (y = 0 \text{ to } 15, 21) \leq 3FFCh$   
(ADDR<sub>y</sub> (y = 0 to 15, 21): Bits 15 and 14 = 00b, bits 13 to 2 = AD11 to AD0, bits 1 and 0 = 00b)

Second conversion:  $0000h \leq ADDR_y (y = 0 \text{ to } 15, 21) \leq 7FF8h$   
(ADDR<sub>y</sub> (y = 0 to 15, 21): Bit 15 = 0b, bits 14 to 2 = AD12 to AD0, bits 1 and 0 = 00b)

Third conversion:  $0000h \leq ADDR_y (y = 0 \text{ to } 15, 21) \leq BFF4h$   
(ADDR<sub>y</sub> (y = 0 to 15, 21): Bits 15 to 2 = AD13 to AD0, bits 1 and 0 = 00b)

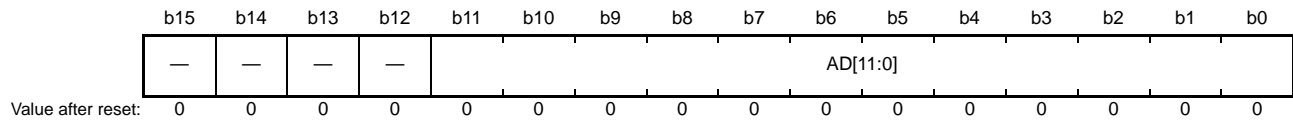
Fourth conversion:  $0000h \leq ADDR_y (y = 0 \text{ to } 15, 21) \leq FFF0h$   
(ADDR<sub>y</sub> (y = 0 to 15, 21): Bits 15 to 2 = AD13 to AD0, bits 1 and 0 = 00b)

### 35.2.2 A/D Data Duplication Register (ADDBLDR)

ADDBLDR is a 16-bit read-only register used in double trigger mode. ADDBLDR holds the results of A/D conversion of the analog input of the channel selected for data duplication when the conversion is started by the second trigger. ADDBLDR uses 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 right-alignment)

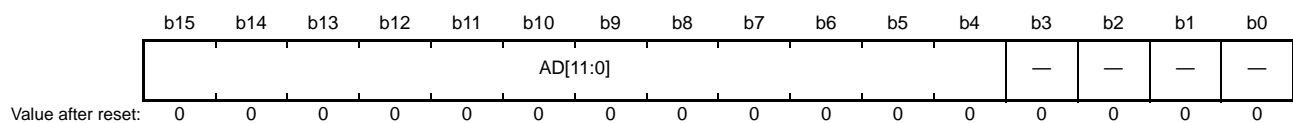
Address(es): 0008 9018h



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 left-alignment)

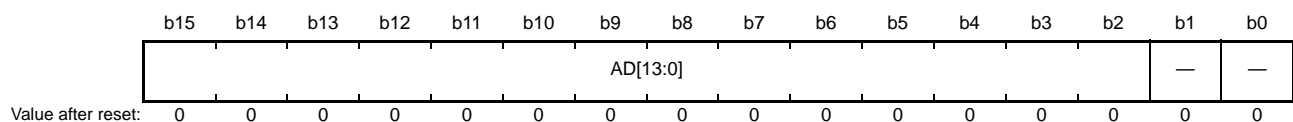
Address(es): 0008 9018h



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 9018h



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

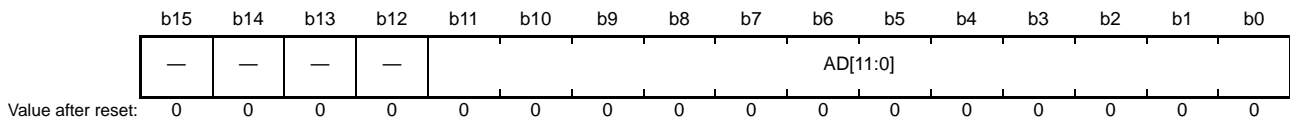
When A/D-converted value addition mode is selected, the ADDBLDR.AD[13:0] bits show the value added by the A/D-converted value of the respective channels. In A/D-converted value addition mode, the setting of the ADCER.ADRFMT bit becomes invalid and the format of the register becomes left-aligned.

### 35.2.3 A/D Temperature Sensor Data Register (ADTSDR)

ADTSDR is a 16-bit read-only register that holds the A/D conversion results of the temperature sensor output. The following different formats are used depending on the settings of the A/D data register format select bit (ADCER.ADRFMT) and A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for right-alignment)

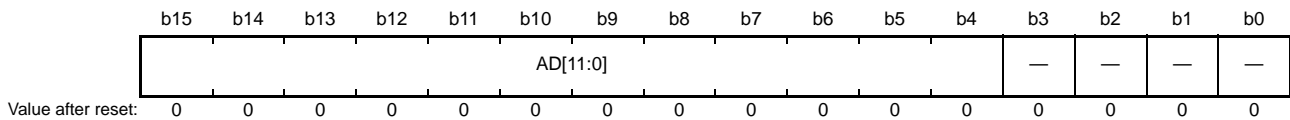
Address(es): 0008 901Ah



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 left-alignment)

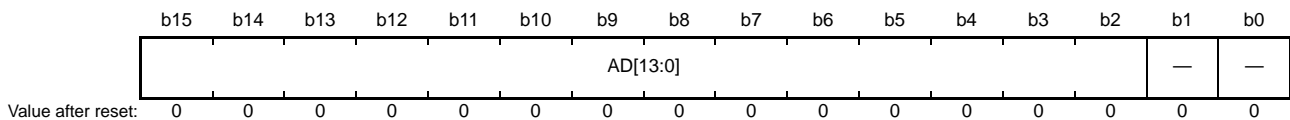
Address(es): 0008 901Ah



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 function 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]	—	A/D-converted value addition result of temperature sensor output	R

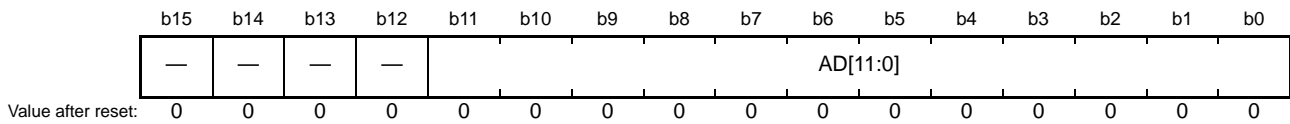
When A/D-converted value addition mode is selected, the ADDBLDR.AD[13:0] bits show the temperature sensor output value added by the A/D-converted value. In A/D-converted value addition mode, the setting of the ADCER.ADRFMT bit becomes invalid and the format of the register becomes left-aligned.

### 35.2.4 A/D Internal Reference Voltage Data Register (ADOCDR)

ADOCDR is a 16-bit read-only register that holds the A/D conversion results of the 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 right-alignment)

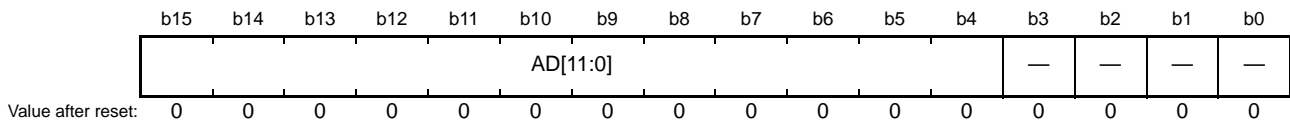
Address(es): 0008 901Ch



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 left-alignment)

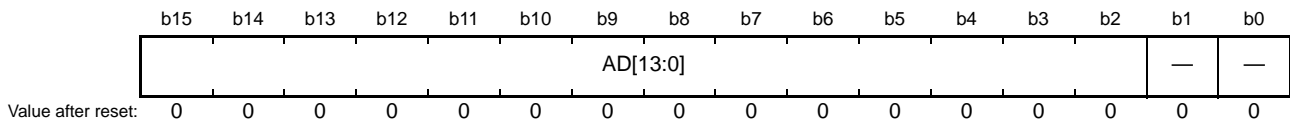
Address(es): 0008 901Ch



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 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 left-aligned.

### 35.2.5 A/D Control Register (ADCSR)

Address(es): 0008 9000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADST	ADCS[1:0]	ADIE	—	ADHSC	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DBLANS[4:0]	A/D Conversion Data Duplication Channel Select	Select one of 17 analog input channels for A/D conversion data duplication. These bits are valid only in double trigger mode.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	GBADIE	Group B Scan End Interrupt Enable	0: Disables GBADI interrupt generation upon group B scan completion. 1: Enables GBADI interrupt generation upon group B scan completion.	R/W
b7	DBLE	Double Trigger Mode Select	0: Deselects double trigger mode. 1: Selects double trigger mode.	R/W
b8	EXTRG	Trigger Select <sup>1</sup>	0: A/D conversion is started by the synchronous trigger (MTU or ELC). 1: A/D conversion is started by the asynchronous trigger (ADTRG0#).	R/W
b9	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
b10	ADHSC	A/D Conversion Mode Select	0: Normal conversion 1: High-speed conversion	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	ADIE	Scan End Interrupt Enable	0: Disables S12ADI0 interrupt generation upon scan completion. 1: Enables S12ADI0 interrupt generation upon scan completion.	R/W
b14, b13	ADCS[1:0]	Scan Mode Select	b14 b13 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
b15	ADST	A/D Conversion Start	0: Stops A/D conversion. 1: Starts A/D conversion.	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 clock cycles.

#### DBLANS[4:0] Bits (A/D Conversion Data Duplication Channel Select)

The DBLANS[4:0] bits select one of the channels for A/D conversion data duplication in double trigger mode. The A/D conversion results of the analog input of the selected channel are stored into A/D data register y when conversion is started by the first trigger, and into the A/D data duplication register when started by the second trigger. Table 35.4 shows the relationship between the DBLANS[4:0] bit settings and selected duplication channel. A/D-converted value addition mode with double trigger mode can be set by selecting the channel selected by the DBLANS[4:0] bits using the ADADS register. If double trigger mode is selected, the channel selected by the ADANSA and ADANSA1 registers are invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead. When converting analog inputs of channels, temperature sensor output and internal reference voltage should not be selected for A/D conversion.

The DBLANS[4:0] bits should be set while the ADST bit is 0 (they should not be set simultaneously when 1 is written to the ADST bit.)

**Table 35.4 Relationship between DBLANS[4:0] Bit Settings and Double Trigger Enabled Channels**

DBLANS[4:0]	Duplication Channel	DBLANS[4:0]	Duplication Channel
00000	AN000	01001	AN009
00001	AN001	01010	AN010
00010	AN002	01011	AN011
00011	AN003	01100	AN012
00100	AN004	01101	AN013
00101	AN005	01110	AN014
00110	AN006	01111	AN015
00111	AN007	10101	AN021
01000	AN008		

**GBADIE Bit (Group B Scan End Interrupt Enable)**

The GBADIE bit enables or disables group B scan end interrupt (GBADI) in group scan mode.

**DBLE Bit (Double Trigger Mode Select)**

In double trigger mode, the following operation is performed after scanning is started by the MTU or ELC trigger selected by the ADSTRGR.TRSA[3:0] bits.

1. When the ADIE bit is 1, a scan end interrupt is not output upon first scan completion but is output upon second scan completion.
2. The A/D conversion results of the analog input of the channel selected by the DBLANS[4:0] bits are stored into A/D data register y for the first time, and into the A/D data duplication register for the second time.

Setting the DBLE bit to 1 invalidates the channel selected by the ADANSA and ADANSA1 registers. In continuous scan mode, double trigger mode should not be selected. Temperature sensor output and internal reference voltage should not be selected for A/D conversion. In double trigger mode, software trigger should not be selected. The DBLE bit should be set while ADST bit is 0 (it should not be set simultaneously when 1 is written to the ADST bit.)

**TRGE Bit (Trigger Start Enable)**

The TRGE bit enables or disables A/D conversion by the synchronous trigger and the asynchronous trigger. This bit should be set to 1 in group scan mode.

**ADHSC Bit (A/D Conversion Mode Select)**

This bit selects whether A/D conversion mode is normal conversion mode or high-speed conversion mode.

Normal conversion mode can be selected when AVCC0 is in the range of 1.8 V to 3.6 V.

High-speed conversion mode can be selected when AVCC0  $\geq$  2.4 V.

Conversion can be performed in 1  $\mu$ s if the ADCLK clock is 32 MHz when AVCC0  $\geq$  2.7 V and high-speed conversion mode is selected.

**ADIE Bit (Scan End Interrupt Enable)**

The ADIE bit enables or disables the A/D scan end interrupt (S12ADI0) in scans except for group B scan in group scan mode.

With double trigger mode deselected, the S12ADI0 interrupt is generated when the first scan is completed if the ADIE bit is set to 1.

With temperature sensor output or internal reference voltage being selected, the S12ADI0 interrupt is also generated when A/D conversion is completed if the ADIE bit is set to 1.

With double trigger mode selected, the S12ADI0 interrupt is generated when the second scan is completed if the ADIE bit is set to 1 as long as the scan is started by the MTU or ELC trigger selected by the ADSTRGR.TRSA[3:0] bits.

When scanning is started by a software trigger, even with double trigger mode selected, the S12ADI0 interrupt is generated when scanning is completed if the ADIE bit is set to 1.

**ADCS[1:0] Bits (Scan Mode Select)**

The ADCS[1:0] bits select the scan mode.

In single scan mode, A/D conversion is performed for the analog inputs of a maximum of 17 channels selected with the ADANSA and ADANSA1 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 ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs of a maximum of 17 channels selected with the ADANSA register in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is repeated beginning at the first channel. A/D conversion is repeated until the ADCSR.ADST bit is set to 0.

In group scan mode, A/D conversion is performed for the analog inputs (group A) of a maximum of 17 channels selected with the ADANSA and ADANSA1 registers in the ascending order of the channel number after scanning is started by the MTU or ELC trigger selected by the ADSTRGR.TRSA[3:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. A/D conversion is also performed for the analog inputs (group B) of a maximum of 17 channels selected with the ADANSB and ADANSB1 registers in the ascending order of the channel number after scanning is started by the MTU or ELC trigger selected by the ADSTRGR.TRSA[3:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. In group scan mode, different channels and triggers should be selected for group A and group B.

When temperature sensor output or internal reference voltage is selected, single scan mode should be selected and all the channels selected by the ADANSA register should be deselected, after which A/D conversion is to be started. A/D conversion stops after completion of A/D conversion of the temperature sensor output or the internal reference voltage selected.

The ADCS[1:0] bits should be set while the ADST bit is 0 (it should not be set simultaneously when 1 is written to the ADST bit.)

**ADST Bit (A/D Conversion Start)**

The ADST bit starts or stops A/D conversion process.

Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input. [Setting conditions]

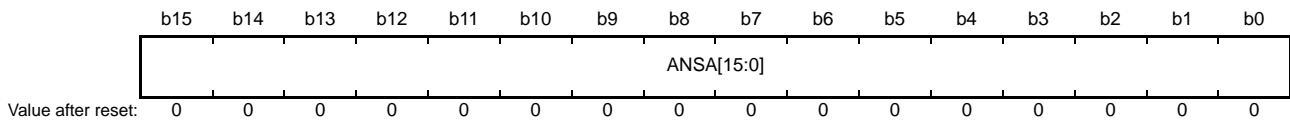
- 1 is written by software.
- The synchronous trigger (MTU or ELC) selected by the ADSTRGR.TRSA[3:0] bits is detected with ADCSR.EXTRG and ADCSR.TRGE bits being set to 0 and 1, respectively.
- A synchronous trigger (MTU or ELC) selected by the ADSTRGR.TRSA[3:0] bits is detected with the ADCSR.TRGE bit being set to 1 in group scan mode.
- The asynchronous trigger is detected with the ADCSR.TRGE and ADCSR.EXTRG bits being set to 1 and the ADSTRGR.TRSA[3:0] bits being set to 0000b.

[Clearing conditions]

- 0 is written by software.
- The A/D conversion of all the channels selected is completed in single scan mode.
- The A/D conversion of the temperature sensor output or the internal reference voltage selected is completed in single scan mode.
- Group A scan is completed in group scan mode.
- Group B scan is completed in group scan mode.

### 35.2.6 A/D Channel Select Register A (ADANSA)

Address(es): 0008 9004h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	ANSA[15:0]	A/D Conversion Channel 0 to 15 Select	0: AN000 to AN015 are not subjected to conversion. 1: AN000 to AN015 are subjected to scan conversion.	R/W

The ADANSA register selects analog input channels for A/D conversion from among AN000 to AN015. In group scan mode, group A channels are to be selected.

#### ANSA[15:0] Bits (A/D Conversion Channels 0 to 15 Select)

The ANSA[15:8, 6, 4:0] bits select analog input channels for A/D conversion from among AN000 to AN015. The channels to be selected and the number of channels can be arbitrarily set. The ANSA[0] bit corresponds to AN000 and the ANSA[15] bit 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 internal reference voltage should not be performed.

When double trigger mode is selected, the channel selected by the ANSA[15:0] bits is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

The ANSA[15:0] bits should be set while the ADCSR.ADST bit is 0.



### 35.2.7 A/D Channel Select Register A1 (ADANSA1)

Address(es): 0008 9006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	ANSA1 [5]	—	—	—	—	—
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	ANSA1[5]	A/D Conversion Channel 21 Select	0: AN021 is not subjected to conversion. 1: AN021 is subjected to scan conversion.	R/W
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADANSA1 register selects an analog input channel for A/D conversion from AN021. In group scan mode, group A channels are to be selected.

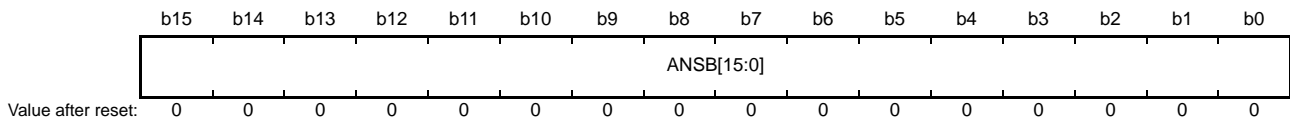
#### ANSA1[5] Bit (A/D Conversion Channel 21 Select)

The ANSA1[5] bit select an analog input channel for A/D conversion from AN021. The channels to be selected and the number of channels can be arbitrarily set. The ANSA1[5] bit corresponds to AN021. When A/D conversion of analog inputs of the channels is to be performed, A/D conversion of the temperature sensor output and the internal reference voltage should not be performed.

When double trigger mode is selected, the channel selected by the ANSA1[5] bit is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead. The ANSA1[15] bit should be set while the ADCSR.ADST bit is 0.

### 35.2.8 A/D Channel Select Register B (ADANSB)

Address(es): 0008 9014h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	ANSB[15:0]	A/D Conversion Channels 0 to 15 Select	0: AN000 to AN015 are not subjected to conversion. 1: AN000 to AN015 are subjected to scan conversion.	R/W

The ADANSB register selects analog inputs 0 to 15 of the channels for A/D conversion in group B in group scan mode. The ADANSB register is not used in any other scan mode.

#### ANSB[15:0] Bits (A/D Conversion Channels 0 to 15 Select)

The ADANSB register selects channels for A/D conversion in group B from among AN000 to AN015 in group scan mode. ADANSB is not used in any other scan mode. The channels for conversion can be selected from among the channels other than group A channels, which are selected by the ADANSA register or ADCSR.DBLANS[4:0] bits in double trigger mode. The ANSB[0] bit corresponds to AN000 and the ANSB[15] bit 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 internal reference voltage should not be performed.

The ANSB[15:0] bits should be set while the ADST bit is 0.

### 35.2.9 A/D Channel Select Register B1 (ADANSB1)

Address(es): 0008 9016h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	ANSB1[5]	—	—	—	—	—
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
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ANSB1[5]	A/D Conversion Channel 21 Select	0: AN021 is not subjected to conversion. 1: AN021 is subjected to scan conversion.	R/W
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

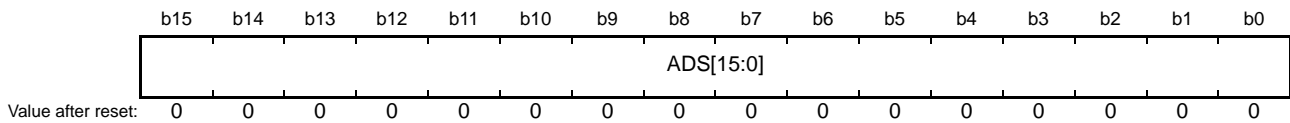
The ADANSB1 register selects an analog input 21 of the channel for A/D conversion in group B in group scan mode. The ADANSB1 register is not used in any other scan mode.

#### ANSB1[5] Bit (A/D Conversion Channel 21 Select)

The ADANSB1 register selects channels for A/D conversion in group B from AN021 in group scan mode. ADANSB1 is not used in any other scan mode. The channels for conversion can be selected from among the channels other than group A channels, which are selected by the ADANSA1 register or ADCSR.DBLANS[4:0] bits in double trigger mode. The ANSB1[5] bit corresponds to AN021. When A/D conversion of analog inputs of the channels is to be performed, A/D conversion of the temperature sensor output and the internal reference voltage should not be performed. The ANSB1[5] bit should be set while the ADST bit is 0.

### 35.2.10 A/D-Converted Value Addition Mode Select Register (ADADS)

Address(es): 0008 9008h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	ADS[15:0]	A/D-Converted Value Addition Channels 0 to 15 Select	0: A/D-converted value addition function for AN000 to AN015 are not selected. 1: A/D-converted value addition function for AN000 to AN015 are selected.	R/W

The ADADS register selects the channels 0 to 15 on which A/D conversion is performed successively two to four times and then converted values are added (integrated).

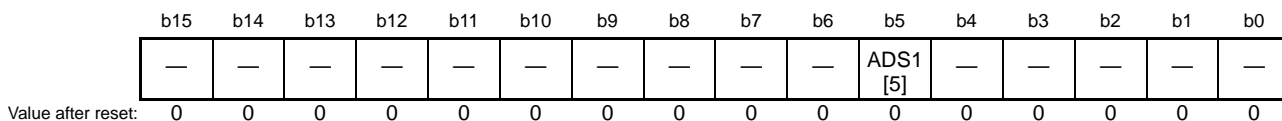
#### ADS[15:0] Bits (A/D-Converted Value Addition Channels 0 to 15 Select)

When the ADS[15:0] bits of the number that is the same as that of A/D converted channel selected by the ADANSA.ANSA[15:0] bits or ADCSR.DBLANS[4:0] bits and ADANSB.ANSB[15:0] bits is set to 1, these bits perform A/D conversion of analog input of the selected channels successively two to four times that is set with the ADADC.ADC[1:0] bits 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 ADS[15:0] bits should be set while the ADCSR.ADST bit is 0.

### 35.2.11 A/D Converted Value Addition Mode Select Register 1 (ADADS1)

Address(es): 0008 900Ah



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	ANSB1[5]	A/D-Converted Value Addition Channels 21 Select	0: A/D-converted value addition function for AN021 is not selected. 1: A/D-converted value addition function for AN021 is selected.	R/W
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADADS1 register selects the channel 21 on which A/D conversion is performed successively two to four times and then converted values are added (integrated).

#### ADS1[5] Bit (A/D-Converted Value Addition Channel 21 Select)

When the ADS1[5] bit of the number that is the same as that of A/D converted channel selected by the ADANSA1.ANSA[5] bit or ADCSR.DBLANS[4:0] bits and ADANSB1.ANSB[5] bit is set to 1, these bits perform A/D conversion of analog input of the selected channels successively two to four times that is set with the ADADC.ADC[1:0] bits 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[5] bit should be set while the ADCSR.ADST bit is 0.

Figure 35.2 shows a scanning operation sequence in which both the ADS[2] and ADS[4] bits are set to 1. In continuous scan mode (ADCSR.ADCS[1:0] = 10b), it is assumed that the addition count is set to 3 (ADADC.ADC[1:0] = 11b) and the channels AN000 to AN004, AN006 are selected (ADANSA.ANSA[15:0] = 005Fh). The conversion process begins with AN000. The AN002 conversion is performed successively 4 times, and the added (integrated) value is returned to A/D data register 2. After that the AN003 conversion process is started. The AN004 conversion is performed successively 4 times and the added (integrated) value is returned to A/D data register 4. After conversion of AN006, 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 ADCER.ADRFMT bit (right-alignment or left-alignment).

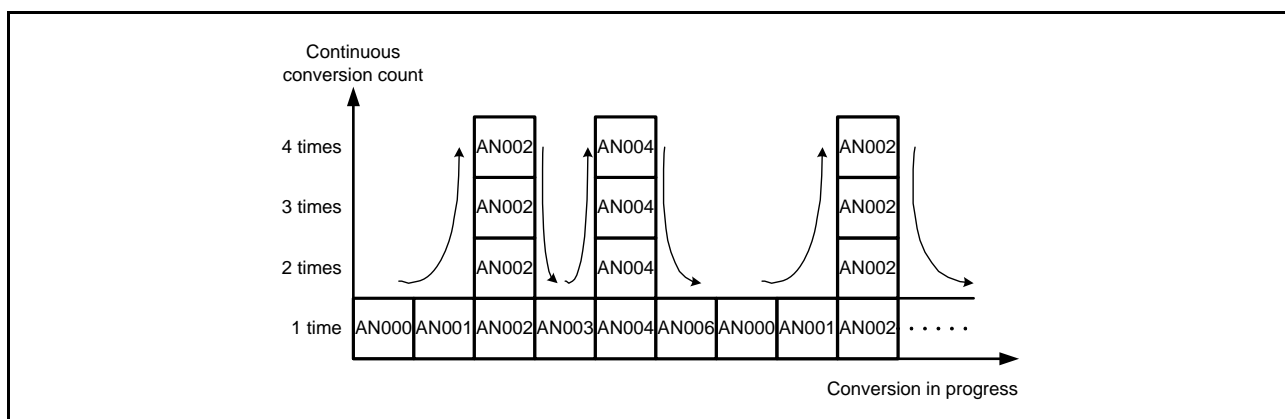
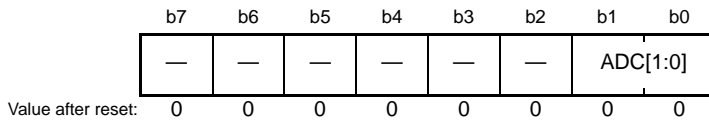


Figure 35.2 Scan Conversion Sequence with ADADC.ADC[1:0] = 11b, ADS[2] = 1, and ADS[4] = 1

### 35.2.12 A/D-Converted Value Addition Count Select Register (ADADC)

Address(es): 0008 900Ch



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

The ADADC register sets the addition count for the channels for which A/D-converted value addition mode is selected, and for A/D conversion of the temperature sensor output and the internal reference voltage.

#### ADC[1:0] Bits (Addition Count Select)

These bits set the addition count common to the channels for which A/D conversion or A/D-converted value addition mode is selected, including the channels selected in double trigger mode (by the ADCSR.DBLANS[4:0] bits), and to A/D conversion of the temperature sensor output and the internal reference voltage.

The ADC[1:0] bits should be set while the ADCSR.ADST bit is 0.

### 35.2.13 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: Right-alignment is selected for the A/D data register format. 1: Left-alignment is selected for the A/D data register format.	R/W

#### ACE Bit (Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (All 0) of ADDR<sub>y</sub>, ADOC<sub>DR</sub>, ADDBLDR, and ADTSDR after the register has been read by the CPU or DTC. This function enables update failures of the ADDR<sub>y</sub>, ADOC<sub>DR</sub>, ADDBLDR, and ADTSDR registers to be detected.

#### ADRFMT Bit (A/D Data Register Format Select)

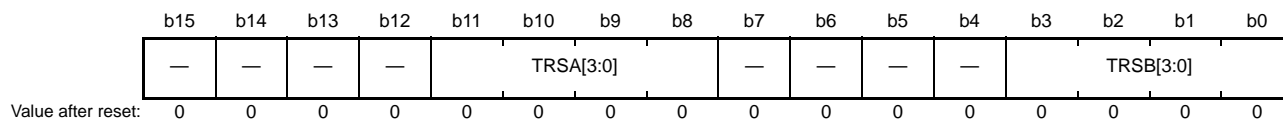
The ADRFMT bit specifies left-alignment or right-alignment for the data to be stored in ADDR<sub>y</sub>, ADOC<sub>DR</sub>, ADDBLDR, and ADTSDR.

When the A/D-converted value addition mode is selected, the format of each data register is fixed to left-alignment, irrespective of the ADCER.ADRFMT bit value.

For details on the format of the data registers, refer to section 35.2.1, A/D Data Registers <sub>y</sub> (ADDR<sub>y</sub>) (<sub>y</sub> = 0 to 15, 21), section 35.2.2, A/D Data Duplication Register (ADDBLDR), section 35.2.3, A/D Temperature Sensor Data Register (ADTSDR), and section 35.2.4, A/D Internal Reference Voltage Data Register (ADOC<sub>DR</sub>).

### 35.2.14 A/D Start Trigger Select Register (ADSTRGR)

Address(es): 0008 9010h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	TRSB[3:0]	A/D Conversion Start Trigger Select for Group B	Select the A/D conversion start trigger for group B in group scan mode.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	TRSA[3:0]	A/D Conversion Start Trigger Select	Select the A/D conversion start trigger in single scan mode and continuous mode. In group scan mode, the A/D conversion start trigger for group A is selected.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### TRSB[3:0] Bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[3:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[3:0] bits require to be set only in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting software trigger or asynchronous trigger is prohibited. Therefore, the TRSB[3:0] bits should be set to the value other than 0000b and the ADCSR.TRGE bit should be set to 1 in group scan mode.

Table 35.5 lists the A/D conversion startup sources selected by the TRSB[3:0] bits.

#### TRSA[3:0] Bits (A/D Conversion Start Trigger Select)

The TRSA[3:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode. In group scan mode, the trigger to start scanning of the analog input selected in group A is selected. For scan execution in group scan mode or double trigger mode, software trigger or asynchronous trigger cannot be used.

- When using the A/D conversion startup source of the synchronous trigger (MTU and ELC), set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 0.
- When using the asynchronous trigger (ADTRG0#), set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the set values of the ADCSR.TRGE and ADCSR.EXTRG bits and the TRSA[3:0] bits.

Table 35.6 lists the A/D conversion startup sources selected by the TRSA[3:0] bits.



**Table 35.5 List of A/D Conversion Startup Sources Selected by TRSB[3:0] Bits**

Module	Source	Remarks	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
MTU	TRG0AN	TRGA input capture/compare match from MTU0	0	0	0	1
	TRG0BN	TRGB input capture/compare match B from MTU0	0	0	1	0
	TRGAN	TRGA input capture/compare match or MTU4.TCNT underflow (trough) in complementary PWM mode from MTU0 to MTU4	0	0	1	1
	TRG0EN	TRGE compare match from MTU0	0	1	0	0
	TRG0FN	TRGF compare match from MTU0	0	1	0	1
	TRG4AN	MTU4.TADCORA and MTU4.TCNT compare match (interrupt skipping function 1)	0	1	1	0
	TRG4BN	MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1)	0	1	1	1
	TRG4ABN	MTU4.TADCORA and MTU4.TCNT compare match and MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1)	1	0	0	0
ELC	ELC	Trigger from ELC	1	0	0	1

**Table 35.6 List of A/D Conversion Startup Sources Selected by TRSA[3:0] Bits**

Module	Source	Remarks	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
ADC	ADST	Software trigger	—	—	—	—
External input	ADTRG0#	A/D conversion start trigger pin	0	0	0	0
MTU	TRG0AN	TRGA input capture/compare match from MTU0	0	0	0	1
	TRG0BN	TRGB input capture/compare match B from MTU0	0	0	1	0
	TRGAN	TRGA input capture/compare match or MTU4.TCNT underflow (trough) in complementary PWM mode from MTU0 to MTU4	0	0	1	1
	TRG0EN	TRGE compare match from MTU0	0	1	0	0
	TRG0FN	TRGF compare match from MTU0	0	1	0	1
	TRG4AN	MTU4.TADCORA and MTU4.TCNT compare match (interrupt skipping function 1)	0	1	1	0
	TRG4BN	MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1)	0	1	1	1
	TRG4ABN	MTU4.TADCORA and MTU4.TCNT compare match and MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1)	1	0	0	0
ELC	ELC	Trigger from ELC	1	0	0	1

### 35.2.15 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 Function Select	0: Temperature sensor output A/D-converted value addition function is not selected. 1: Temperature sensor output A/D-converted value addition function is selected.	R/W
b1	OCSAD	Internal Reference Voltage A/D-Converted Value Addition Mode Select	0: Internal reference voltage A/D-converted value addition mode is not selected 1: 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	Internal Reference Voltage A/D Conversion Select	0: A/D conversion of internal reference voltage is not performed 1: A/D conversion of internal reference voltage is performed	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### TSSAD Bit (Temperature Sensor Output A/D-Converted Value Addition Function Select)

When the TSSAD bit is set to 1, A/D conversion of temperature sensor output is selected and performed successively 2 to 4 times that is set with the ADADC.ADC[1:0] bits, and the added value is returned to the A/D temperature sensor data register (ADTSDR). The TSSAD bit should be set while the ADCSR.ADST bit is 0.

#### OCSAD Bit (Internal Reference Voltage A/D-Converted Value Addition Mode Select)

The OCSAD bit selects A/D conversion for the internal reference voltage. Setting the OCSAD bit to 1 performs A/D conversion of the internal reference voltage successively two to four times that is set with the ADADC.ADC[1:0] bits and returns the integrated value to the A/D internal reference voltage data register (ADOCADR). The OCSAD bit should be set while the ADCSR.ADST bit is 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 registers ADANSA, ADANSA1, ADANSB, and ADANSB1, the ADCSR.DBLE bit, and the OCS bit should be set to 0 in single scan mode. The TSS bit should be set while the ADST bit is 0. When comparator B (CMPB) is set to use the internal reference voltage, the temperature sensor output is not selected even by setting the TSS bit to 1.

Do not use the first conversion result after the TSS bit is set to 1. Insert a stabilization wait time of 5  $\mu$ s after the first conversion is completed and before the second conversion is started.

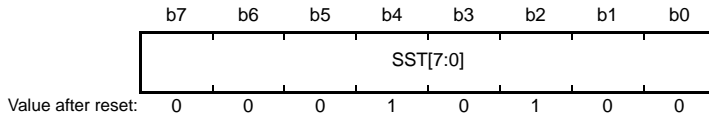
#### OCS Bit (Internal Reference Voltage A/D Conversion Select)

The OCS bit selects A/D conversion for the internal reference voltage. When A/D conversion of the internal reference voltage is to be performed, all the bits in the ADANSA and ADANSA1 registers should be set to 0 in single scan mode. The OCS bit should be set while the ADST bit is 0.

Do not use the first conversion result after the OCS bit is set to 1. Insert a stabilization wait time of 5  $\mu$ s after the first conversion is completed and before the second conversion is started.

### 35.2.16 A/D Sampling State Register n (ADSSTRn) (n = 0 to 7, 21, L, T, O)

Address(es): ADSSTR0: 0008 9060h, ADSSTR1: 0008 9073h, ADSSTR2: 0008 9074h, ADSSTR3: 0008 9075h,  
ADSSTR4: 0008 9076h, ADSSTR5: 0008 9077h, ADSSTR6: 0008 9078h, ADSSTR7: 0008 9079h,  
ADSSTR21: 0008 9080h, ADSSTRL: 0008 9061h, ADSSTRT: 0008 9070h, ADSSTRO: 0008 9071h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SST[7:0]	Sampling Time LO Setting	Sets the sampling time for each state (6 to 255 states).	R/W

ADSSTRn sets the sampling time for analog input.

The actual sampling time is the register setting value + one state.

One state is one ADCLK (A/D conversion clock) cycle. When the ADCLK is 32 MHz, one state is 31.25 ns. The initial value is 20 states. If the impedance of the analog input signal source is too high to secure sufficient sampling time or if the ADCLK is slow, the sampling time can be adjusted. These bits should be set while the ADCSR.ADST bit is 0. The set value for sampling time should be between 6 to 255 states.

Table 35.7 shows the A/D sampling state registers and corresponding channels.

**Table 35.7 A/D Sampling State Registers and Corresponding Channels**

Bit Name	Corresponding Channels
ADSSTR0.SST[7:0]	AN000
ADSSTR1.SST[7:0]	AN001
ADSSTR2.SST[7:0]	AN002
ADSSTR3.SST[7:0]	AN003
ADSSTR4.SST[7:0]	AN004
ADSSTR5.SST[7:0]	AN005
ADSSTR6.SST[7:0]	AN006
ADSSTR7.SST[7:0]	AN007
ADSSTRL.SST[7:0]	AN008 to AN015
ADSSTR21.SST[7:0]	AN021
ADSSTRT.SST[7:0]	Temperature sensor output
ADSSTRO.SST[7:0]	Internal reference voltage

### 35.2.17 A/D High-Side Reference Voltage Control Register (ADHVREFCNT)

Address(es): 0008 907Ch

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	OCSV SEL	HVRE FDIS
0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b0	HVREFDIS	High-Side Reference Voltage Discharge	0: Discharging of high-side reference voltage path stops 1: Discharging of high-side reference voltage path starts	R/W
b1	OCSVSEL	High-Side Reference Voltage	0: AVCC0 or VREFH0 is selected as high-side reference voltage (PJ6PFS.ASEL bit = 0: High-side reference voltage = AVCC0) (PJ6PFS.ASEL bit = 1: High-side reference voltage = VREFH0) 1: Internal reference voltage is selected as high-side reference voltage	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

The ADHVREFCNT register specifies the high-side reference voltage.

#### OCSVSEL Bit (High-Side Reference Voltage)

This bit specifies the high-side reference voltage. AVCC0, VREFH0, or the internal reference voltage can be selected as the high-side reference voltage. When this bit is set to 0, AVCC0 is selected as the high-side reference voltage while PJ6PFS.ASEL bit is 0, and VREFH0 is selected as the high-side reference voltage while PJ6PFS.ASEL bit is 1. When this bit is set to 1, the internal reference voltage is selected as the high-side reference voltage. When setting the OCSVSEL bit to 1, set HVREFDIS to 1 at the same time, and discharge the high-side reference voltage path. After discharging is completed, set the HVREFDIS bit to 0 and start A/D conversion.

The OCSVSEL bit can be set to 1 and the HVREFDIS bit can be set to 0 only after both bits are set to 1. When selecting the internal reference voltage, read the OCSVSEL and HVREFDIS bits and confirm that the OCSVSEL bit is 1 and the HVREFDIS bit is 0. A/D conversion can be performed on channels AN000 to AN015 and AN021, and A/D conversion of the internal reference voltage and temperature sensor output is prohibited.

When the internal reference voltage is selected, the voltage generation circuit operates and current increases by about 75  $\mu$ A. This circuit is not automatically turned off even if the MCU enters software standby mode with the internal reference voltage selected.

#### HVREFDIS Bit (High-Side Reference Voltage Discharge)

This bit starts or stops discharging of the high-side reference voltage path. This bit is enabled only when OCSVSEL is 1. Set the HVREFDIS and OCSVSEL bits to 1 at the same time, and discharge the high-side reference voltage path. After discharging is completed, set the HVREFDIS bit to 0 and start A/D conversion. The OCSVSEL bit can be set to 1 and the HVREFDIS bit can be set to 0 only after both bits are set to 1. When selecting the internal reference voltage, read the OCSVSEL and HVREFDIS bits after setting them, and then confirm that the OCSVSEL bit is 1 and the HVREFDIS bit is 0.

## 35.3 Operation

### 35.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

A scan conversion is performed in three operating modes: single scan mode, continuous scan mode, and group scan mode. Also, conversion modes are divided into high-speed conversion mode and normal conversion mode. In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until the ADCSR.ADST bit is cleared to 0 from 1 by software. In group scan mode, the selected channels of group A and the selected channels of group B are scanned once after starting to be scanned according to the respective triggers.

In single scan mode and continuous scan mode, A/D conversion is performed for ANn channels selected by the ADANSA and ADANSA1 registers, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for ANn channels of group A and group B selected by the ADANSA, ADANSA1, ADANSB, and ADANSB1 registers, respectively, starting from the channel with the smallest number n.

High-speed mode can be selected when  $AVCC0 \geq 2.4$  V. Normal conversion mode can be selected when  $AVCC0 \geq 1.8$  V. High-speed mode reduces conversion time by six cycles compared to normal conversion mode.

When temperature sensor output or internal reference voltage is selected, single scan mode should be used for A/D conversion.

This operation is similar to the scan operation when only one channel is selected in single scan mode.

Double trigger mode is to be used with single scan mode or group scan mode. With double trigger mode being enabled, A/D conversion data of a channel selected by the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by any of the MTU or ELC triggers selected by the ADSTRGR.TRSA[3:0] bits.

In any scanning mode, the software trigger, synchronous trigger, or asynchronous trigger input to be used as the A/D conversion start condition is disabled while the ADCSR.ADST bit is 1 (during scanning process).

### 35.3.2 Single Scan Mode

#### 35.3.2.1 Basic Operation

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below. In selected channel scanning, the temperature sensor output A/D conversion select bit (ADEXICR.TSS) and the internal reference voltage A/D conversion select bit (ADEXICR.OCS) should be set to 0 (not selected). The following describes an operation when double trigger mode is not selected.

- (1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger (MTU or ELC), or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA and ADANSA1 registers, starting from the channel with the smallest number n.
- (2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
- (3) When A/D conversion of all the selected channels is completed, an S12ADI0 interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).
- (4) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

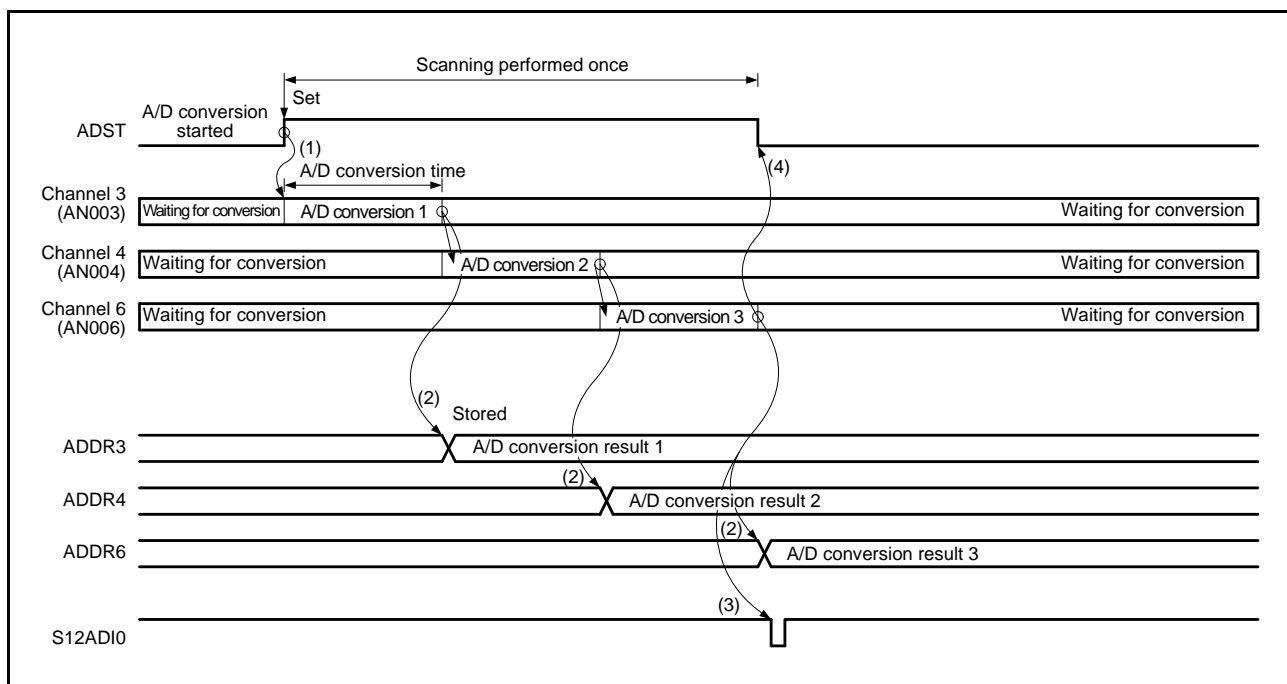


Figure 35.3 Example of Operation in Single Scan Mode (Basic Operation: AN003, AN004, AN006 Selected)

### 35.3.2.2 A/D Conversion When Temperature Sensor Output is Selected

To perform A/D conversion of the temperature sensor output, make the following settings before starting conversion operation.

- Set the ADCSR.ADST bit to 0 (stops A/D conversion).
- Set the ADCSR.ADCS[1:0] bits to 00b (single scan mode).
- Set the ADANSA and ADANSA1 registers to 0000h (all external analog inputs are not subject to conversion)
- Set the ADCSR.DBLE bit to 0 (deselects double trigger mode).
- Set the ADEXICR.TSS bit to 1 (A/D conversion of temperature sensor output is performed).
- Set the ADEXICR.OCS bit to 0 (A/D conversion of internal reference voltage is not performed).

Set an appropriate value in the ADSSTRO register to set the sampling time to 5  $\mu$ s or longer.

When comparator B (CMPB) is set to use the internal reference voltage, the temperature sensor output cannot be selected even by setting the ADEXICR.TSS bit to 1.

Figure 35.4 shows the procedure for A/D conversion of the temperature sensor output.

- (1) Set the ADST bit to 1 (starts A/D conversion) to discharge the electric charge stored in the A/D converter ((1) in the figure).  
Do not use the conversion result at this time.
- (2) When a trigger is input or the ADST bit is set to 1, A/D conversion is started for the temperature sensor output ((2) in the figure).
- (3) When A/D conversion is completed, the conversion result is stored into the ADTSDR register. If the ADCSR.ADIE bit is 1 (enables S12ADI0 interrupt generation upon scan completion.) at this time, an S12ADI0 interrupt request is generated ((3) in the figure).
- (4) The ADST bit is changed to 0 and the A/D converter enters a wait state ((4) in the figure).

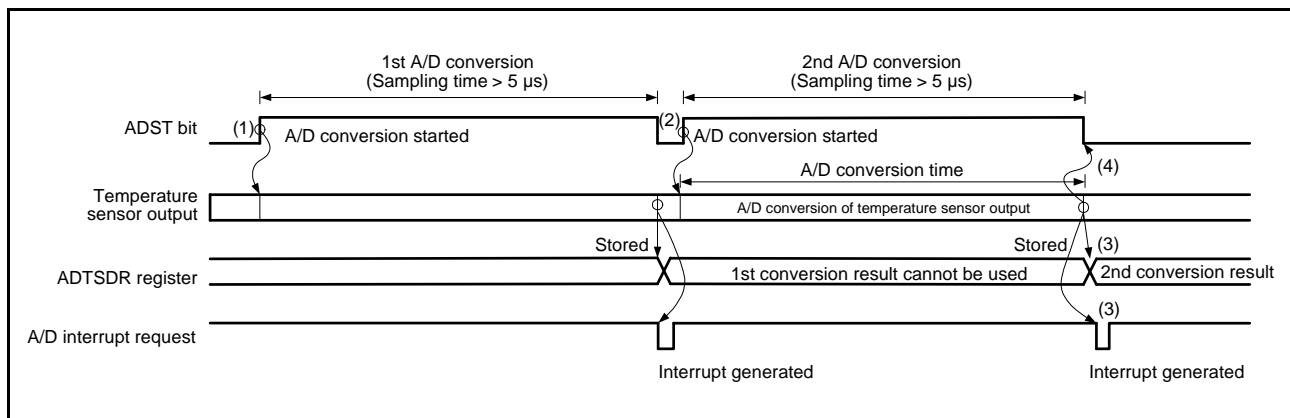


Figure 35.4 Example of Operation in Single Scan Mode (Temperature Sensor Output Selected)

### 35.3.2.3 A/D Conversion When Internal Reference Voltage is Selected

To perform A/D conversion of the internal reference voltage, make the following settings before starting conversion operation.

- Set the ADCSR.ADST bit to 0 (stops A/D conversion).
- Set the ADCSR.ADCS[1:0] bits to 00b (single scan mode).
- Set the ADANSA and ADANSA1 registers to 0000h (all external analog inputs are not subject to conversion)
- Set the ADCSR.DBLE bit to 0 (deselects double trigger mode).
- Set the ADEXICR.TSS bit to 0 (A/D conversion of temperature sensor output is not performed).
- Set the ADEXICR.OCS bit to 1 (A/D conversion of internal reference voltage is performed).

Set an appropriate value in the ADSSTRO register to set the sampling time to 5  $\mu$ s or longer.

Figure 35.5 shows the procedure for A/D conversion of the internal reference voltage.

- (1) Set the ADST bit to 1 (starts A/D conversion) to discharge the electric charge stored in the A/D converter ((1) in the figure).  
Do not use the conversion result at this time.
- (2) When a trigger is input or the ADST bit is set to 1, A/D conversion is started for the internal reference voltage ((2) in the figure).
- (3) When A/D conversion is completed, the conversion result is stored into the ADOCDR register. If the ADCSR.ADIE bit is 1 (enables S12ADI0 interrupt generation upon scan completion.) at this time, an S12ADI0 interrupt request is generated ((3) in the figure).
- (4) The ADST bit is changed to 0 and the A/D converter enters a wait state ((4) in the figure).

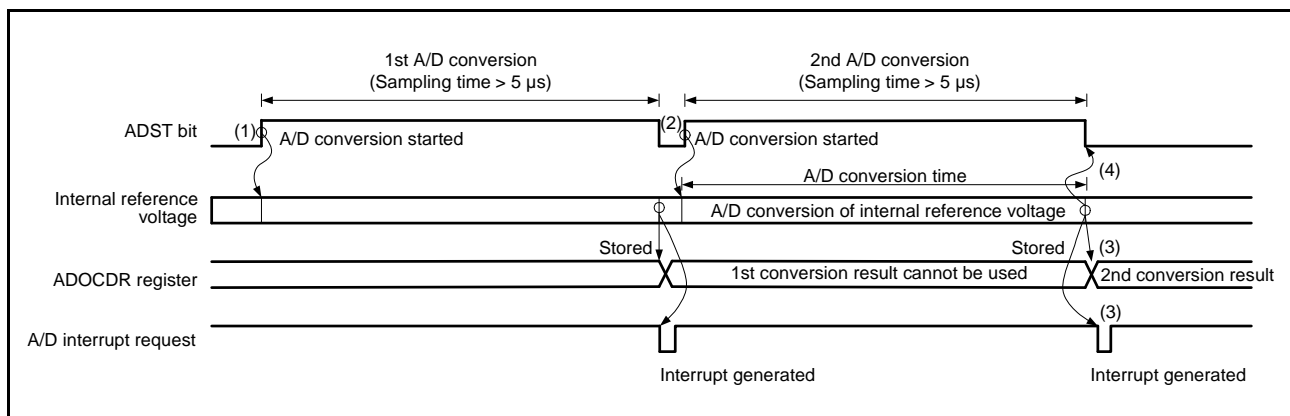


Figure 35.5 Example of Operation in Single Scan Mode (Internal Reference Voltage Selected)



### 35.3.2.4 A/D Conversion in Double Trigger Mode

In single scan mode with double trigger mode, single scan operation started by synchronous trigger is performed twice as below.

The temperature sensor output A/D conversion select bit (ADEXICR.TSS) and the internal reference voltage A/D conversion select bit (ADEXICR.OCS) should be set to 0 (not selected).

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the DBLANS[4:0] bits in ADCSR and setting the DBLE bit in ADCSR to 1. When the DBLE bit in ADCSR is set to 1, channel selection using the ADANSA and ADANSA1 registers is invalid. In double trigger mode, synchronous triggers should be selected using the ADSTRGR.TRSA[3:0] bits, the ADCSR.EXTRG bit should be set to 0, and the ADCSR.TRGE bit should be set to 1. Software trigger should not be used.

- (1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by synchronous trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
- (2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
- (3) The ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, an S12ADI0 interrupt request is not generated irrespective of the ADIE (S12ADI0 interrupt upon scanning completion enabled) bit setting in ADCSR.
- (4) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by the second trigger input, A/D conversion is started on the single channel selected by the DBLANS[4:0] bits in ADCSR.
- (5) When A/D conversion is completed, the A/D conversion result is stored into the A/D data duplication register (ADDBLDR), which is exclusively used in double trigger mode.
- (6) If the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled), an S12ADI0 interrupt request is generated.
- (7) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion is completed. Then the 12-bit A/D converter enters a wait state.

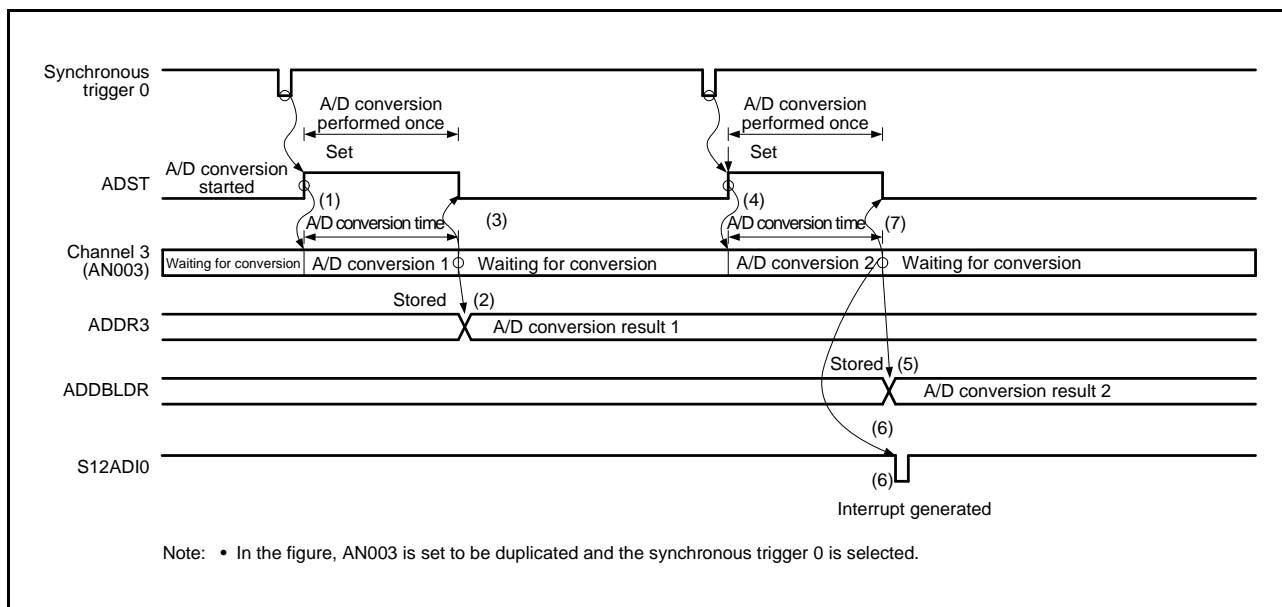


Figure 35.6 Example of Operation in Single Scan Mode (Double Trigger Mode Selected; AN003 Duplicated)

### 35.3.3 Continuous Scan Mode

#### 35.3.3.1 Basic Operation

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as below.

In continuous scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSS) and the internal reference voltage A/D conversion select bit (ADEXICR.OCS) should both be set to 0 (not selected).

- (1) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU or ELC), or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA and ADANSA1 registers, starting from the channel with the smallest number n.
- (2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
- (3) When A/D conversion of all the selected channels is completed, an S12ADI0 interrupt request is generated if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled).  
The 12-bit A/D converter sequentially starts A/D conversion for ANn channels selected by the ADANSA and ADANSA1 registers, starting from the channel with the smallest number n.
- (4) The ADST bit in ADCSR is not automatically cleared to 0 and steps 2 and 3 are repeated as long as the bit remains 1 (A/D conversion start). When the ADST bit in ADCSR is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
- (5) When the ADST bit is later set to 1 (A/D conversion start), A/D conversion is started again for ANn channels selected by the ADANSA and ADANSA1 registers, starting from the channel with the smallest number n.

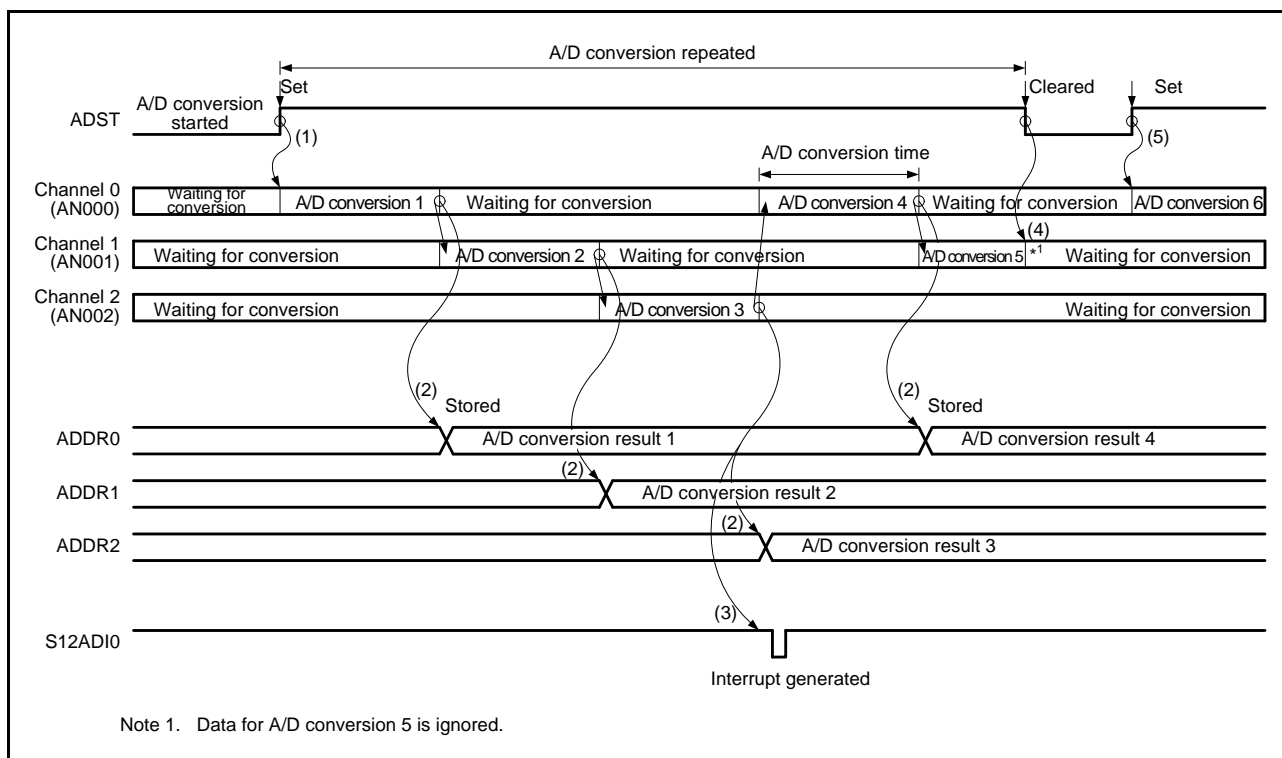


Figure 35.7 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 to AN002 Selected)

### 35.3.4 Group Scan Mode

#### 35.3.4.1 Basic Operation

In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in group A and group B after scanning is started by synchronous trigger as below. Scan operation of each group is similar to the scan operation in single scan mode.

The group A trigger and group B trigger can be selected using the TRSA[3:0] and TRSB[3:0] bits in ADSTRGR, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger should not be used.

The group A and group B channels to be A/D-converted are selected using the ADANSA and ADANSA1 registers, and ADANSB and ADANSB1 registers, respectively. Group A and group B cannot use the same channels.

In group scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSS) and the internal reference voltage A/D conversion select bit (ADEXICR.OCS) should both be set to 0 (not selected).

The following describes operation in group scan mode using a trigger from the MTU. Specifically, the TRG4AN and TRG4BN triggers from the MTU are assumed to be used to start conversion of group A and group B, respectively.

- (1) Scanning of group A is started by the TRG4AN trigger from the MTU.
- (2) When group A scanning is completed, an S12ADI0 interrupt is output if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt enabled).
- (3) Scanning of group B is started by the TRG4BN trigger from the MTU.
- (4) When group B scanning is completed, a GBADI interrupt is output if the GBADIE bit in ADCSR is 1 (GBADI interrupt enabled).

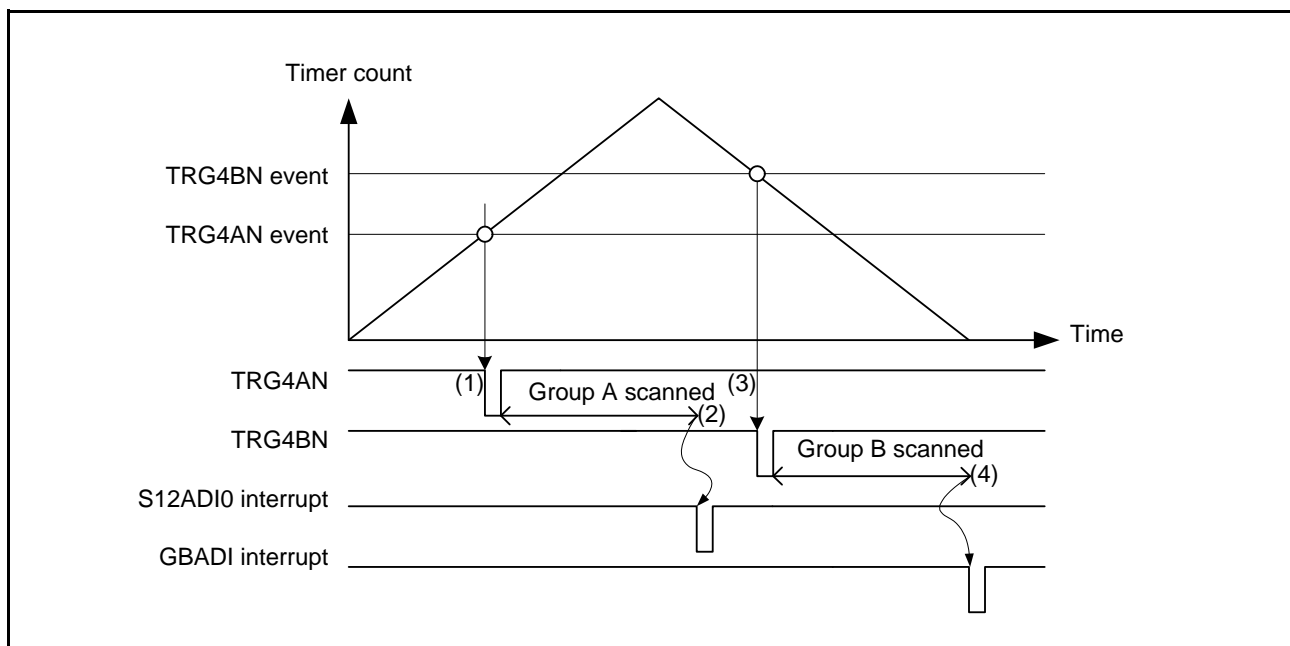


Figure 35.8 Example of Operation in Group Scan Mode (Basic Operation: MTU Triggers Used)

### 35.3.4.2 A/D Conversion in Double Trigger Mode

In group scan mode with double trigger mode, single scan operation started by synchronous trigger is performed twice for group A. For group B, single scan operation started by synchronous trigger is performed once.

In group scan mode, the group A trigger and group B trigger can be selected using the TRSA[3:0] and TRSB[3:0] bits in ADSTRGR, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger, or asynchronous trigger (ADTRG0#) should not be used.

The group A and group B channels to be A/D-converted are selected using the DBLANS[4:0] bits in the ADCSR register and ADANSB and ADANSB1 registers, respectively. The same channels cannot be selected for both groups.

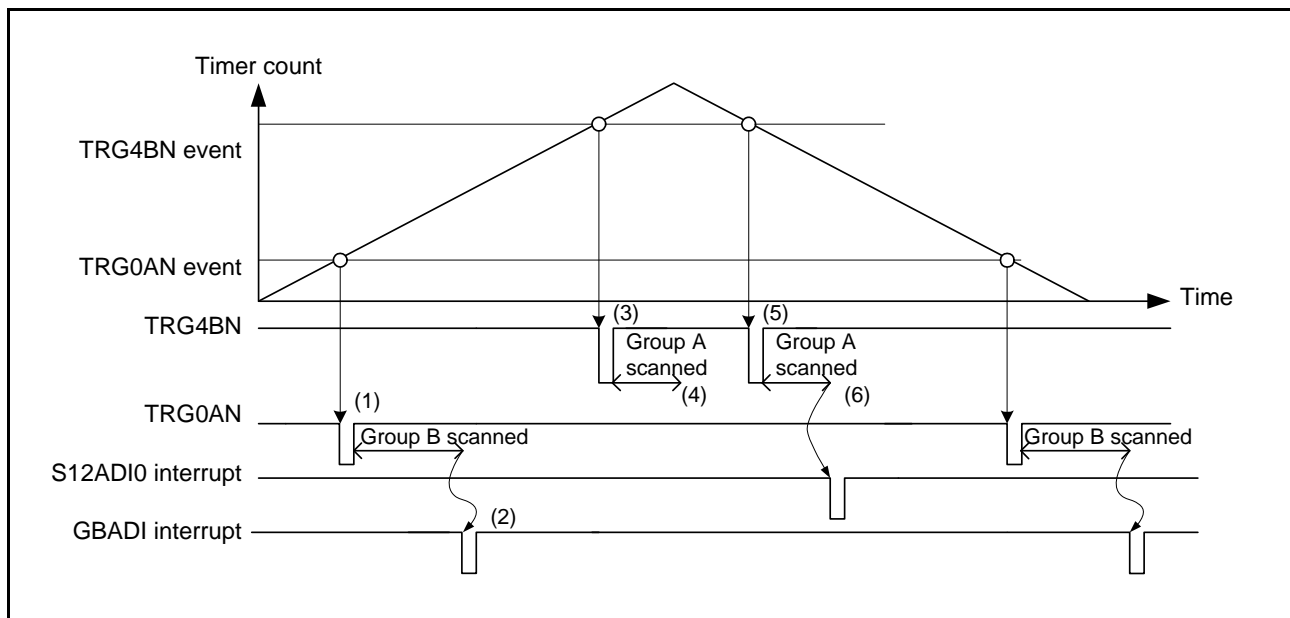
In group scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSS) and the internal reference voltage A/D conversion select bit (ADEXICR.OCS) should both be set to 0 (not selected).

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the DBLANS[4:0] bits in ADCSR and setting the DBLE bit in ADCSR to 1.

The following describes operation in group scan mode with double trigger mode using a trigger from the MTU.

Specifically, the TRG4BN and TRG0AN triggers from the MTU are assumed to be used to start conversion of group A and group B, respectively.

- (1) Scanning of group B is started by the TRG0AN trigger from the MTU.
- (2) When group B scanning is completed, a GBADI interrupt is output if the GBADIE bit in ADCSR is 1 (GBADI interrupt enabled).
- (3) The first scanning of group A is started by the first TRG4BN trigger from the MTU.
- (4) When the first scanning of group A is completed, the conversion result is stored into ADDRy; an S12ADI0 interrupt request is not generated irrespective of the ADIE bit setting in ADCSR.
- (5) The second scanning of group A is started by the second TRG4BN trigger from the MTU.
- (6) When the second scanning of group A is completed, the conversion result is stored into ADDBLDR. An S12ADI0 interrupt is output if the ADIE bit is 1 (S12ADI0 interrupt enabled).



**Figure 35.9 Example of Operation in Group Scan Mode with Double Trigger Mode (Basic Operation: MTU Triggers Used)**

### 35.3.4.3 Notes on Using Software Trigger

When a software trigger is input with double trigger mode selected, scanning of the selected channels is performed, and the S12ADI0 interrupt is output if the ADCSR.ADIE bit is 1 (enables S12ADI0 interrupt), regardless of scanning even or odd number of times. In addition, data is not duplicated even if scanning by a software trigger is performed even number of times.

The following shows an example when a software trigger is input while scanning by a synchronous trigger with double trigger mode selected.

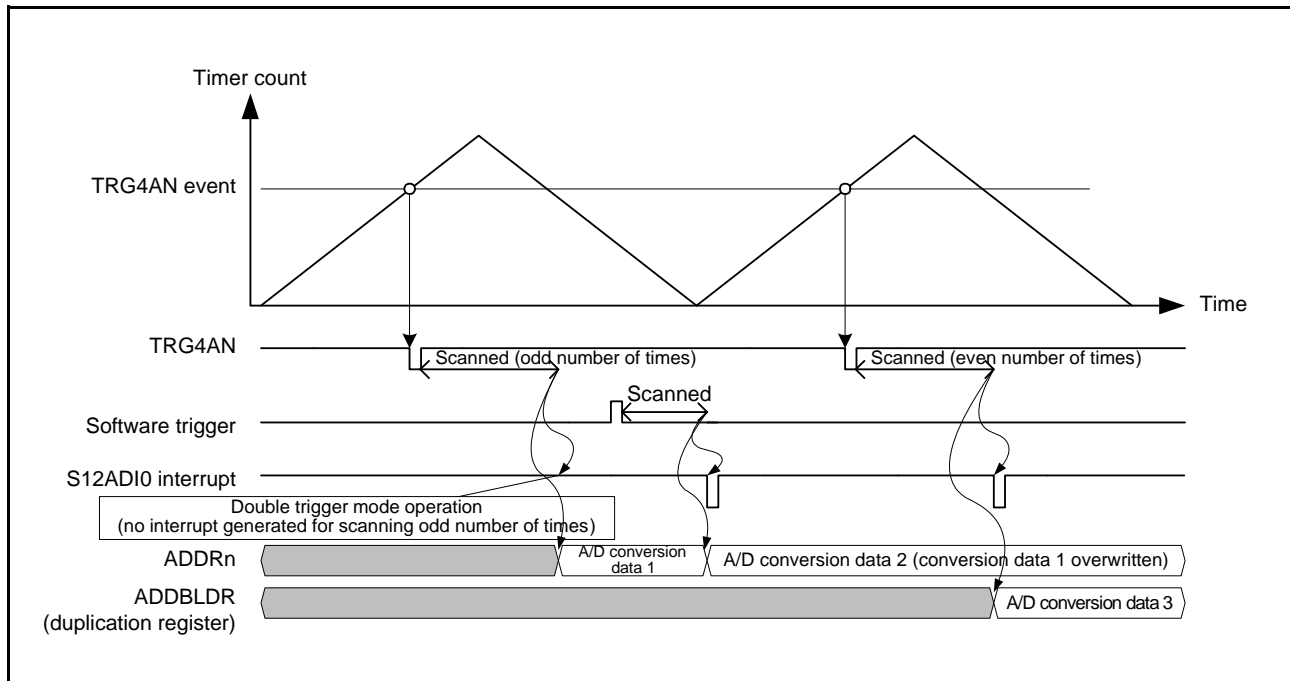


Figure 35.10 Example of Double Trigger Operation When Using Software Trigger

### 35.3.5 Analog Input Sampling and Scan Conversion Time

Scan conversion can be activated either by software trigger; the triggers from the MTU, ELC, or ADTRG0# (external trigger). After start-of-scanning-delay time ( $t_D$ ) has passed, then starts the A/D conversion process.

Figure 35.11 shows the scan conversion timing in single scan mode, in which scan conversion is activated by software trigger or triggers from the MTU or ELC. Figure 35.12 shows the scan conversion timing in single scan mode, in which scan conversion is activated by ADTRG0# (external trigger). The scan conversion time ( $t_{SCAN}$ ) includes start-of-scanning-delay time ( $t_D$ ), A/D conversion processing time ( $t_{CONV}$ ), and end-of-scanning-delay time ( $t_{ED}$ ). Table 35.8 shows the specific scanning time.

The scan conversion time ( $t_{SCAN}$ ) in single scan mode for which the number of selected channels is  $n$  can be determined as follows:

$$t_{SCAN} = t_D + (t_{SPL} + t_{CONV}) n + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is  $t_{SCAN}$  for single scan minus  $t_{ED}$ .

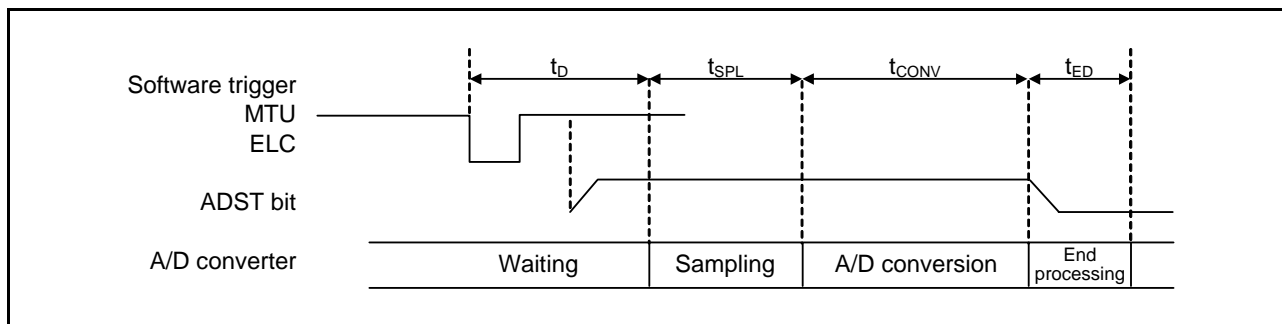
The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed to  $(t_{SPL} + t_{CONV}) n$ .

**Table 35.8 Scan Conversion Time (in Terms of PCLK and ADCLK Cycles)**

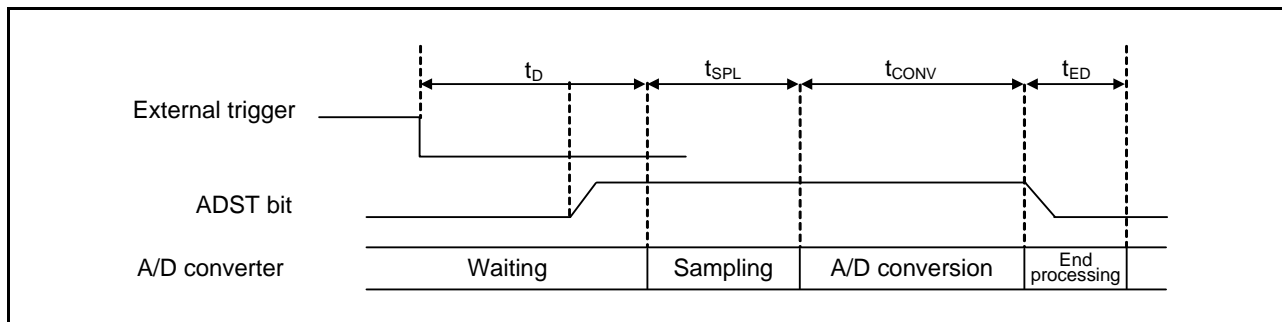
Item	Symbol	Conditions	Scan Conversion Time (Cycles)
Start-of-scanning-delay time*1	$t_D$	MTU, ELC, or software trigger	2 PCLK + 3 ADCLK
		External trigger	4 PCLK + 3 ADCLK
Sampling time*1	$t_{SPL}$	ADSSTRn.SST[7:0] bits (initial set value 14h)	(register set value + 1) ADCLK
A/D conversion processing time*1	$t_{CONV}$	High-speed mode	23 ADCLK
		Normal-speed mode	29 ADCLK
Scan conversion time*1	$t_{ED}$	—	1 PCLK + 2 ADCLK*2

Note 1. For  $t_D$ ,  $t_{SPL}$ ,  $t_{CONV}$ , and  $t_{ED}$ , see Figure 35.11 and Figure 35.12.

Note 2. The value of 2 ADCLK is fixed and an interrupt is output within plus 1 PCLK. For details on the processing time for termination, refer to section 35.8.3, A/D Conversion Restarting Timing and Termination Timing.



**Figure 35.11 Scan Conversion Timing (Activated by Software, or Triggers from the MTU or ELC)**



**Figure 35.12 Scan Conversion Timing (Activated by ADTRG0#)**

### 35.3.6 Usage Example of Automatic Register Clearing Function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADOCDR, ADTSDR, and ADDBLDR) to 0000h when the A/D data registers are read by the CPU or DTC.

This function enables detection of update failures of the A/D data registers. The following describes the examples in which the function to automatically clear the ADDRy register is enabled and disabled.

In a case where the ADCER.ACE bit is 0 (automatic clearing is disabled), if the A/D conversion result (0222h) is not written to the ADDRy register for some reason, the old data (0111h) will be the ADDRy value. Furthermore, if this ADDRy value is written to a general register using an A/D scan end interrupt, the old data (0111h) can be saved in the general register. When checking whether there is an update failure, it is necessary to frequently save the old data in the RAM or a general register.

In a case where the ADCER.ACE bit is 1 (automatic clearing is enabled), when ADDRy = 0111h is read by the CPU, DTC, or ADDRy is automatically cleared to 0000h. After that, if the A/D conversion result 0222h cannot be transferred to ADDRy for some reason, the cleared data (0000h) remains as the ADDRy value. If this ADDRy value is read into a general register using an A/D scan end interrupt at this point, 0000h will be saved in the general register. Occurrence of an ADDRy update failure can be determined by simply checking that the read data value is 0000h.

### 35.3.7 A/D-Converted Value Addition Function

The same channel is A/D converted two to four consecutive 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.

### 35.3.8 Starting A/D Conversion with an Asynchronous Trigger

The A/D conversion can be started by the input of an asynchronous trigger. To start up the A/D converter by an asynchronous trigger, the A/D conversion start trigger select bits (ADSTRGR.TRSA[3:0]) should be set to 0000b and a high-level signal should be input to the asynchronous trigger (ADTRG0# pin), and both the ADCSR.TRGE and ADCSR.EXTRG bits should be set to 1. Figure 35.13 shows a timing of the asynchronous trigger input.

For the time required for the A/D conversion start after the ADCSR.ADST bit is set, refer to section 35.8.3, A/D Conversion Restarting Timing and Termination Timing.

An asynchronous trigger cannot be selected by the A/D conversion start trigger select bits (ADSTRGR.TRSA[3:0]) for group B to be used in group scan mode.

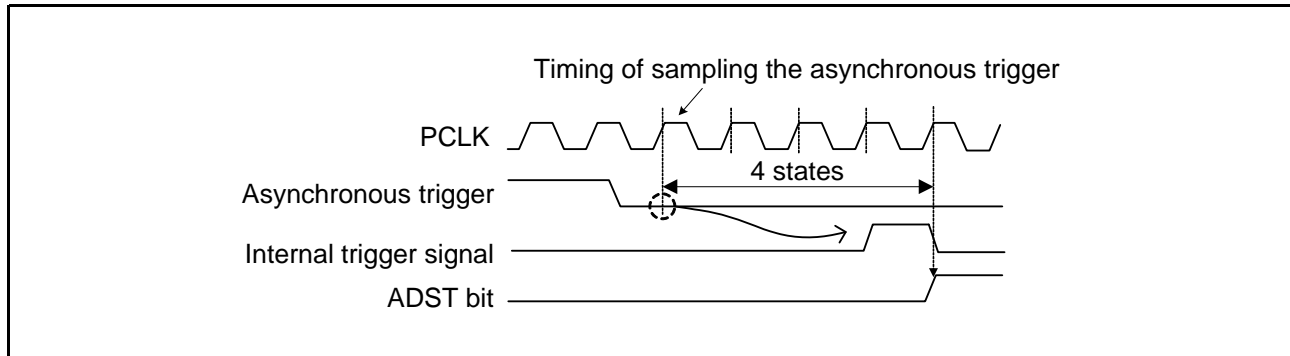


Figure 35.13 Asynchronous Trigger Input Timing

### 35.3.9 Starting A/D Conversion with Synchronous Trigger from Peripheral Modules

The A/D conversion can be started by a synchronous trigger of the MTU or ELC. To start the A/D conversion by a synchronous trigger, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, and the relevant source should be selected by ADSTRGR.TRSA[3:0] and TRSB[3:0] bits.

The A/D conversion startup sources for group B to be used in group scan mode should be selected by the ADSTRGR.TRSB[3:0] bits. In group scan mode, the different A/D start conversion startup sources should be selected by the ADSTRGR.TRSA[3:0] bits and the ADSTRGR.TRSB[3:0] bits.

## 35.4 Interrupt Sources

### 35.4.1 Interrupt Request on Completion of Each Scanning Conversion

The 12-bit A/D converter can send scan end interrupt requests S12ADI0 and GBADI to the CPU.

Setting the ADCSR.ADIE bit to 1 and 0 enables and disables an S12ADI0 interrupt, respectively. Similarly, setting the ADCSR.GBADIE bit to 1 and 0 enables and disables a GBADI interrupt, respectively.

In addition, the DTC can be started up when an S12ADI0 or a GBADI interrupt is generated. Using an S12ADI0 or a GBADI interrupt to allow the DTC to read the converted data enables continuous conversion without burden on software. For details on DTC settings, refer to section 16, Data Transfer Controller (DTCa).



## 35.5 Event Linkage

### 35.5.1 Event Output to ELC

The ELC connects the S12ADI0 interrupt request signal to the predetermined module as the event signal (i.e., event linkage). The GBADI interrupt request signal cannot be used as the event signal. The event signal can be output irrespective of the setting of the corresponding interrupt request enable bit. The 12-bit A/D converter outputs the A/D conversion end event.

### 35.5.2 12-Bit A/D Converter Operation by Event from ELC

The 12-bit A/D converter can be started by the predetermined event by so setting ELSRn of the ELC.

### 35.5.3 Notes on Event Reception from ELC during 12-Bit A/D Conversion

When an event occurs during A/D conversion, it is invalid.

## 35.6 Selecting Reference Voltage

For the A/D converter, VREFH0, AVCC0, or the internal reference voltage can be selected as the high-side reference voltage, and VREFL0 or AVSS0 can be selected as the low-side reference voltage. Set these voltages before performing A/D conversion. Select VREFH0 or AVCC0 as the high-side reference voltage by setting the PJ6PFS.ASEL bit when the ADHVREFCNT.OCSVSEL bit is 0, and select the internal reference voltage when the ADHVREFCNT.OCSVSEL bit is 1. For the procedure for selecting the internal reference voltage, refer to section 35.6.1, When Selecting Internal Reference Voltage as High-Side Reference Voltage. The low-side reference voltage is selected by the PJ7PFS.ASEL bit. For details, section 19, Multi-Function Pin Controller (MPC).

### 35.6.1 When Selecting Internal Reference Voltage as High-Side Reference Voltage

The following shows the procedure for selecting the internal reference voltage as the high-side reference voltage to perform A/D conversion. At this time, A/D conversion can be performed on channels AN000 to AN015 and AN021, and A/D conversion of the internal reference voltage and temperature sensor output is prohibited.

1. Set the ADHVREFCNT.OCSVSEL and ADHVREFCNT.HVREFDIS bits to 1 at the same time, and discharge the internal high-side reference voltage path in the A/D converter.
2. Wait for the discharge period of 1  $\mu$ s by software.
3. Set the ADHVREFCNT.HVREFDIS bit to 0 to complete discharging of the path. At this time, read the OCSVSEL and HVREFDIS bits and confirm that the OCSVSEL bit is 1 and the HVREFDIS bit is 0.  
Even if an attempt is made to select the internal reference voltage without discharging the path after VREFH0 or AVCC0 is selected, the OCSVSEL bit cannot be set to 1 and the HVREFDIS bit cannot be set to 0.
4. Perform A/D conversion after waiting for the stabilization wait time (5  $\mu$ s) of the internal reference voltage.

Figure 35.14 shows waveforms when using the procedure for selecting the high-side reference voltage as the internal reference voltage.

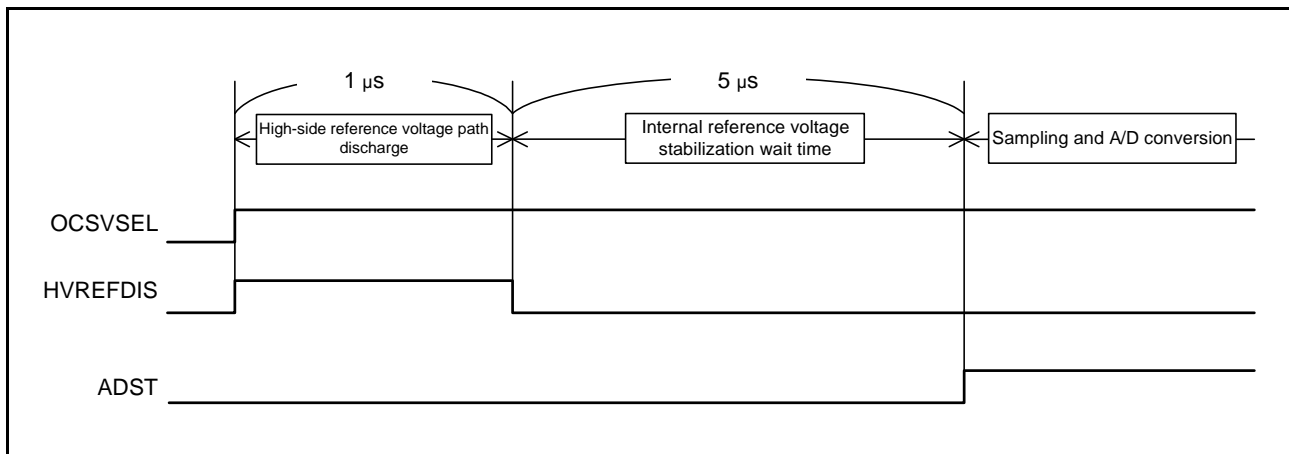


Figure 35.14 Procedure for Selecting Internal Reference Voltage as Reference Voltage

### 35.7 A/D Conversion Accuracy Definitions

The A/D conversion accuracy is defined as below:

- Resolution  
The number of 12-bit A/D converter digital output codes
- 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 000000000000 to 000000000001, excluding quantization error.
- Full-scale error  
The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 111111111110 to 111111111111, excluding quantization error.
- Quantization error  
The deviation inherent in the 12-bit A/D converter, given by 1/2 LSB
- Nonlinearity error  
The error with respect to the ideal A/D conversion characteristic between zero voltage and the full-scale error, excluding offset error, full-scale error, and quantization error.
- Absolute accuracy  
The deviation between the digital value and analog input value, including offset error, full-scale error, quantization error, and nonlinearity error.

## 35.8 Usage Notes

### 35.8.1 Notes on Reading Data Registers

The A/D data registers, A/D data duplication register, A/D temperature sensor data register, and A/D internal reference voltage data register should be read in word units. If a register is read twice in byte units, that is, the higher-order byte and lower-order byte are separately read, the A/D converted value having been read first may disagree with the A/D converted value having been read for the second time. To prevent this, the data registers should never be read in byte units.

### 35.8.2 Notes on Stopping A/D Conversion

To stop A/D conversion when an asynchronous trigger or a synchronous trigger has been selected as the condition for starting A/D conversion, set the ADCSR.TRGE bit to 0 and select the software trigger as the condition for starting A/D conversion, and then set the ADCSR.TRGE bit to 0 (to stop A/D conversion).

### 35.8.3 A/D Conversion Restarting Timing and Termination Timing

It takes a maximum of four ADCLK cycles for the idle analog unit of the 12-bit A/D converter to be restarted by setting the ADCSR.TRGE bit to 1. It takes a maximum of two ADCLK cycles for the operating analog unit of the 12-bit A/D converter to be terminated by setting the ADCSR.TRGE bit to 0.

### 35.8.4 Notes on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D converted data is overwritten with the second A/D converted data in the case that the CPU does not complete reading the A/D converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

### 35.8.5 Module Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled by setting module stop control register A (MSTPCRA). The initial setting is for operation of the 12-bit A/D converter to be halted. Register access is enabled by releasing the module stop state.

After the module stop state is released, wait for 1  $\mu$ s to start A/D conversion. For details, refer to section 11, Low Power Consumption.

### 35.8.6 Notes on Entering Low Power Consumption States

Before entering module stop mode or software standby mode, make sure to stop A/D conversion. Here, set the ADCSR.ADST bit to 0, and allow time for stopping the analog unit of the 12-bit A/D converter.

Follow the procedure given below to secure this time.

1. Set the ADCSR.TRGE bit to 0 (software trigger).
2. Set the ADCSR.ADST bit to 0.
3. After confirming that the A/D converter has been stopped, place the MCU in the module stop state mode or software standby mode.

### 35.8.7 Voltage Relationship between AVCC0 and VCC

AVCC0 and VCC can be set individually, but the following restrictions apply.

- When the PJ0 and PJ2 pins are used as D/A output pins, make sure that the D/A output voltage does not exceed VCC.
- When the PJ0 and PJ2 pins are used as I/O ports, make sure that the VCC does not exceed the AVCC0 voltage.
- When the PJ0 and PJ2 pins are not used, these restrictions do not apply.

### 35.8.8 Notes on Releasing Software Standby Mode

After software standby mode is released, wait until the crystal oscillation stabilization time or the PLL circuit stabilization time elapses, and then wait for 1  $\mu$ s before starting A/D conversion. For details, refer to section 11, Low Power Consumption.

### 35.8.9 Allowable Impedance of Signal Source

To achieve high-speed conversion of 1.0  $\mu$ s, the analog input pins of this MCU are designed so that the conversion accuracy is guaranteed if the impedance of the input signal source is 0.3 k $\Omega$  or less. If an external capacitor of large capacitance is attached in the application in which only a single pin input is converted in single scan mode, the only load on input is virtually 2.6 k $\Omega$  of the internal input resistor; therefore, the impedance of the signal source can be ignored. Being a low-pass filter, however, an analog input circuit may not follow the analog signal with a large differential coefficient as shown in Figure 35.15. When high-speed analog signals are to be converted or multiple pins are to be converted in scan mode, a low-impedance buffer should be used.

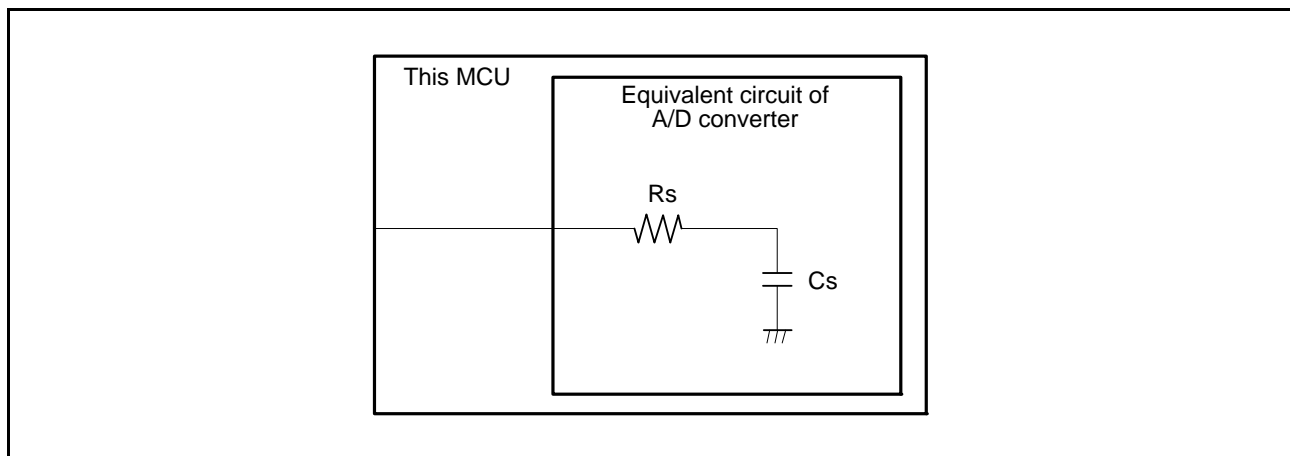


Figure 35.15 Internal Equivalent Circuit of Analog Input Pin

Figure 35.16 shows an equivalent circuit of an analog input pin and an external sensor.

To perform A/D conversion accurately, charging of the internal capacitor C shown in Figure 35.16 must be completed within the specified period of time. This specified period is referred to as sampling time.

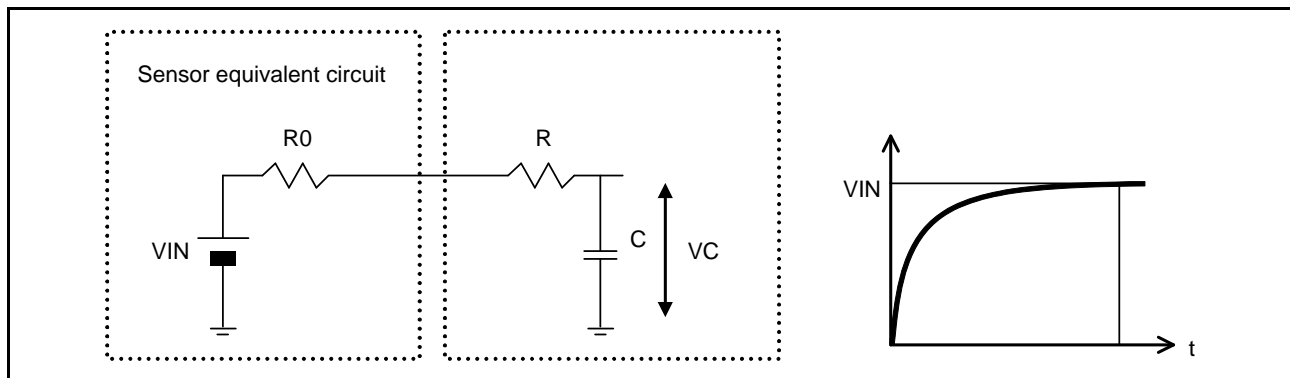


Figure 35.16 Equivalent Circuit of Analog Input Pin and External Sensor

Let the sampling time be  $T$  [s], the output impedance of the sensor be  $R0$  [ $\Omega$ ], the internal resistance of the MCU be  $R$  [ $\Omega$ ], the accuracy (error) of the A/D converter be  $x$  [LSB], and the resolution of the A/D converter be  $y$  [tone] (4096 in 12-bit mode), the voltage difference  $VC$  between both sides of the capacitor  $C$  is expressed as follows:

$$VC = VIN \left\{ 1 - e^{-\frac{T}{C(R0+R)}} \right\}$$

When  $t = T$ , to obtain a conversion error of  $x$  or less,

$$VC = VIN - \frac{x}{y}VIN = VIN \left( 1 - \frac{x}{y} \right)$$

Thus,  $R0$  can be calculated as follows:

$$\begin{aligned} e^{-\frac{T}{C(R0+R)}} &= \frac{x}{y} \\ -\frac{T}{C(R0+R)} &= \ln \frac{x}{y} \\ R0 &= -\frac{T}{C \ln \frac{x}{y}} - R \end{aligned}$$

When  $fPCLKD = 32$  MHz, the output impedance  $R0$  of the sensor for an error of 0.1 LSB or less is obtained by the following equation:

$T = 0.3 \mu\text{s}$ ,  $x = 0.1$ ,  $y = 4096$ ,  $R = 2.6 \text{ k}\Omega$  (reference value), and  $C = 7 \text{ pF}$  (reference value)

Hence,

$$\begin{aligned} R0 &= -\frac{0.3 \times 10^{-6}}{7 \times 10^{-12} \times \ln \left( \frac{0.1}{4096} \right)} - 2.6 \times 10^3 \\ &= 1435 \end{aligned}$$

Thus, the output impedance  $R0$  of the sensor must be approximately 1.4 k $\Omega$  or less to obtain the A/D converter accuracy (error) of 0.1 LSB or less.

Actual error, however, is the value of absolute accuracy added to the above 0.1 LSB.

These values are for reference and operation must be verified by performing evaluations.

### 35.8.10 Influence on Absolute Accuracy

Attaching a capacitor creates coupling with GND and may affect the absolute accuracy when noisy GND is used; therefore, a capacitor should be connected to electrically stable GND such as AVSS0.

The filter circuit should be designed so that it does not interfere digital signals or it does not serve as an antenna on the circuit board.

### 35.8.11 Voltage Range of Analog Power Supply Pins

If this MCU is used with the voltages outside the following ranges, the reliability of the MCU may be affected.

- Analog input voltage range

Voltage (VAN) applied to analog input pins ANn:  $VREFL0 \leq VAN \leq VREFH0$

Reference voltage range applied to the VREFH0 pin:  $VREFH0 \leq AVCC0$

Voltage applied to analog input pins ANn (n = 0 to 7, 21):  $AVSS0 \leq VAN \leq AVCC0$

Voltage applied to analog input pins ANn (n = 8 to 15):  $VSS \leq VAN \leq VCC$  and  $VSS \leq VAN \leq AVCC$

- Relationship between power supply pin pairs (AVCC0–AVSS0, VREFH0–VREFL0, VCC–VSS)

Relationship between AVSS0 and VSS:  $AVSS0 = VSS$ . A 0.1- $\mu$ F capacitor should be connected between each pair of power supply pins to create a closed loop with the shortest route possible as shown in Figure 35.17, and connection should be made so that the following conditions are satisfied at the supply side.

$VREFL0 = AVSS0 = VSS$

When the 12-bit A/D converter is not used, the following conditions should be satisfied.

$VREFH0 = AVCC0 = VCC$  and  $VREFL0 = AVSS0 = VSS$

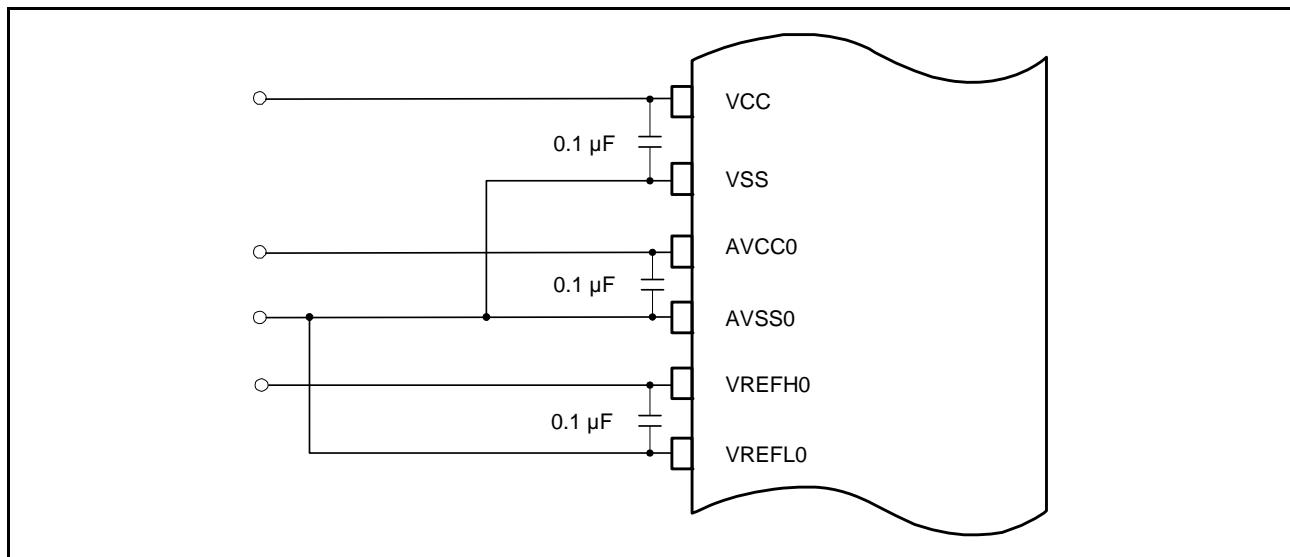


Figure 35.17 Power Supply Pin Connection Example

### 35.8.12 Notes on Board Design

The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (AN000 to AN015, AN021), reference power supply pin (VREFH0), reference ground pin (VREFL0), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0). The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

### 35.8.13 Notes on Noise Prevention

To prevent the analog input pins (AN000 to AN015, AN021) from being destroyed by abnormal voltage such as excessive surge, a capacitor should be inserted between AVCC0 and AVSS0 and between VREFH0 and VREFL0, and a protection circuit should be connected to protect the analog input pins (AN000 to AN015, AN021) as shown Figure 35.18.

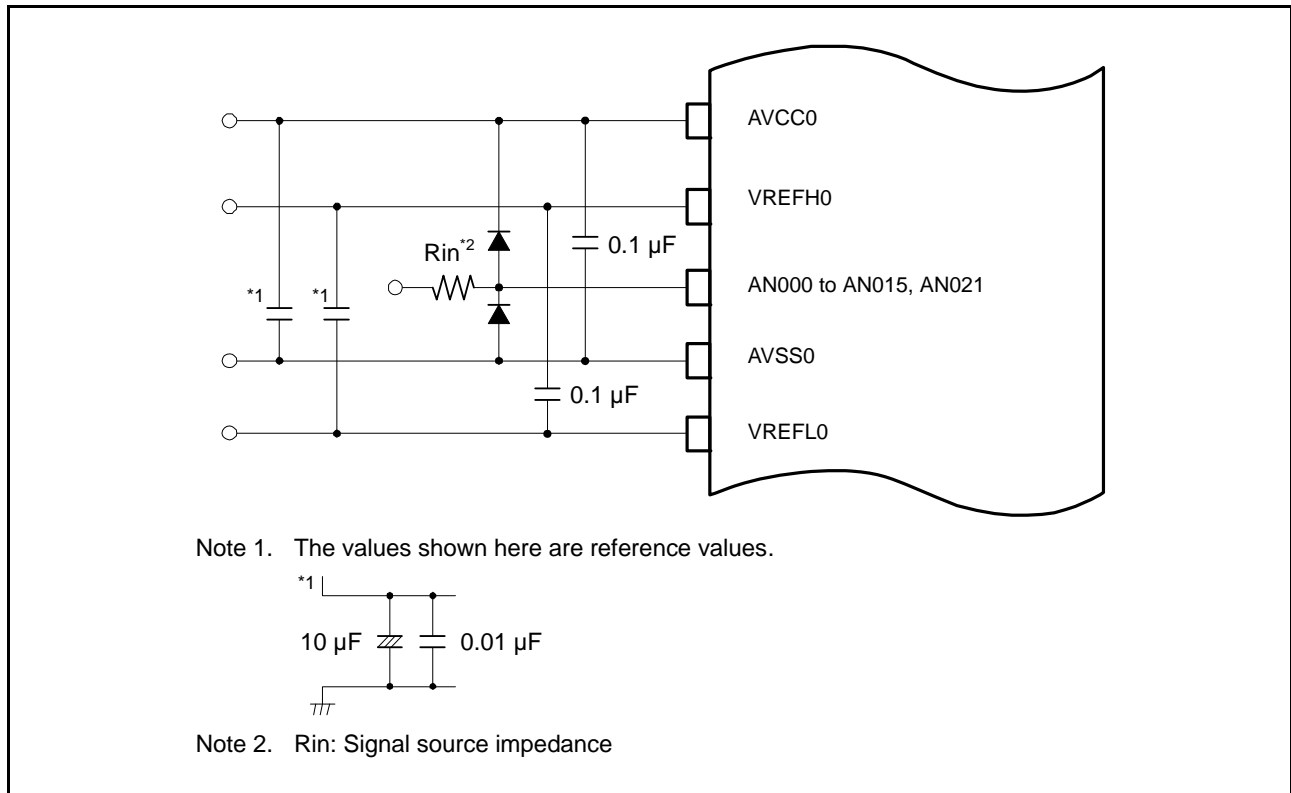


Figure 35.18 Sample Protection Circuit for Analog Inputs

### 35.8.14 Port Setting When 12-Bit A/D Converter Inputs are Used

When the 12-bit A/D converter is used, output from port 4 and port 9 should not be used. This is because an analog power supply is used for parts of the port 4 and port 9 circuits.

### 35.8.15 Sequence of Powering on AVCC0 and VCC

When powering on AVCC0 and VCC, power them on at the same time or VCC first.

## 36. 12-Bit D/A Converter (R12DAA)

In this section, “PCLK” is used to refer to PCLKB.

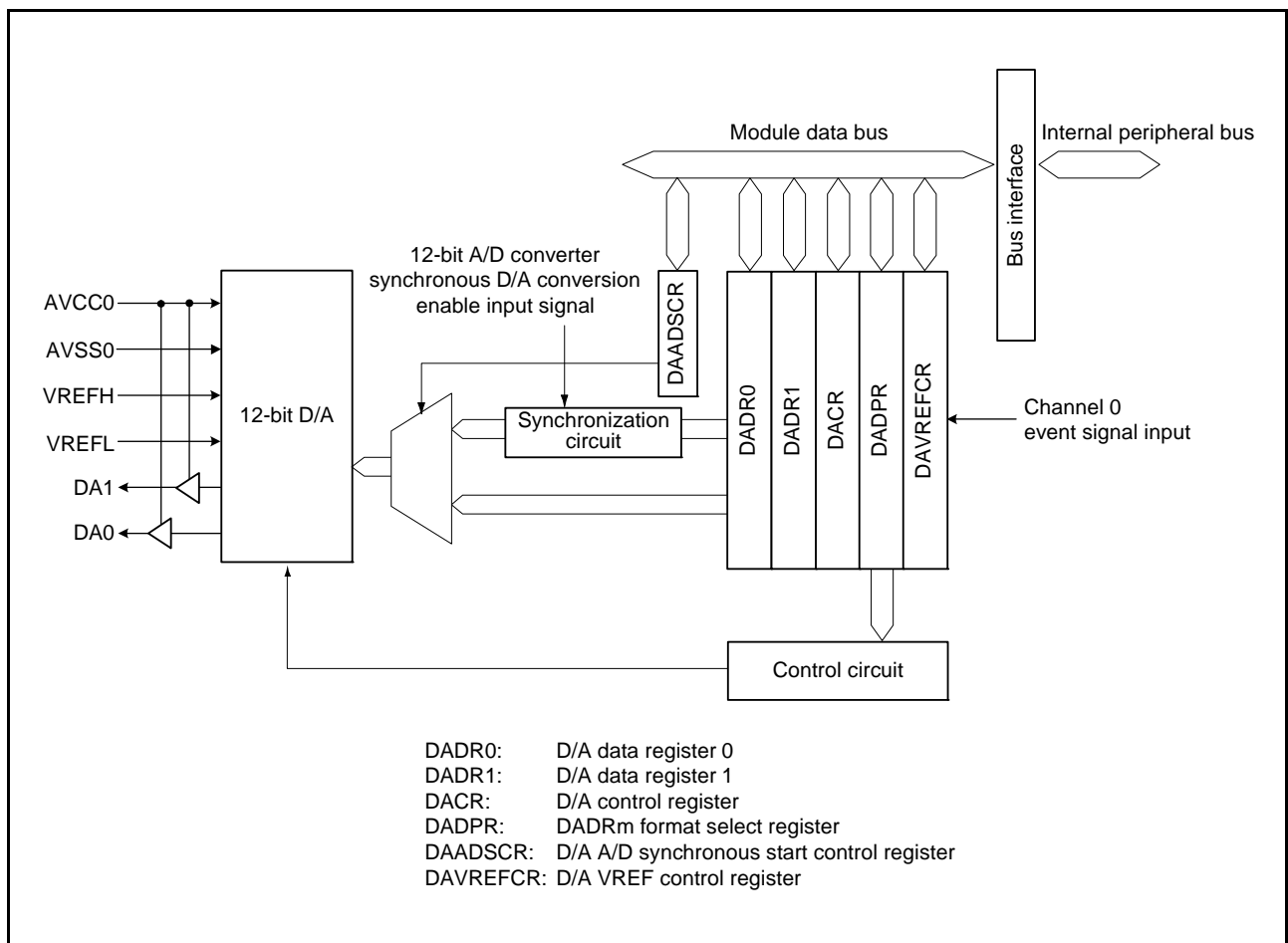
### 36.1 Overview

This MCU includes two channels of 12-bit D/A converter.

Table 36.1 lists the specifications of the 12-bit D/A converter and Figure 36.1 shows a block diagram of the 12-bit D/A converter.

**Table 36.1 Specifications of 12-Bit D/A Converter**

Item	Specifications
Resolution	12 bits
Output channels	Two channels
Countermeasure against mutual interference between analog modules	Measure against interference between D/A and A/D conversion D/A converted data update timing is controlled by the 12-bit A/D converter synchronous D/A conversion enable input signal from the 12-bit A/D converter. Therefore, degradation of A/D conversion accuracy caused by interference is reduced by controlling the 12-bit D/A converter inrush current generation timing with the enable signal.
Low power consumption function	Module stop state can be set.
Event link function (input)	DA0 conversion can be started when an event signal is input.



**Figure 36.1 Block Diagram of 12-Bit D/A Converter**



Table 36.2 lists the pin configuration of the 12-bit D/A converter.

**Table 36.2 Pin Configuration of 12-Bit D/A Converter**

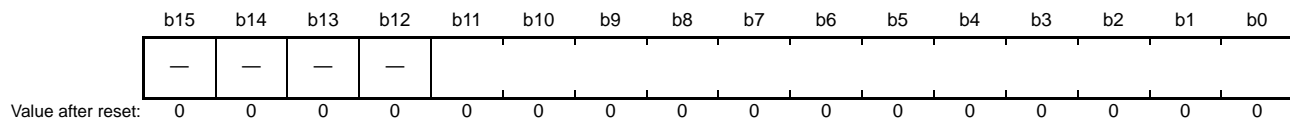
Pin Name	I/O	Function
AVCC0	Input	Analog voltage supply pin for the 12-bit A/D converter and 12-bit D/A converter. Connect this pin to VCC when not using the 12-bit A/D converter and 12-bit D/A converter.
AVSS0	Input	Analog ground pin for the 12-bit A/D converter and 12-bit D/A converter. Connect this pin to VSS when not using the 12-bit A/D converter and 12-bit D/A converter.
VREFH	Input	Analog reference voltage supply pin for the 12-bit D/A converter.
VREFL	Input	Analog reference ground pin for the 12-bit D/A converter.
DA0	Output	Channel 0 analog output pin
DA1	Output	Channel 1 analog output pin

## 36.2 Register Descriptions

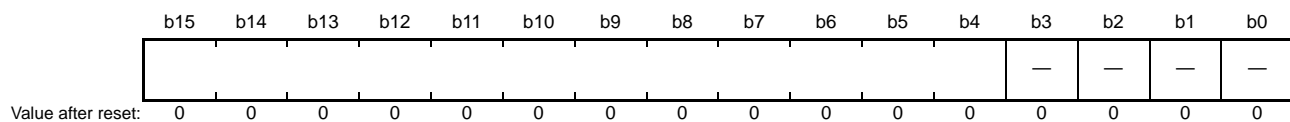
### 36.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Address(es): DADR0 0008 8040h, DADR1 0008 8042h

- DADPR.DPSEL bit = 0 (data is flush with the right end of the register)



- DADPR.DPSEL bit = 1 (data is 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.

12-bit data can be relocated by setting the DADPR.DPSEL bit.

Bits “—” are read as 0. The write value should be 0.

### 36.2.2 D/A Control Register (DACR)

Address(es): 0008 8044h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAOE1	DAOE0	—	—	—	—	—	—
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R
b6	DAOE0	D/A Output Enable 0	0: 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.	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.	R/W

This register should be set when the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled) while the 12-bit A/D converter is halted (the ADCSR.ADST bit is 0). At that time, the software trigger should be selected for the 12-bit A/D converter trigger to securely stop the 12-bit A/D converter.

#### DAOE0 Bit (D/A Output Enable 0)

The DAOE0 bit controls the D/A conversion and analog output.

The event link function can be used to set the DAOE0 bit to 1. The DAOE0 bit becomes 1 when the event specified by setting the ELSR17 register of the ELC occurs, and output of the D/A conversion results starts.

#### DAOE1 Bit (D/A Output Enable 1)

The DAOE1 bit controls the D/A conversion and analog output.

### 36.2.3 DADRm Format Select Register (DADPR)

Address(es): 0008 8045h

	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

### 36.2.4 D/A A/D Synchronous Start Control Register (DAADSCR)

Address(es): 0008 8046h

	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: 12-bit D/A converter operation does not synchronize with 12-bit A/D converter operation. (measure against interference between D/A and A/D conversion is disabled) 1: 12-bit D/A converter operation synchronizes with 12-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, the DAADSCR register switches synchronization of the timing of the start of D/A conversion with the 12-bit A/D converter synchronous D/A conversion enable input signal off or on.

This register should be set while the 12-bit A/D converter is halted (while the ADCSR.ADST bit is 0 after selecting software trigger as the 12-bit A/D converter trigger).

#### DAADST Bit (D/A A/D Synchronous Conversion)

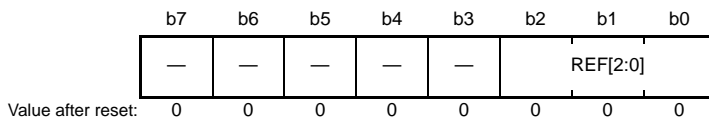
Setting the DAADST bit to 0 allows the DADRm register value to be converted into analog data at any time. Setting the DAADST bit to 1 allows synchronous D/A conversion with the 12-bit A/D converter synchronous D/A conversion enable input signal from the 12-bit A/D converter. Therefore, even if the DADRm register value is modified, D/A conversion does not start until the 12-bit A/D converter completes A/D conversion.

Set this bit while the ADCSR.ADST bit is set to 0. At this time, the software trigger should be selected for the 12-bit A/D converter trigger to securely stop the 12-bit A/D converter. Set bit 1 in the DAADUSR register to 1 before setting the DAADST bit to 1.

The event link function cannot be used when the DAADST bit is set to 1. Stop the event link function by setting the ELSR17 register of the ELC. The setting of the DAADST bit is common to channels 0 and 1 of the 12-bit D/A converter.

### 36.2.5 D/A VREF Control Register (DAVREFCR)

Address(es): 0008 8047h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	REF[2:0]	D/A Reference Voltage Select	b2 b0 0 0 0: Not selected 0 0 1: AVCC0/AVSS0 0 1 1: Internal reference voltage/AVSS0 1 1 0: VREFH/VREFL Settings other than above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DAVREFCR register selects the reference voltage of the 12-bit D/A converter.

#### REF[2:0]Bits (D/A Reference Voltage Select)

The REF[2:0] bits select the reference voltage of the 12-bit D/A converter 0 and 1. When changing the value of these bits, write 000h to the DAVREFCR.REF[2:0] bits in advance. Read the REF[2:0] bits after changing their value, and confirm that it has been changed. When selecting the internal reference voltage, set the DADR0 and DADR1 registers to 0000h and discharge the VREF path before switching the voltage. As the path remains discharged after the reset is released, the internal reference voltage can be selected. For details on discharging, refer to section 36.3.2, Notes on Using the Internal Reference Voltage as the Reference Voltage. Do not rewrite this register during A/D conversion using the 12-bit A/D converter. If this register is rewritten, the accuracy of A/D conversion is not guaranteed.

When the internal reference voltage is selected, the voltage generation circuit operates and current increases by about 75  $\mu$ A. This circuit is not automatically turned off even if the MCU enters software standby mode with the internal reference voltage selected.

### 36.3 Operation

The 12-bit D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DAOEn bit (n = 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 36.2 shows the timing of this operation.

1. Set the data for D/A conversion in the DADPR.DPSEL bit and the DADR0 register.
2. Set the DACR.DAOE0 bit to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time tDCONV 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 (reference) is expressed by the following formula:

$$\frac{\text{Setting value of DADRm}}{4096} \times \text{Reference voltage}$$

3. If DADR0 is written to again, the conversion is started. The conversion result is output after the conversion time tDCONV 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. When ADCLK is faster than the peripheral clock, it may take longer than one A/D conversion time.

4. If the DAOE0 bit is set to 0, analog output is disabled.

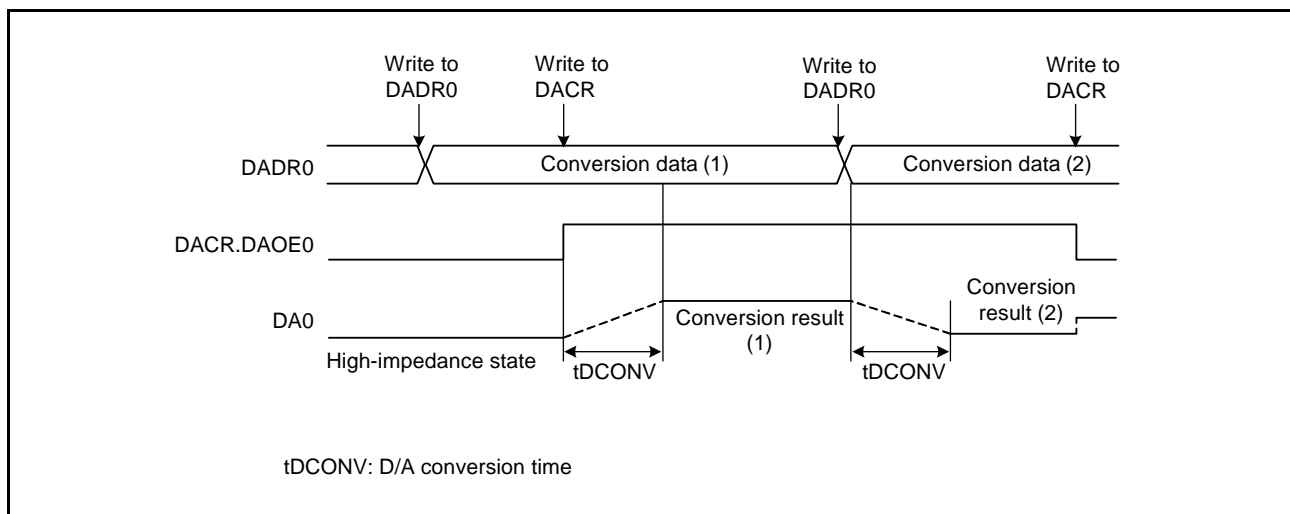


Figure 36.2 Example of 12-Bit D/A Converter Operation

### 36.3.1 Measure against Interference between D/A and A/D Conversion

When D/A conversion starts, the 12-bit D/A converter generates inrush current. Since the same analog power supply is shared by the 12-bit D/A converter and 12-bit A/D converter, the generated inrush current may interfere with 12-bit A/D converter operation.

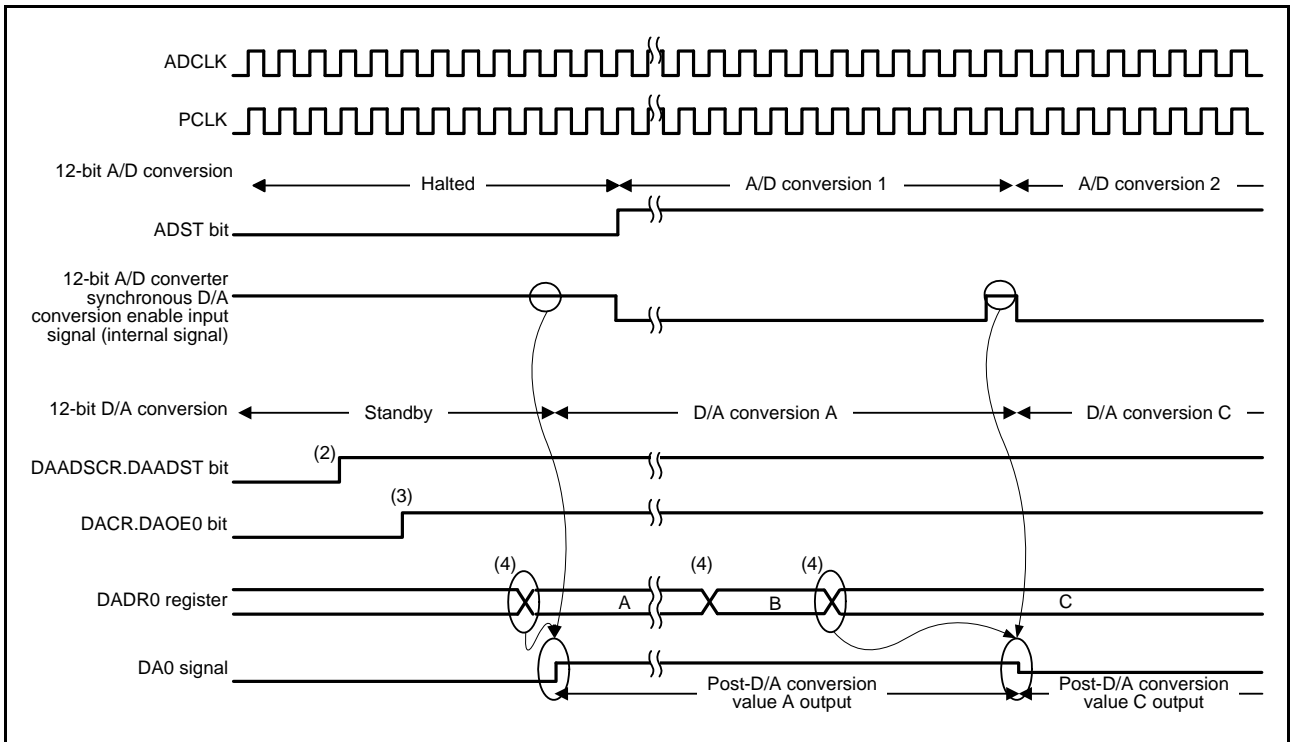
With the DAADSCR.DAADST bit being 1, even if the DADR<sub>m</sub> register data is modified during 12-bit A/D converter operation, D/A conversion does not start immediately but starts synchronously with A/D conversion completion. It takes a maximum of one A/D conversion time for the DADR<sub>m</sub> register data update to be reflected as the D/A conversion circuit input. Before reflection, the DADR<sub>m</sub> 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 DADR<sub>m</sub> register value has been D/A converted or not.

Even with DAADSCR.DAADST being 1, when the DADR<sub>m</sub> register data is modified while the 12-bit A/D converter is halted, D/A conversion starts in one PCLK cycle.

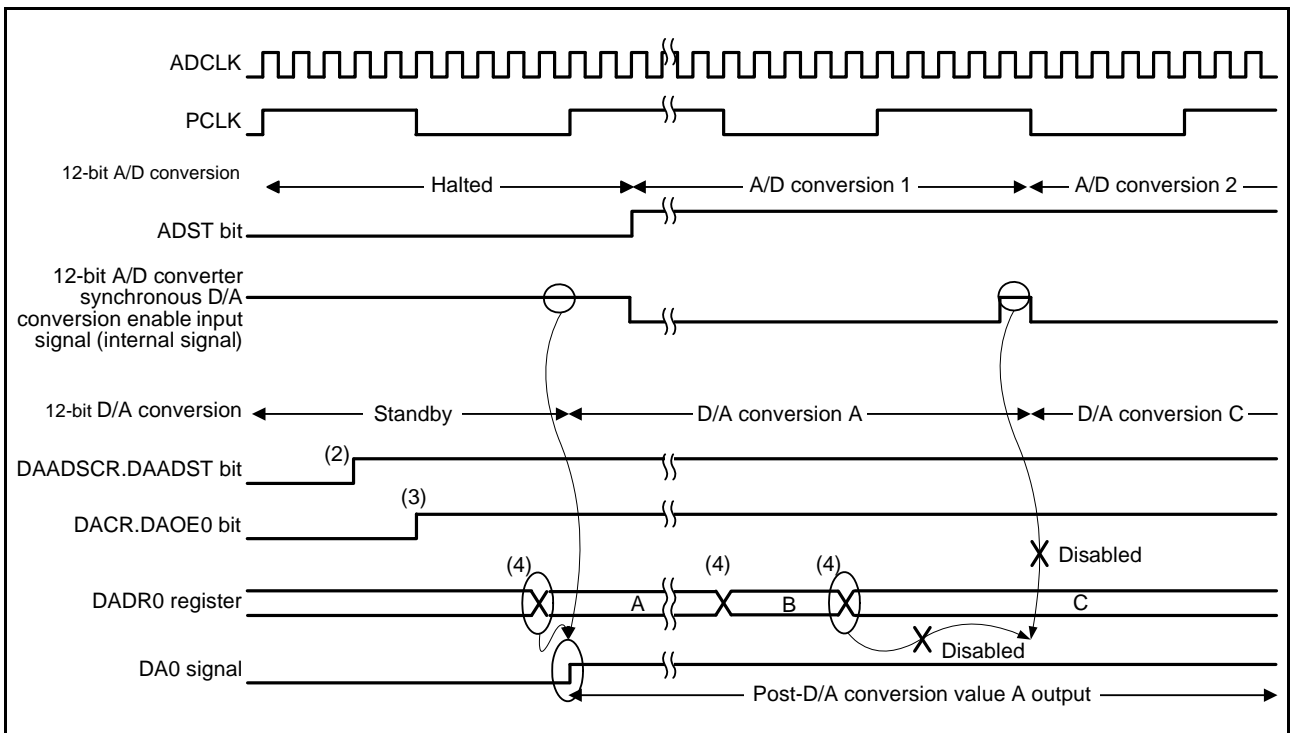
Figure 36.3 shows an example of channel 0 D/A conversion, in which the 12-bit D/A converter operates synchronously with the 12-bit A/D converter.

- (1) Confirm that the 12-bit A/D converter is halted. Set the DAADUSR.AMASEL1 bit to 1.
- (2) Confirm that the 12-bit A/D converter is halted. Set the DAADSCR.DAADST bit to 1.
- (3) Confirm that the 12-bit A/D converter is halted. Set the DACR.DAOE0 bit to 1.
- (4) Set the DADR0 register. When ADCLK is faster than the peripheral clock, it may take longer than one A/D conversion time for D/A conversion to start.
  - If the 12-bit A/D conversion is halted (ADCSR.ADST bit = 0) when the DADR0 register is modified, D/A conversion starts in one PCLK cycle.
  - If the 12-bit A/D conversion is in progress (ADCSR.ADST bit = 1) 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 36.3 Example of Conversion When the 12-Bit D/A Converter is Synchronized with the 12-Bit A/D Converter**

When ADCLK is faster than PCLK, the DA may not be able to capture a 12-bit A/D converter synchronous D/A conversion enable input signal for one ADCLK cycle which is output between A/D conversion 1 and A/D conversion 2. Figure 36.4 shows example when the 12-bit D/A converter cannot capture the 12-bit A/D converter synchronous D/A conversion enable input signal. In this case, post-D/A conversion value A is continuously output as the DA0 signal.



**Figure 36.4 Example When the 12-Bit D/A Converter Cannot Capture the 12-Bit A/D Converter Synchronous D/A Conversion Enable Input Signal**

### 36.3.2 Notes on Using the Internal Reference Voltage as the Reference Voltage

When setting the DAVREFCR.REF[2:0] bits to 011b to use the internal reference voltage/AVSS0 as the reference voltage, the VREF path needs to be discharged before selecting the voltage. The following shows the discharging procedure.

- (1) Write 000h to the REF[2:0] bits.
- (2) Set the DADR0 register to 0000h and the DADR1 register to 0000h.
- (3) Keep the state of step (2) for 10  $\mu$ s (discharging).
- (4) After discharging is completed, write 011b to the DAVREFCR.REF[2:0] bits and select the internal reference voltage/AVSS0.
- (5) Set the DACR.DAOEn bit to 1 and wait for the stabilization wait time (5  $\mu$ s) of the internal reference voltage.
- (6) Write data to the DADRm register and start D/A conversion.

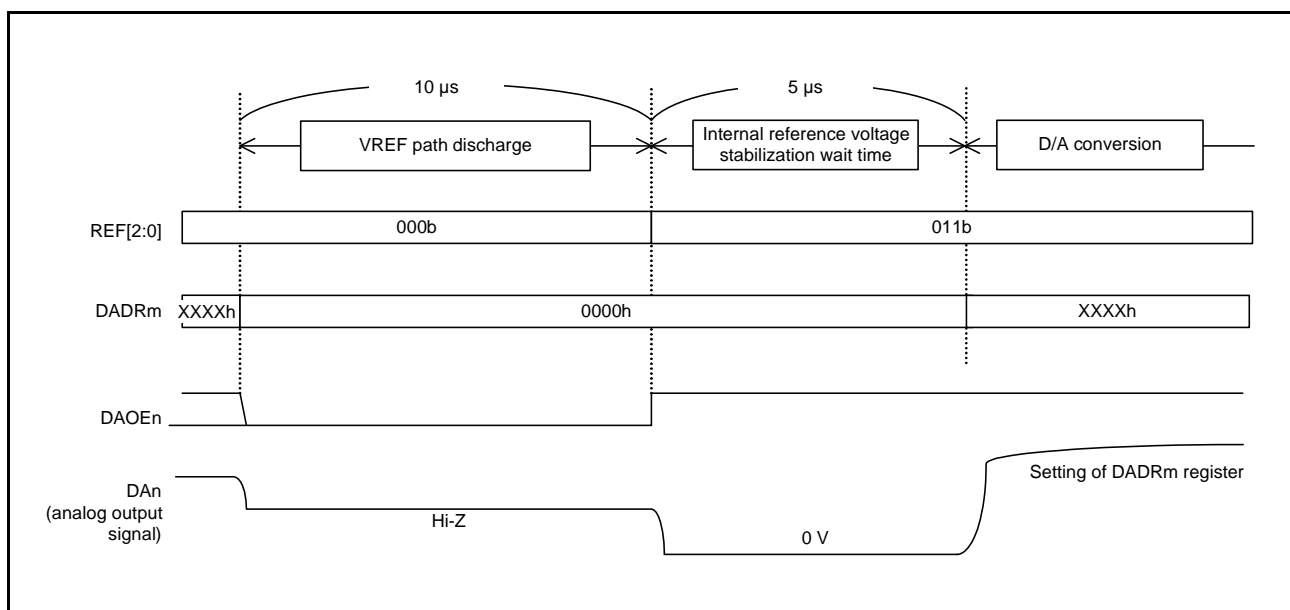


Figure 36.5 Procedure for Selecting the Internal Reference Voltage as the Reference Voltage



### 36.4 Event Link Operation Setting Procedure

The event link operation procedure is described below.

1. Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR0 register.
2. Set the bit value of the ELSR17 setting event signal to link the ELSR17 register of the ELC.
3. Set the ELCR.ELCON bit to 1. This procedure enables event link operation for all modules with the event link function selected.
4. Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE0 bit becomes 1, and D/A conversion on channel 0 starts.
5. Set the ELSR17.ELS[7:0] bits to 0000 0000b to stop event link operation of 12-bit D/A converter channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

### 36.5 Usage Notes on Event Link Operation

1. When the event specified by the ELSR17 register is generated while the write cycle is performed to the DACR.DAOE0 bit, the write cycle is stopped, and the setting to 1 by the generated event takes precedence.
2. Use of the event link function is prohibited when the DAADSCR.DAADST bit is set to 1 as the countermeasure against an interfere between D/A and A/D conversions.

## 36.6 Usage Notes

### 36.6.1 Module Stop Function Setting

Operation of the 12-bit D/A converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 12-bit D/A converter to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

### 36.6.2 Operation of the D/A Converter in Module Stop State

When the MCU enters the module stop state with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in the module stop state, disable D/A conversion by setting the DACR.DAOE1, and DAOE0 bits to 0.

### 36.6.3 Operation of the D/A Converter in Software Standby Mode

When the MCU enters software standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in software standby mode, disable D/A conversion by setting the DACR.DAOE1, and DAOE0 bits to 0.

### 36.6.4 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 12-bit A/D converter in the module stop state. It may halt D/A conversion in addition to A/D conversion.

### 36.6.5 Voltage Relationship between AVCC0 and VCC

AVCC0 and VCC can be set individually, but the following restrictions apply.

- When the PJ0 and PJ2 pins are used as D/A output pins, make sure that the D/A output voltage does not exceed the VCC.
- When the PJ0 and PJ2 pins are used as I/O ports, make sure that the VCC voltage does not exceed the AVCC0 voltage.
- When the PJ0 and PJ2 pins are not used, these restrictions do not apply.

## 37. Temperature Sensor (TEMPSA)

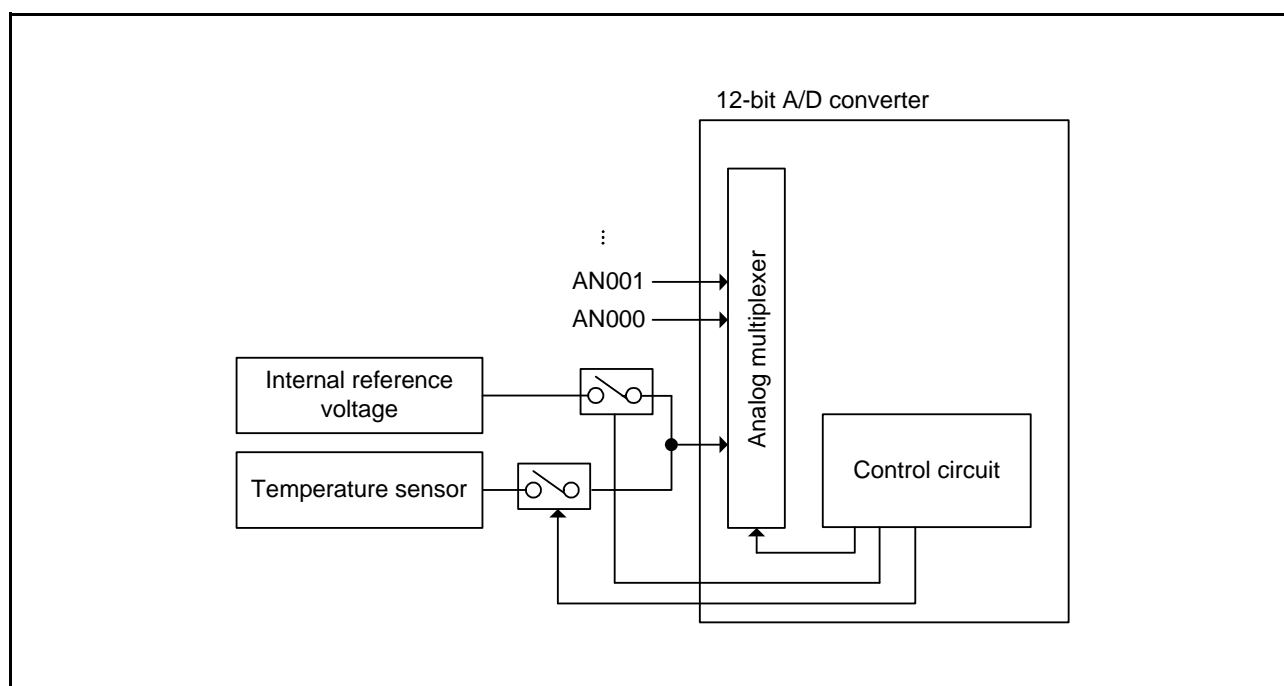
### 37.1 Overview

This MCU includes a temperature sensor. The temperature sensor outputs a voltage which varies with the temperature. The user can obtain the temperature surrounding the MCU using the 12-bit A/D converter to convert the voltage output from the temperature sensor into a digital value.

Table 37.1 lists the specifications of the temperature sensor. Figure 37.1 shows a overall block diagram of the temperature sensor system.

**Table 37.1 Temperature Sensor Specifications**

Item	Description
Temperature sensor voltage output	The temperature sensor voltage is output to the 12-bit A/D converter.

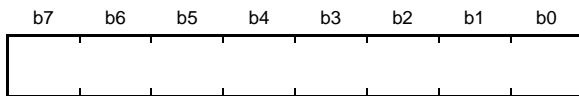


**Figure 37.1 Block Diagram of Temperature Sensor System**

## 37.2 Register Descriptions

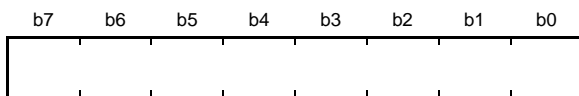
### 37.2.1 Temperature Sensor Calibration Data Register (TSCDRH, TSCDRL)

Address(es): TSCDRL 007F C0ACh



Value after reset: Unique value for each chip

Address(es): TSCDRH 007F C0ADh



Value after reset: Unique value for each chip

The TSCDRH and TSCDRL registers store temperature sensor calibration data measured for each chip at factory shipment.

Temperature sensor calibration data is a digital value obtained using the 12-bit A/D converter to convert the voltage output by the temperature sensor under the condition  $T_a = T_j = 88^\circ\text{C}$  and  $AVCC0 = VREFH0 = 3.3\text{ V}$ . The TSCDRH register stores the higher 4 bits of the converted value, and the TSCDRL register stores the lower 8 bits.

### 37.3 Using the Temperature Sensor

The temperature sensor outputs a voltage which varies with the temperature. The user can obtain the temperature surrounding the MCU using the 12-bit A/D converter to convert this voltage into a digital value.

#### 37.3.1 Before Using the Temperature Sensor

The temperature characteristics of the temperature sensor are shown below. The voltage output by the temperature sensor is proportional to the temperature, which can be calculated according to the formula below.

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 when the temperature is measured (V)

T<sub>1</sub>: Sample temperature measurement at first point (°C)

V<sub>1</sub>: Voltage output by the temperature sensor when T<sub>1</sub> is measured (V)

T<sub>2</sub>: Sample temperature measurement at second point (°C)

V<sub>2</sub>: Voltage output by the temperature sensor when T<sub>2</sub> is measured (V)

$(V_2 - V_1)/(T_2 - T_1)$  = Slope: Temperature gradient of the temperature sensor (V/°C)

Characteristics vary from sensor to sensor. Therefore, it is recommended that two different sample temperatures are measured.

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 =  $(V_2 - V_1)/(T_2 - T_1)$ ) from these results.

Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic ( $T = (V_s - V_1)/\text{Slope} + T_1$ ).

If you are using the temperature gradient given in section 42, Electrical Characteristics, use the 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 produces less accurate temperatures than measurement at two points.

$$T = (V_s - V_1)/\text{Slope} + T_1$$

In this MCU, the TSCDRH and TSCDRL registers store the temperature value (CAL<sub>88</sub>) of the temperature sensor measured under the condition T<sub>a</sub> = T<sub>j</sub> = 88°C and AVCC0 = VREFH0 = 3.3 V. By using this value as the sample measurement result at the first point, preparation before using the temperature sensor can be omitted.

This measured value CAL<sub>88</sub> can be calculated as follows:

$$\text{CAL}_{88} = \text{TSCDRH register value} \ll 8 + \text{TSCDRL register value}$$

If V<sub>1</sub> is calculated from CAL<sub>88</sub>,

$$V_1 = 3.3 \times \text{CAL}_{88}/4096 \text{ [V]}$$

Using this, the measured temperature can be calculated according to the formula below.

$$T = (V_s - V_1)/\text{Slope} + 88 \text{ [°C]}$$

T: Measured temperature (°C)

Vs: Voltage output by the temperature sensor when the temperature is measured (V)

V1: Voltage output by the temperature sensor when  $T_a = T_j = 88^\circ\text{C}$  and  $AVCC0 = VREFH0 = 3.3\text{ V}$  (V)

Slope: Temperature gradient listed in Table 42.46  $\div 1000$  (V/°C)

Error in the measured temperature (variation range is  $3\sigma$ ) is shown in Figure 37.2.

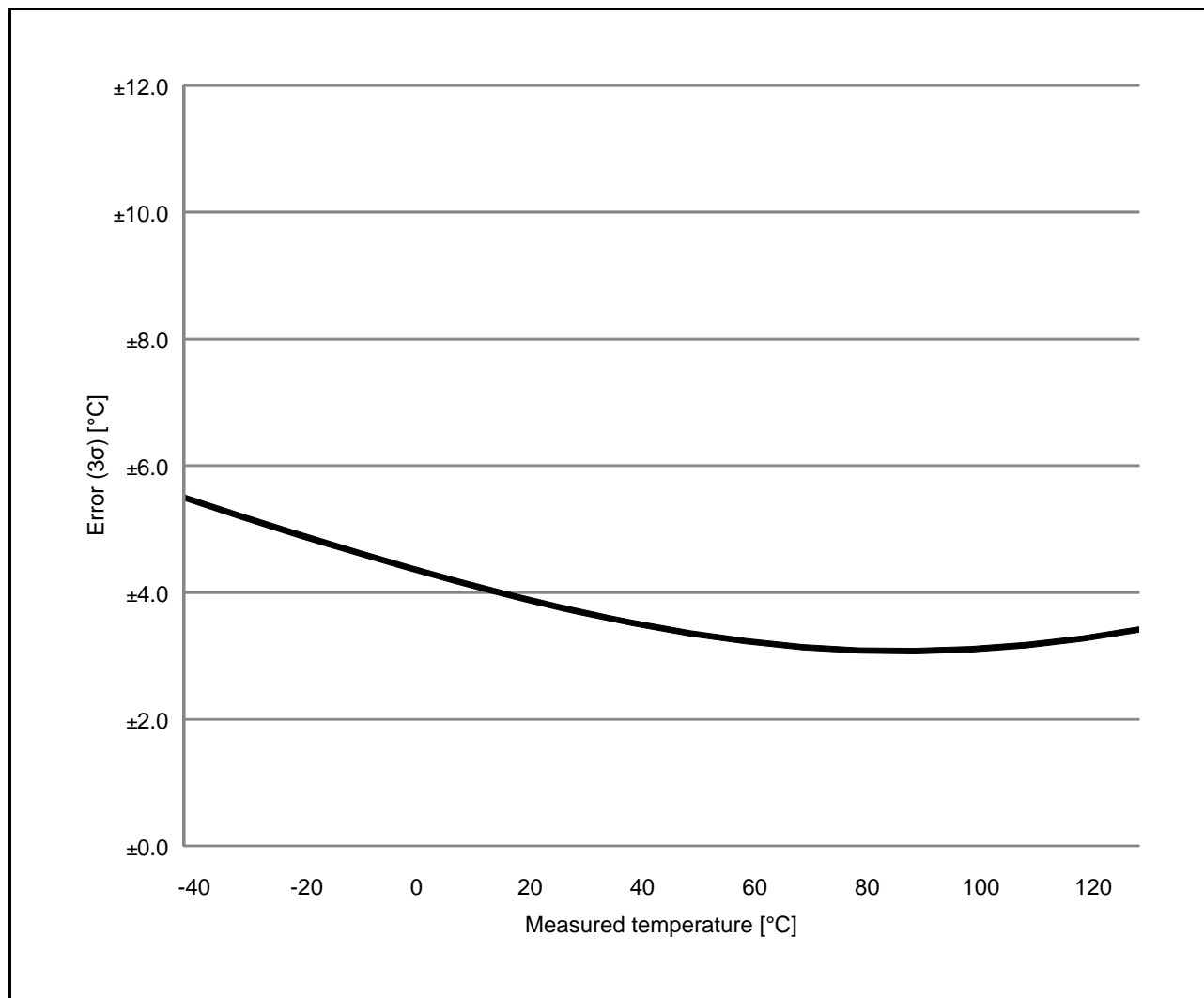


Figure 37.2 Error in the Measured Temperature (Designed Value)

### 37.3.2 Setting the 12-Bit A/D Converter

The temperature sensor can provide temperature data through the A/D conversion of the temperature sensor output. In order to A/D convert output from the temperature sensor, 12-bit A/D converter registers must be set as follows.

- Selecting the temperature sensor voltage as an A/D conversion target  
Set the ADEXICR.TSS bit to 1 to A/D convert temperature sensor output. Additionally, remove other sources from the scope of conversion by setting all the bits in registers ADANSA and ADANSB to 0.  
When comparator B (CMPB) is set to use the internal reference voltage, do not set the ADEXICR.TSS bit to 1.
- Setting single scan mode  
Set the ADCSR.ADCS[1:0] bits to 00b to select single scan mode. Do not select the other mode.

### 37.3.3 A/D Conversion Result of Temperature Sensor Output

After the temperature sensor output is A/D converted, the conversion result is stored in the ADTSDR register. Set the sampling time to 5  $\mu$ s or longer. After switching to A/D conversion of the temperature sensor output, set the ADST bit to 1 and start the first conversion. However, do not use the first conversion result. Figure 37.3 shows an example of operating the temperature sensor.

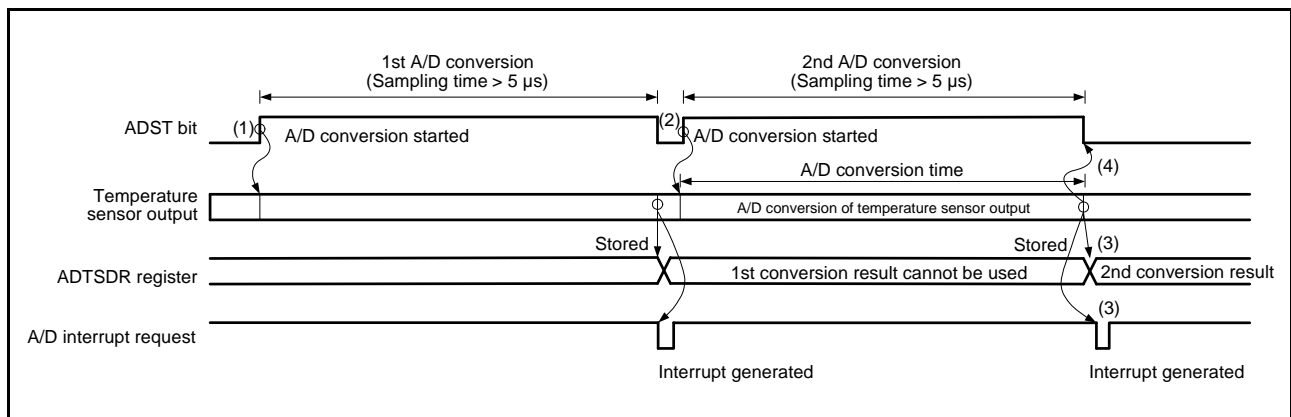


Figure 37.3 Example of Temperature Sensor Operation

## 38. Comparator B (CMPBa)

Comparator B compares a reference input voltage and an analog input voltage. Comparator B0 and comparator B1 operate independently.

In this section, “PCLK” is used to refer to PCLKB.

### 38.1 Overview

The comparison result of the reference input voltage and analog input voltage can be read by software. The comparison result can also be output externally. The reference input voltage can be selected from either an input to the CVREFBn (n = 0, 1) pin or the internal reference voltage (1.44 V) generated internally in the MCU.

The comparator B response speed can be set before starting an operation. Setting high-speed mode decreases the response delay time, but increases current consumption. Setting low-speed mode increases the response delay time, but decreases current consumption.

Table 38.1 lists the specifications of comparator B, Figure 38.1 shows a block diagram of comparator B when the window function is disabled, and Figure 38.2 shows a block diagram of comparator B when the window function is enabled. Table 38.2 lists the I/O pins of comparator B.

**Table 38.1 Comparator B Specifications**

Item	Specification
Analog input voltage	Input voltage to the CMPBn pin (n = 0, 1)
Reference input voltage	Input voltage to the CVREFBn pin (n = 0, 1) or internal reference voltage
Comparison result	Read from the CPBFLG.CPBnOUT flag (n = 0, 1) The comparison result can be output to the CMPOBn pin (n = 0, 1).
Interrupt request generation timing	When comparator B0 comparison result changes When comparator B1 comparison result changes
Event generation timing to ELC	When comparator B0 comparison result changes When comparator B0 or comparator B1 comparison result changes
Selectable function	<ul style="list-style-type: none"> <li>• Digital filter function Whether the digital filter is applied or not, and the sampling frequency can be selected.</li> <li>• Window function Whether the window function is enabled or disabled (low-side reference (VRFL) &lt; CMPBn (n = 0, 1) &lt; high-side reference (VRFH)) can be selected.</li> <li>• Reference input voltage CVREFBn pin input or internal reference voltage (generated internally) can be selected (n = 0, 1).</li> <li>• Comparator B response speed High-speed mode/low-speed mode can be selected.</li> </ul>
Low power consumption function	Module stop state can be set.



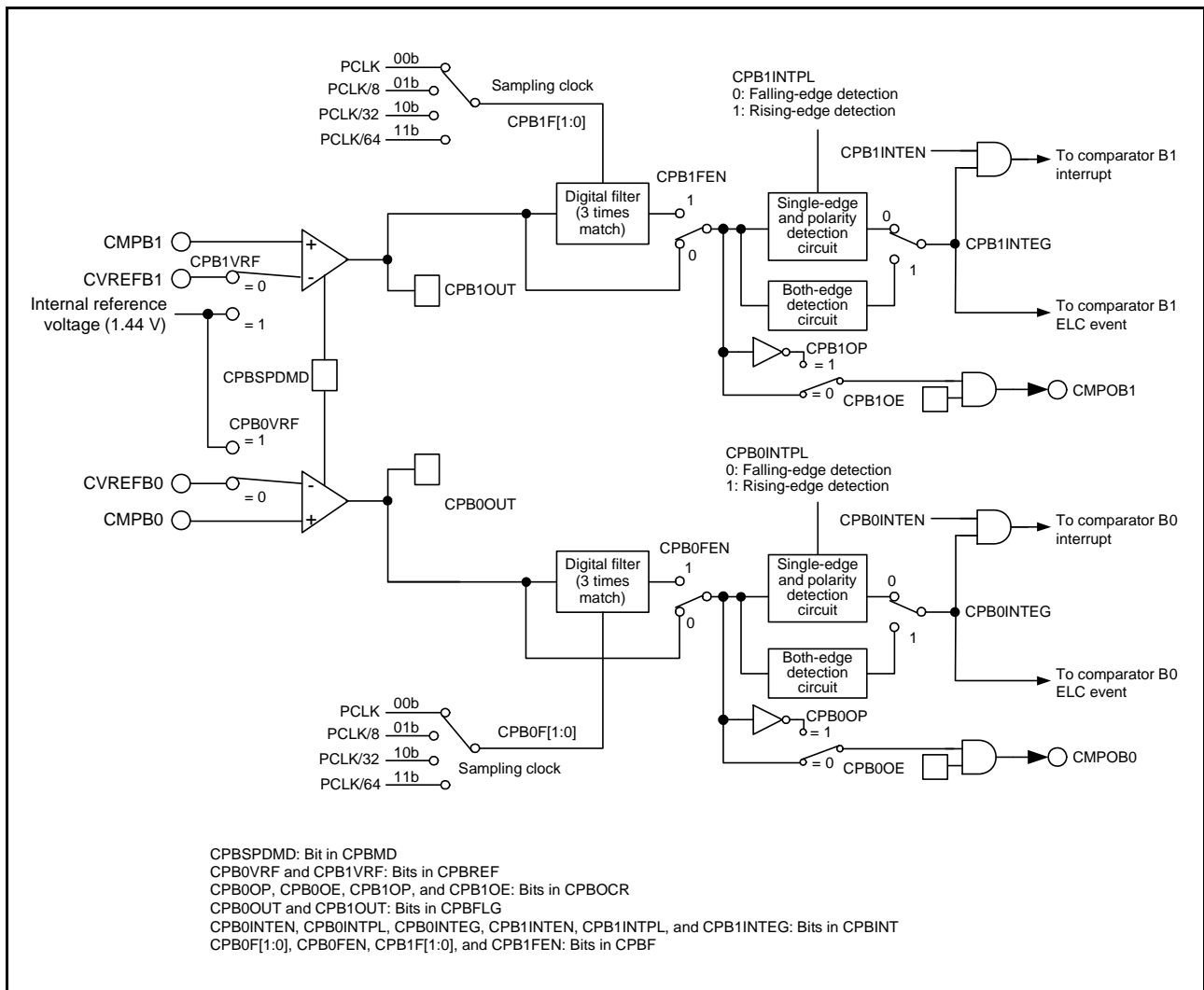


Figure 38.1 Block Diagram of Comparator B When Window Function is Disabled

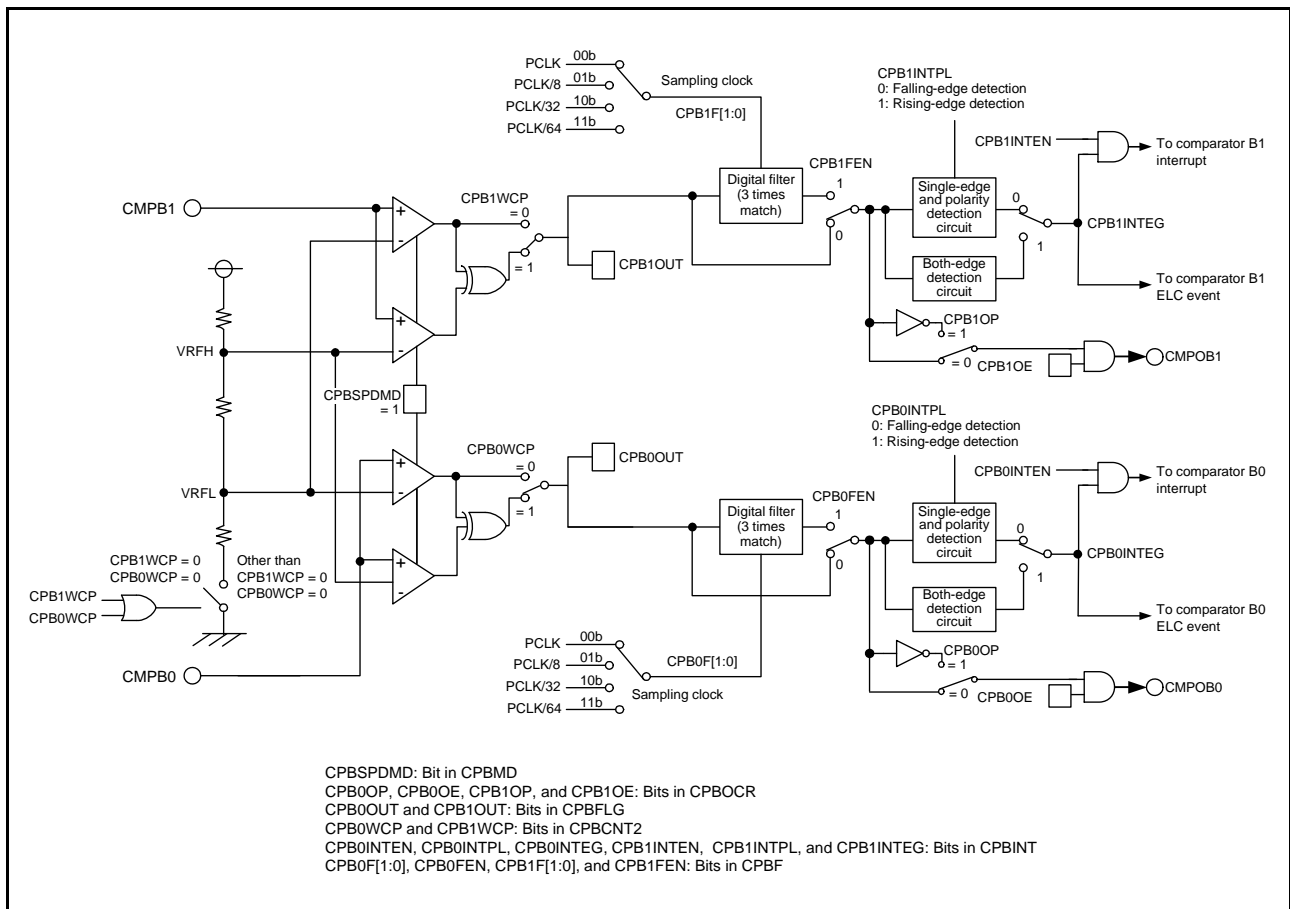


Figure 38.2 Block Diagram of Comparator B When Window Function is Enabled

Table 38.2 I/O Pins of Comparator B

Pin Name	I/O	Function
CMPB0	Input	Comparator B0 analog pin
CVREFB0	Input	Comparator B0 reference input voltage pin
CMPB1	Input	Comparator B1 analog pin
CVREFB1	Input	Comparator B1 reference input voltage pin
CMPOB0	Output	Comparator B0 output
CMPOB1	Output	Comparator B1 output

## 38.2 Register Descriptions

### 38.2.1 Comparator B Control Register 1 (CPBCNT1)

Address: 0008 C580h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	CPB1I NI	—	—	—	CPB0I NI

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

Bit	Symbol	Bit Name	Description	R/W
b0	CPB0INI	Comparator B0 Power Enable	0: Disabled 1: Enabled (comparator powered on)	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CPB1INI	Comparator B1 Power Enable	0: Disabled 1: Enabled (comparator powered on)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

### 38.2.2 Comparator B Control Register 2 (CPBCNT2)

Address: 0008 C581h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	CPB1W CP	—	—	—	CPB0W CP

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

Bit	Symbol	Bit Name	Description	R/W
b0	CPB0WCP	Comparator B0 Window Function Enable	0: Disabled 1: Enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CPB1WCP	Comparator B1 Window Function Enable	0: Disabled 1: Enabled	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

### 38.2.3 Comparator B Flag Register (CPBFLG)

Address: 0008 C582h

b7	b6	b5	b4	b3	b2	b1	b0
CPB1OUT	—	—	—	CPB0OUT	—	—	—

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	CPB0OUT	Comparator B0 Monitor Flag	When the window function is disabled 0: CMPB0 < CVREFB0, CMPB0 < internal reference voltage, or comparator B0 operation disabled 1: CMPB0 > CVREFB0, or CMPB0 > internal reference voltage When the window function is enabled 0: CMPB0 < low-side reference (VRFL), CMPB0 > high-side reference (VRFH), or comparator B0 operation disabled 1: Low-side reference (VRFL) < CMPB0 < high-side reference (VRFH)	R
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CPB1OUT	Comparator B1 Monitor Flag	When the window function is disabled 0: CMPB1 < CVREFB1, CMPB1 < internal reference voltage, or comparator B1 operation disabled 1: CMPB1 > CVREFB1, or CMPB1 > internal reference voltage When the window function is enabled 0: CMPB1 < low-side reference (VRFL), CMPB1 > high-side reference (VRFH), or comparator B1 operation disabled 1: Low-side reference (VRFL) < CMPB1 < high-side reference (VRFH)	R

### 38.2.4 Comparator B Interrupt Control Register (CPBINT)

Address: 0008 C583h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CPB1 NTPL	CPB1 NTEG	CPB1 NTEN	—	CPB0 NTPL	CPB0 NTEG	CPB0 NTEN
Value after reset:	0	0	0	0	0	0	0	0

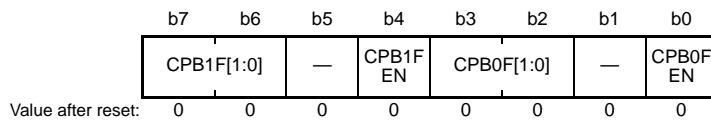
Bit	Symbol	Bit Name	Description	R/W
b0	CPB0INTEN	Comparator B0 Interrupt Enable	0: Disabled 1: Enabled	R/W
b1	CPB0INTEG	Comparator B0 Interrupt/ELC Edge Select*1	0: Single edge 1: Both edges	R/W
b2	CPB0INTPL	Comparator B0 Interrupt/ELC Edge Polarity Select*2	0: Falling edge 1: Rising edge	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	CPB1INTEN	Comparator B1 Interrupt Enable	0: Disabled 1: Enabled	R/W
b5	CPB1INTEG	Comparator B1 Interrupt/ELC Edge Select*1	0: Single edge 1: Both edges	R/W
b6	CPB1INTPL	Comparator B1 Interrupt/ELC Edge Polarity Select*2	0: Falling edge 1: Rising edge	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The IR058.IR bit may become 1 (interrupt request is generated) when the CPB0INTPL bit is modified, and the IR059.IR bit may become 1 (interrupt request is generated) when the CPB1INTPL bit is modified. For details, refer to section 14, Interrupt Controller (ICUb).

Note 2. The CPBnINTPL bit setting is valid only when the CPBnINTEG bit is 0 (single edge is selected as the comparator interrupt edge).

### 38.2.5 Comparator B Filter Select Register (CPBF)

Address: 0008 C584h

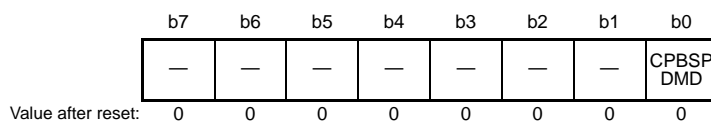


Bit	Symbol	Bit Name	Description	R/W
b0	CPB0FEN	Comparator B0 Filter Enable/Disable Select* <sup>1</sup>	0: Filter is disabled. 1: Filter is enabled.	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	RW
b3, b2	CPB0F[1:0]	Comparator B0 Filter Select* <sup>1</sup>	b3 b2 0 0: Sampling at PCLK 0 1: Sampling at PCLK/8 1 0: Sampling at PCLK/32 1 1: Sampling at PCLK/64	R/W
b4	CPB1FEN	Comparator B1 Filter Enable/Disable Select* <sup>1</sup>	0: Filter is disabled. 1: Filter is enabled.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	RW
b7, b6	CPB1F[1:0]	Comparator B1 Filter Select* <sup>1</sup>	b7 b6 0 0: Sampling at PCLK 0 1: Sampling at PCLK/8 1 0: Sampling at PCLK/32 1 1: Sampling at PCLK/64	R/W

Note 1. The CPBnF[1:0] bits are enabled only when the CPBnFEN bit = 1 (filter is enabled).

### 38.2.6 Comparator B Mode Select Register (CPBMD)

Address: 0008 C585h

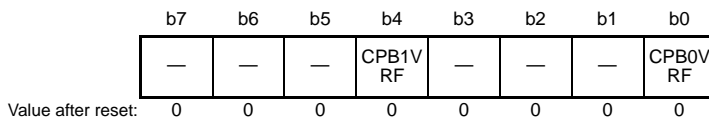


Bit	Symbol	Bit Name	Description	R/W
b0	CPBSPDMD	Comparator B Speed Select	0: High-speed mode 1: Low-speed mode* <sup>1</sup>	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	RW

Note 1. When rewriting the CPBSPDMD bit, be sure to set the CPBnINI bit (n = 0, 1) in the CPBCNT1 register to 0 in advance.

### 38.2.7 Comparator B Reference Input Voltage Select Register (CPBREF)

Address: 0008 C586h



Bit	Symbol	Bit Name	Description	R/W
b0	CPB0VRF	Comparator B0 Reference Input Voltage Select	0: Comparator B0 reference input voltage is CVREFB0 input 1: Comparator B0 reference input voltage is internal reference voltage*1, *2, *3	R/W*4
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	RW
b4	CPB1VRF	Comparator B1 Reference Input Voltage Select	0: Comparator B1 reference input voltage is CVREFB1 input 1: Comparator B1 reference input voltage is internal reference voltage*1, *2, *3	R/W*4
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	RW

Note 1. Enabled only when the window function is disabled. When the window function is enabled, the internal reference voltage of comparator B is selected regardless of the setting of this bit.

Note 2. When the internal reference voltage is selected, the temperature sensor output cannot be selected for the A/D converter.

Note 3. When the internal reference voltage is selected, the voltage generation circuit operates and current increases by about 75  $\mu$ A. This circuit is not automatically turned off even if the MCU enters software standby mode with the internal reference voltage selected.

Note 4. Do not rewrite the CPBnVRF bit when CPBCNT2.CPBnWCP = 0.

[Notes on changing the reference input voltage]

◆ When changing the reference input voltage from CVREFBn (n = 0, 1) to the internal reference voltage, use the following procedure.

1. Set the CPBCNT1.CPBnINI bit to 1.
2. Set the CPBCNT2.CPBnWCP bit to 1.
3. Set the CPBREF.CPBnVRF bit to 1 to select the internal reference voltage.
4. Set the analog select bit (ASEL) in the pin function control register of the port that is used as the CVREFBn pin to 0.
5. Wait for the comparator stabilization time (min. 100  $\mu$ s).
6. Set the CPBCNT2.CPBnWCP bit to 0.

◆ When changing the reference input voltage from the internal reference voltage to CVREFBn (n = 0, 1), use the following procedure.

1. Set the CPBCNT1.CPBnINI bit to 1.
2. Set the CPBCNT2.CPBnWCP bit to 1.
3. Set the CPBREF.CPBnVRF bit to 0 to select the CVREFBn pin input.
4. Set the analog select bit (ASEL) in the pin function control register of the port that is used as the CVREFBn pin to 1.
5. Wait for the comparator stabilization time (min. 100  $\mu$ s).
6. Set the CPBCNT2.CPBnWCP bit to 0.

### 38.2.8 Comparator B Output Control Register (CPBOCR)

Address: 0008 C587h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	CPB1OP	CPB1OE	—	—	CPB0OP	CPB0OE
0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b0	CPB0OE	CMPOB0 Pin Output Enable	0: Comparator B0 CMPOB0 pin output disabled*1 1: Comparator B0 CMPOB0 pin output enabled	R/W
b1	CPB0OP	CMPOB0 Output Polarity Select	0: Comparator B0 output is output to CMPOB0 1: Inverted comparator B0 output is output to CMPOB0	RW
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	RW
b4	CPB1OE	CMPOB1 Pin Output Enable	0: Comparator B1 CMPOB1 pin output disabled*1 1: Comparator B1 CMPOB1 pin output enabled	RW
b5	CPB1OP	CMPOB1 Output Polarity Select	0: Comparator B1 output is output to CMPOB1 1: Inverted comparator B1 output is output to CMPOB1	RW
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	RW

Note 1. When the CPBnOE bit (n = 0, 1) is set to 0 to disable the CMPOBn (n = 0, 1) pin output, 0 is output to CMPOBn (n = 0, 1) regardless of the value of the CPBnOP bit (n = 0, 1).



### 38.3 Operation

Comparator B0 and comparator B1 operate independently, and their operations are the same. Operation is not guaranteed when the values of registers are changed during comparator operation. Table 38.3 shows the procedure of setting comparator B associated registers when the window function is disabled. Table 38.4 shows the procedure of setting comparator B associated registers when the window function is enabled.

**Table 38.3 Procedure for Setting Registers Associated with Comparator B When Window Function is Disabled**

Step No.	Register	Bit	Setting
1	P**PFS of the port to which the CMPBn pin is assigned	ASEL	1
2	CPBMD	CPBSPDMD	Select the comparator response speed (0: High-speed mode/1: Low-speed mode)
3	CPBCNT1	CPBnINI (n = 0, 1)	Powered on: 1
4	CPBCNT2	CPBnWCP (n = 0, 1)	1*1
5	CPBREF	CPBnVRF (n = 0, 1)	0: Reference input voltage = CVREFBn input*1 1: Reference input voltage = Internal reference voltage
6	P**PFS of the port to which the CVREFBn pin is assigned	ASEL	1 0
7	Waiting for the comparator stabilization time (min. 100 μs)*1		
8	CPBCNT2	CPBnWCP (n = 0, 1)	0*1
9	CPBF	Select whether to enable or disable the filter and select the sampling clock.	
10	Waiting for the comparator stabilization time (min. 100 μs)		
11	CPBOCR	CPBnOP, CPBnOE (n = 0, 1)	Set the CMPOBn output (select the polarity and set output enabled or disabled).
12	CPBINT	CPBnINTEN (n = 0, 1)	When using an interrupt: 1 (interrupt enabled)
		CPBnINTEG (n = 0, 1)	When using an interrupt or the ELC: Select the input edge (1 = both edges or 0 = single edge).
		CPBnINTPL (n = 0, 1)	When using an interrupt or the ELC: For CPBnINTEG = 0 (single edge selected), select the input polarity (1 = rising edge or 0 = falling edge).
13	IPR058 (comparator B0), IPR059 (comparator B1)	IPR[3:0]	When using an interrupt: Select the interrupt priority level.
	IR058 (comparator B0), IR059 (comparator B1)	IR	When using an interrupt: 0 (no interrupt requested: initialization)
	IER07	IEN2 (comparator B0) IEN3 (comparator B1)	When using an interrupt: 1 (interrupt is enabled on the interrupt controller (ICU) side)

Note 1. This setting is necessary when changing the reference input voltage from the CVREFBn input to the internal reference voltage or from the internal reference voltage to the CVREFBn input. When selecting the CVREFBn input after the reset is released, steps 4, 5, 7, and 8 are not necessary because the initial value of the CPBREF.CPBnVRF bit is 0.

**Table 38.4 Procedure for Setting Registers Associated with Comparator B When Window Function is Enabled**

Step No.	Register	Bit	Setting
1	P**PFS of the port to which the CMPBn pin is assigned	ASEL	1
2	CPBMD	CPBSPDMD	0 (always specify high-speed mode)
3	CPBCNT1	CPBnINI (n = 0, 1)	Powered on: 1
4	CPBF		Select whether to enable or disable the filter and select the sampling clock.
5	CPBCNT2	CPBnWCP (n = 0, 1)	1 (operation enabled)
6	Waiting for the comparator stabilization time (min. 100 μs)		
7	CPBOCR	CPBnOP, CPBnOE (n = 0, 1)	Set the CMPOBn output (select the polarity and set output enabled or disabled).
8	CPBINT	CPBnINTEN (n = 0, 1)	When using an interrupt: 1 (interrupt enabled)
		CPBnINTEG (n = 0, 1)	When using an interrupt or the ELC: Select the input edge (1 = both edges or 0 = single edge).
		CPBnINTPL (n = 0, 1)	When using an interrupt or the ELC: For CPBnINTEG = 0 (single edge selected), select the input polarity (1 = rising edge or 0 = falling edge).
9	IPR058 (comparator B0), IPR059 (comparator B1)	IPR[3:0]	When using an interrupt: Select the interrupt priority level.
	IR058 (comparator B0), IR059 (comparator B1)	IR	When using an interrupt: 0 (no interrupt requested: initialization)
	IER07	IEN2 (comparator B0) IEN3 (comparator B1)	When using an interrupt: 1 (interrupt enabled)

Figure 38.3 shows an operating example of comparator B<sub>n</sub> (n = 0, 1) when window function is disabled.

The reference input voltage (CVREFB0/CVREFB1 or internal reference voltage) and the analog input voltage are compared. If the analog input voltage is higher than the reference input voltage, the CPBFLG.CPBnOUT bit is set to 1. If the analog input voltage is lower than the reference input voltage, the CPBnOUT bit is set to 0.

To use the comparator B<sub>n</sub> interrupt, set the CPBINT.CPBnINTEN bit to 1 (interrupt enabled). If the comparison result changes at this time, a comparator B<sub>n</sub> interrupt request is generated. For details on interrupts, refer to section 38.4, Comparator B0 and Comparator B1 Interrupts.

Comparator B<sub>n</sub> outputs event signals to the ELC to activate other modules. For details on the ELC, refer to section 38.5, Event Link Output.

The values of the registers should not be changed during comparison.

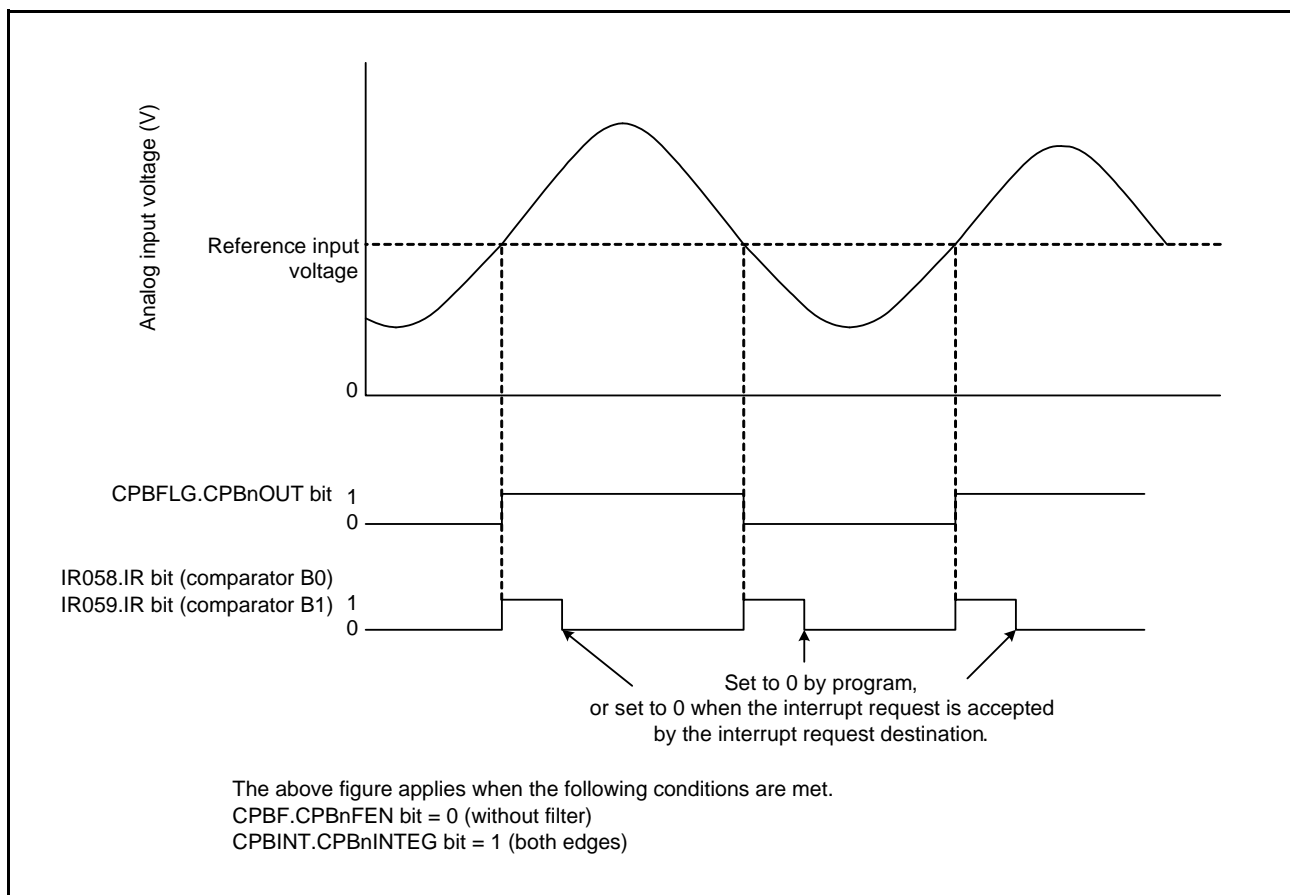


Figure 38.3 Operating Example of Comparator B<sub>n</sub> (n = 0, 1) When Window Function is Disabled

Figure 38.4 shows an operation example of comparator Bn (n = 0, 1) when the window function is enabled.

The internal reference voltage (VRFH/VRFL) for the window function and the analog input voltage are compared. The CPBnOUT bit is set to 1 when  $VRFL < \text{analog input voltage} < VRFH$ , and the CPBnOUT bit is set to 0 when the analog input voltage  $< VRFL$ , or  $VRFH < \text{analog input voltage}$ .

To use the comparator Bn interrupt, set the CPBINT.CPBnINTEN bit to 1 (interrupt enabled). If the comparison result changes at this time, a comparator Bn interrupt request is generated. For details on interrupts, refer to section 38.4, Comparator B0 and Comparator B1 Interrupts.

Comparator Bn outputs event signals to the ELC to activate other modules. For details on the ELC, refer to section 38.5, Event Link Output.

The values of the registers should not be changed during comparison.

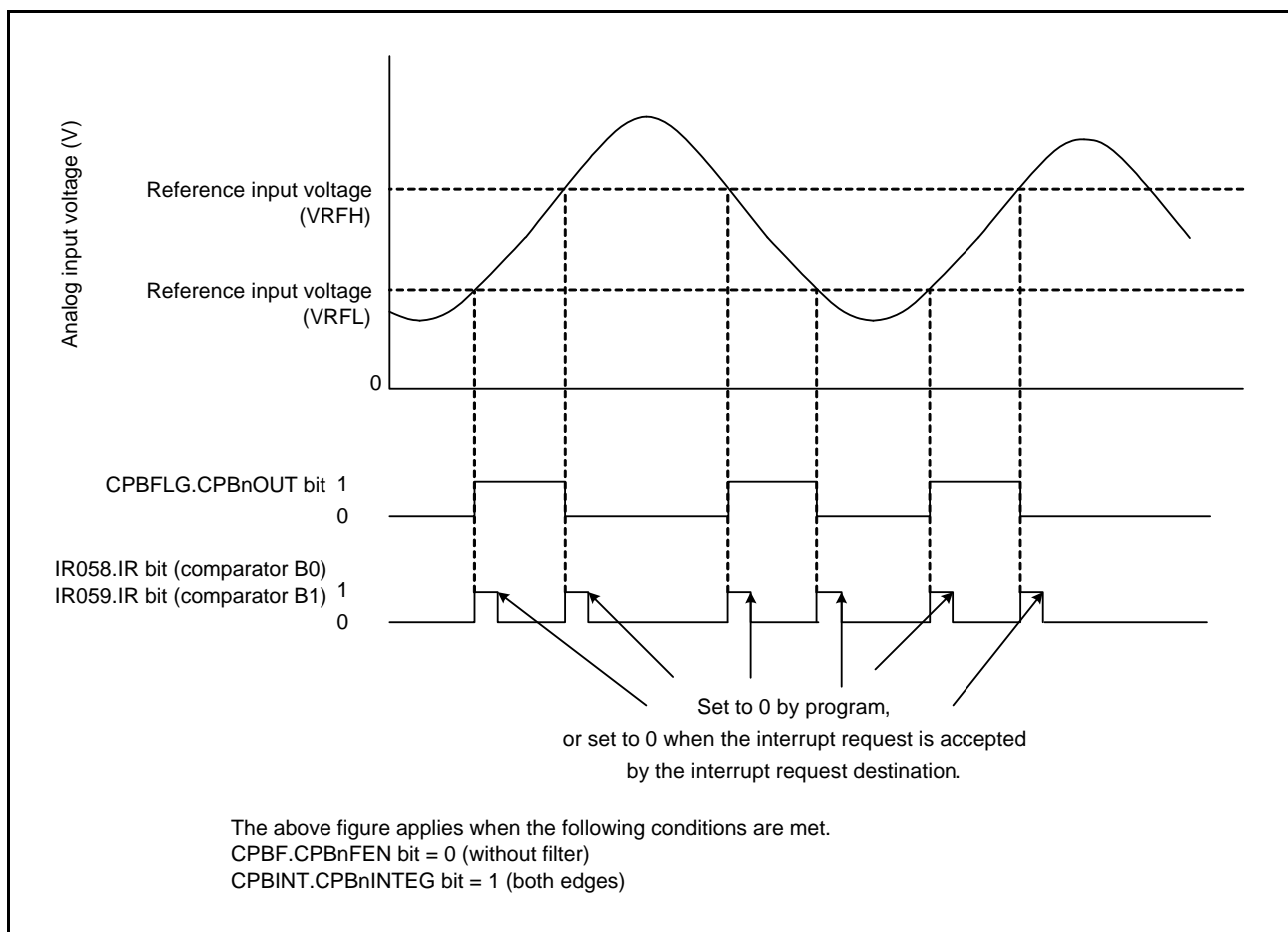


Figure 38.4 Operating Example of Comparator Bn (n = 0, 1) When Window Function is Enabled

### 38.3.1 Comparator Bn Digital Filter (n = 0, 1)

The sampling clock can be selected by the CPBF.CPBnF[1:0] bits. The CPBnOUT signal (internal signal) output from comparator Bn is sampled at every sampling clock cycle. At the next clock timing after the level matches three times, the IR058.IR bit (when comparator B0 selected) or IR059.IR bit (when comparator B1 selected) is set to 1 (interrupt requested) and an ELC event is output.

Figure 38.5 shows the configuration of the comparator Bn digital filter, and Figure 38.6 shows an operating example of the comparator Bn digital filter.

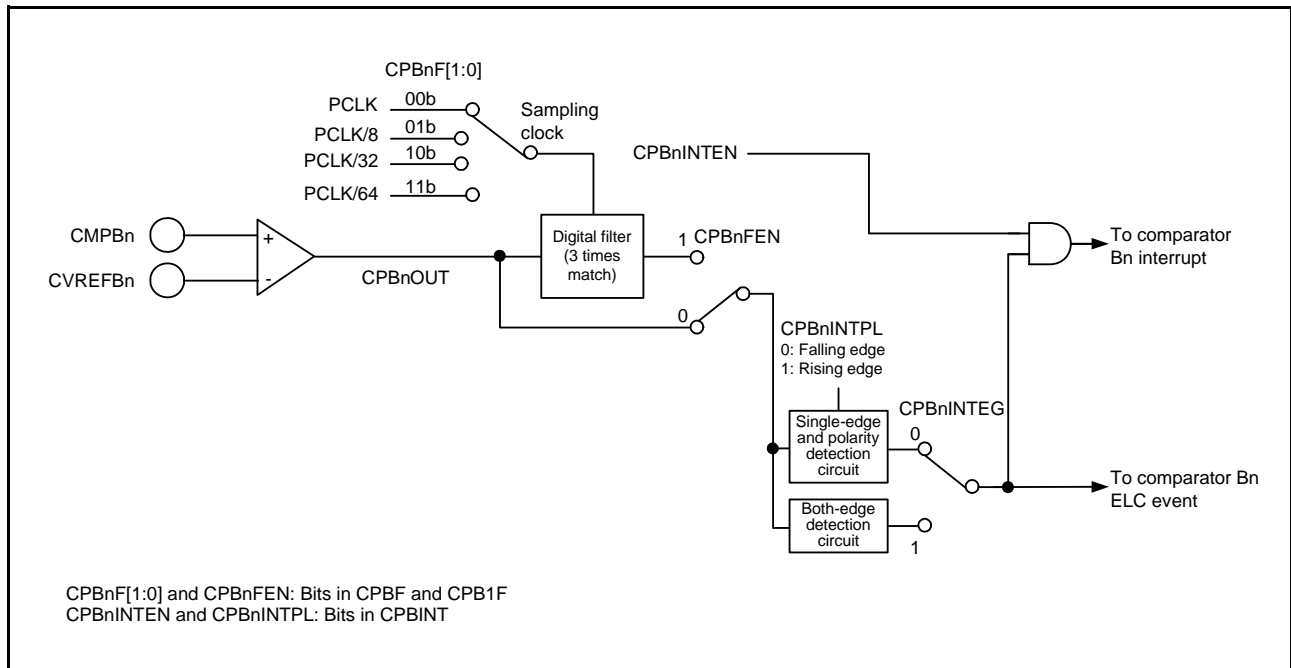


Figure 38.5 Configuration of Comparator Bn Digital Filter (n = 0, 1)

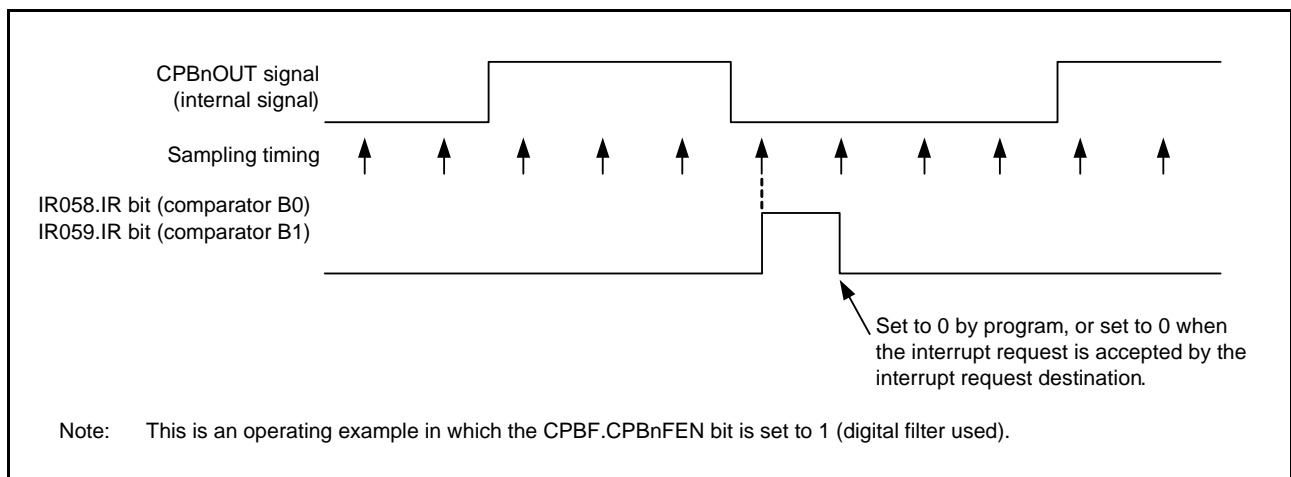


Figure 38.6 Operating Example of Comparator Bn Digital Filter (n = 0, 1)

## 38.4 Comparator B0 and Comparator B1 Interrupts

Comparator B generates two interrupt requests from sources, comparator B0 and comparator B1. The comparator Bn interrupt (n = 0, 1) uses the IR058.IR bit, IR059.IR bit, IPR058.IPR[3:0] bits, IPR059.IPR[3:0] bits, and the respective single interrupt vector.

To use the comparator Bn interrupt, set the CPBINT.CPBnINTEN bit to 1 (interrupt enabled). In addition, select either single-edge detection or both-edge detection using the CPBINT.CPBnINTEG bit. When single-edge detection is selected, select the polarity using the CPBINT.CPBnINTPL bit.

Inputs can also be passed through the digital filter with four different sampling clocks.

## 38.5 Event Link Output

Comparator B outputs the following events to the event link controller (ELC).

- (1) Comparison result of comparator B0 is changed.
- (2) Comparison result of comparator B0 or B1 is changed.

If the comparison results of comparators B0 and B1 are output simultaneously or in succession, they are output as a single event.

### 38.5.1 Relationship between Interrupt Handling and Event Linking

Comparator Bn outputs event signals to the event link controller (ELC) to initiate operations of other modules selected in advance. Event signals to the event link controller (ELC) is output independent of the CPBnINTEN bit value.

In the same way as for the interrupt sources, the conditions for generation of the event signals output from comparator Bn to the ELC can be selected as a single-edge detection or both-edge detection by setting the CPBINT.CPBnINTEG bit. When the single-edge detection is selected, the polarity can be selected by the CPBINT.CPBnINTPL bit.

### 38.5.2 Comparator Bn Output (n = 0, 1) Function

The comparison result from comparator B can be output to external pins. The CPBOCR.CPBnOP and CPBOCR.CPBnOE (n = 0, 1) bits can be used to set the output polarity (non-inverted output or inverted output) and output enabled or disabled. For the register settings and corresponding comparator output, refer to section 38.2.8, Comparator B Output Control Register (CPBOCR).

To output the comparator B comparison result to the CMPOB0 or CMPOB1 output pin, use the following procedure to make port settings. Note that the ports are set to input after a reset.

- (1) Set the mode and input for comparator B (steps 1 to 10 listed in Table 38.3 and steps 1 to 6 listed in Table 38.4).
- (2) Select the polarity of the CMPOB0 or CMPOB1 output and enable the output (set the CPBOCR.CPBnOP and CPBOCR.CPBnOE bits).
- (3) Set the port mode register and pin function control register corresponding to the CMPOB0 or CMPOB1 output pin (start outputting from the pin).

### 38.5.3 Example of Using Comparator B to Exit Software Standby Mode

The following shows an example of using comparator B0 output to exit software standby mode.

In this example, it is assumed that the reference input voltage (CVREFB0) > analog input voltage (CMPB0).

Set the following steps (1) to (3) before entering software standby mode.

- (1) Set the registers associated with comparator B0 according to section 38.3, Operation.  
However, set the CPBF.CPBOFEN bit to 'filter is disabled', the CPBOCR.CPB0OE bit to 'output enabled', and the CPBOCR.CPB0OP bit to 'comparator B0 output is output to CMPOB0'.
- (2) Make the IRQ7 interrupt settings according to section 14.4.8, External Pin Interrupts.  
However, set the IRQFLTE0.FLTEN7 bit to 0 (digital filter disabled) and set the IRQCRi.IRQMD[1:0] bits to the same polarity as that of comparator B0 output.  
In this example, a rising edge is selected.
- (3) Set the multi-function pin controller to select the CMPOB0 function and enable IRQ7.

When exiting software standby mode, input a voltage from the comparator B0 analog pin (CMPB0) so that the reference input voltage (CVREFB0) is less than the analog input voltage (CMPB0). This allows the IRQ7 interrupt to be generated through the comparator B0 output pin (CMPOB0) and the MCU exits software standby mode.

## 38.6 Usage Note

### 38.6.1 Module Stop Function Setting

Operation of comparator B can be disabled or enabled using module stop control register B (MSTPCRB). The initial setting is for comparator B to be halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

## 39. Data Operation Circuit (DOC)

### 39.1 Overview

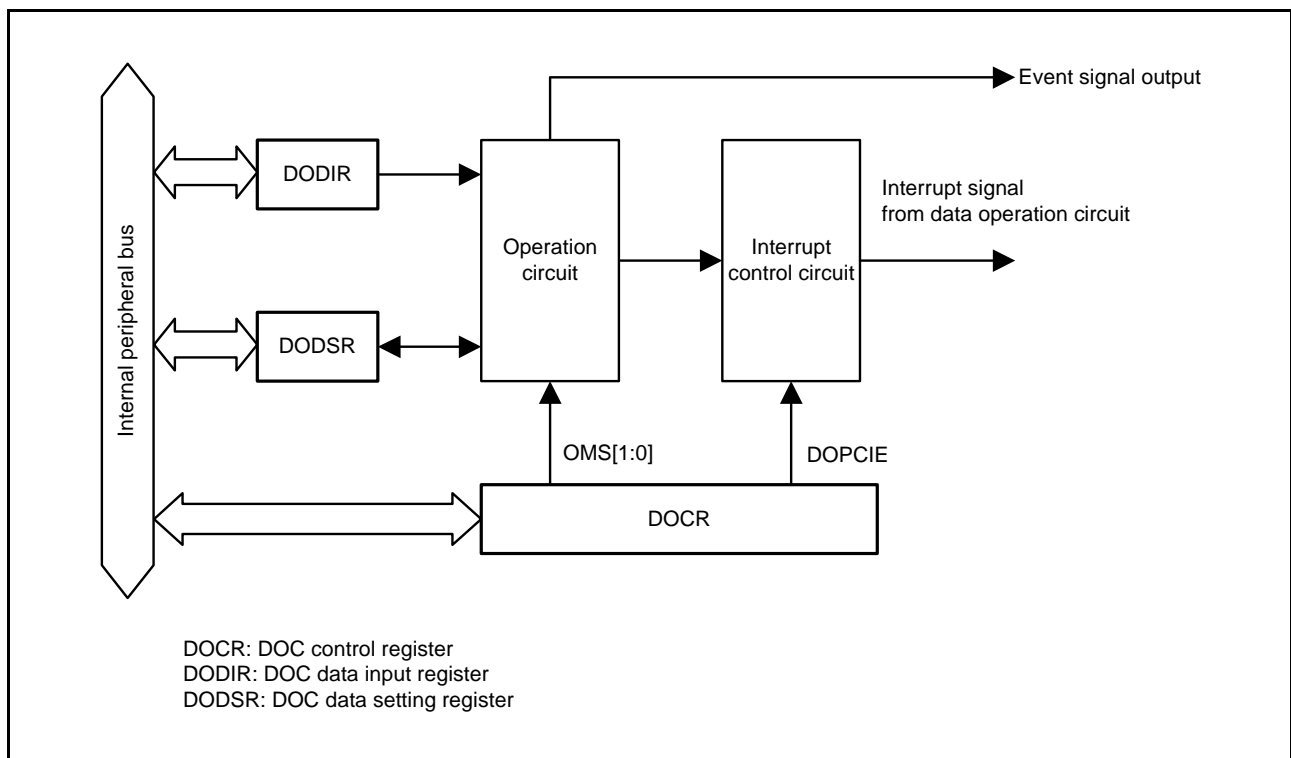
The data operation circuit (DOC) is used to compare, add, and subtract 16-bit data.

Table 39.1 lists the data operation circuit specifications and Figure 39.1 shows a block diagram of the data operation circuit.

16-bit data is compared and an interrupt can be generated when a selected condition applies.

**Table 39.1 DOC Specifications**

Item	Description
Data operation function	16-bit data comparison, addition, and subtraction
Lower power consumption function	Module stop state can be set.
Interrupts	An interrupt occurs at the following timings: <ul style="list-style-type: none"> <li>• The compared values either match or mismatch</li> <li>• The result of data addition is greater than FFFFh</li> <li>• The result of data subtraction is less than 0000h</li> </ul>
Event link function (output)	An interrupt occurs at the following timings: <ul style="list-style-type: none"> <li>• The compared values either match or mismatch</li> <li>• The result of data addition is greater than FFFFh</li> <li>• The result of data subtraction is less than 0000h</li> </ul>



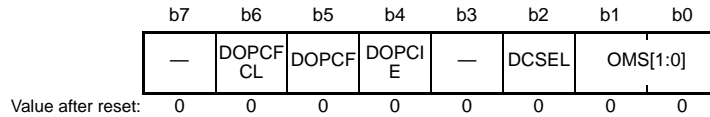
**Figure 39.1 DOC Block Diagram**



## 39.2 Register Descriptions

### 39.2.1 DOC Control Register (DOCR)

Address(es): 0008 B080h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	OMS[1:0]	Operating Mode Select	b1 b0 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
b2	DCSEL*1	Detection Condition Select	Result of data comparison 0: Data mismatch is detected. 1: Data match is detected.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	DOPCIE	Data Operation Circuit Interrupt Enable	0: Disables interrupts from the data operation circuit. 1: Enables interrupts from the data operation circuit.	R/W
b5	DOPCF	Data Operation Circuit Flag	Indicates the result of an operation.	R
b6	DOPCFCL	DOPCF Clear	0: Maintains the DOPCF flag state. 1: Clears the DOPCF flag.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Valid only when data comparison mode is selected.

#### OMS[1:0] Bits (Operating Mode Select)

These bits select the operating mode of the data operation circuit.

#### DCSEL Bit (Detection Condition Select)

This bit is valid only when data comparison mode is selected.

This bit selects the condition for detection in data comparison mode.

#### DOPCIE Bit (Data Operation Circuit Interrupt Enable)

Setting this bit to 1 enables interrupts from the data operation circuit.

#### DOPCF Flag (Data Operation Circuit Flag)

[Setting conditions]

- The condition selected by the DCSEL bit is met
- A result of data addition is greater than FFFFh
- A result of data subtraction is less than 0000h

[Clearing condition]

- Writing 1 to the DOPCFCL bit

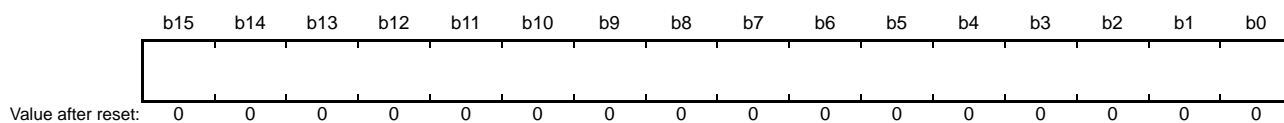
#### DOPCFCL Bit (DOPCF Clear)

Setting this bit to 1 clears the DOPCF flag.

This bit is read as 0.

### 39.2.2 DOC Data Input Register (DODIR)

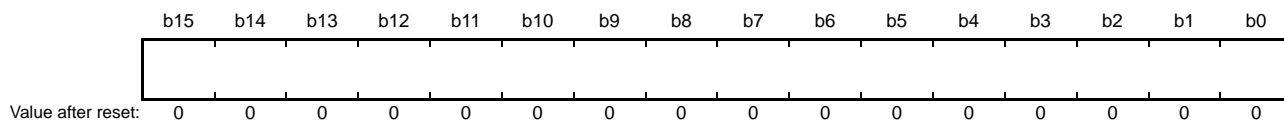
Address(es): 0008 B082h



DODIR is a 16-bit readable/writable register in which 16-bit data for use in the operations are stored.

### 39.2.3 DOC Data Setting Register (DODSR)

Address(es): 0008 B084h



DODSR is a 16-bit readable/writable register. This register stores 16-bit data for use as a reference in data comparison mode. This register also stores the results of operations in data addition and data subtraction modes.

### 39.3 Operation

#### 39.3.1 Data Comparison Mode

Figure 39.2 shows an example of the steps involved in data comparison mode operation by the data operation circuit. The following is an example of operation when DCSEL is set to 0 (data mismatch is detected as a result of data comparison).

- (1) Writing 00b to the DOCR.OMS[1:0] bits selects data comparison mode.
- (2) The 16-bit reference data is set in DODSR.
- (3) 16-bit data for comparison is written to DODIR.
- (4) Writing of 16-bit data continues until all data for comparison have been written to DODIR.
- (5) If a value written to DODIR does not match that in DODSR\*1, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note 1. When DOCR.DCSEL = 0

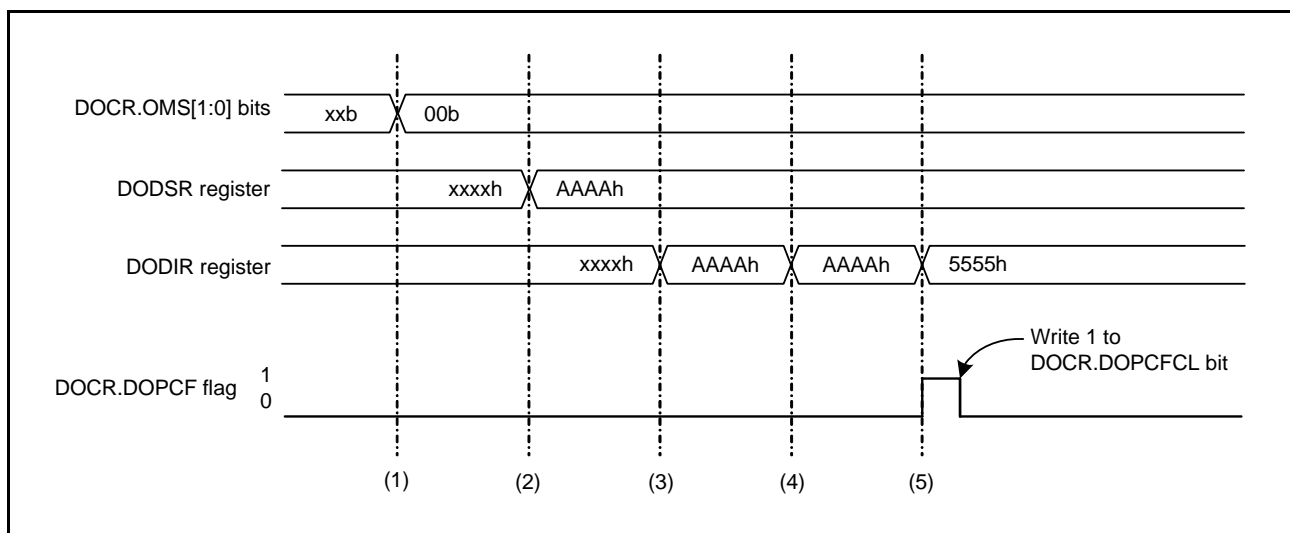


Figure 39.2 Example of Operation in Data Comparison Mode

### 39.3.2 Data Addition Mode

Figure 39.3 shows an example of the steps involved in data addition mode operation by the data operation circuit.

- (1) Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
- (2) 16-bit data is set in the DODSR register as the initial value.
- (3) 16-bit data to be added is written to DODIR. The result of the operation is stored in DODSR.
- (4) Writing of 16-bit data continues until all data for addition have been written to DODIR.
- (5) If the result of an operation is greater than FFFFh, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

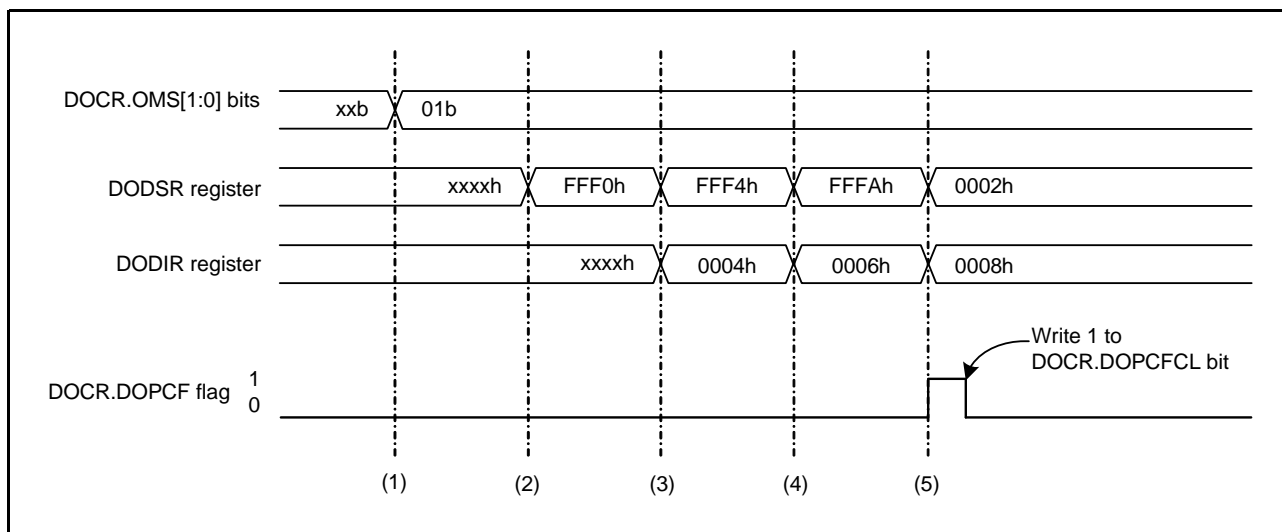


Figure 39.3 Example of Operation in Data Addition Mode

### 39.3.3 Data Subtraction Mode

Figure 39.4 shows an example of the steps involved in data subtraction mode operation by the data operation circuit.

- (1) Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
- (2) 16-bit data is set in the DODSR register as the initial value.
- (3) 16-bit data to be subtracted is written to DODIR. The result of the operation is stored in DODSR.
- (4) Writing of 16-bit data continues until all data for subtraction have been written to DODIR.
- (5) If the result of an operation is less than 0000h, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

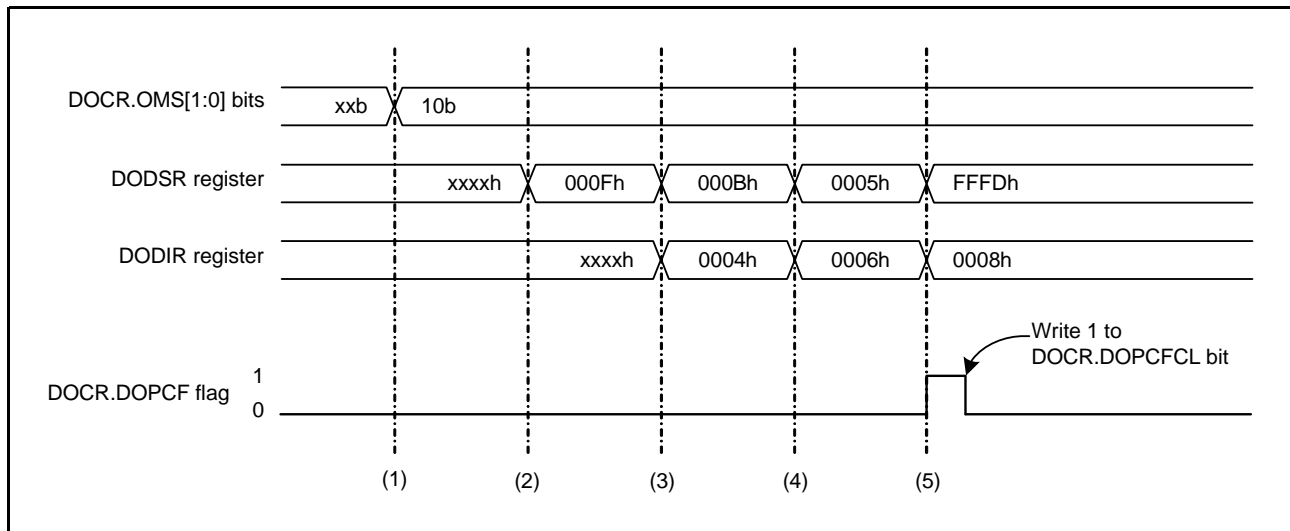


Figure 39.4 Example of Operation in Data Subtraction Mode

## 39.4 Interrupt Requests

The data operation circuit generates the data operation circuit interrupt as an interrupt request. When an interrupt source is generated, the data operation circuit flag corresponding to the interrupt is set to 1. Table 39.2 describes the interrupt request.

Table 39.2 Interrupt Request from Data Operation Circuit

Interrupt Request	Data Operation Circuit Flag	Interrupt Generation Timing
Data operation circuit interrupt	DOPCF	<ul style="list-style-type: none"> <li>• The compared values either match or mismatch</li> <li>• The result of data addition is greater than FFFFh</li> <li>• The result of data subtraction is less than 0000h</li> </ul>

## 39.5 Event Link Output

The DOC outputs event signals for the event link controller (ELC) under the following conditions, and these can be used to initiate operations by other modules selected in advance.

- The compared values either match or mismatch
- The result of data addition is greater than FFFFh
- The result of data subtraction is less than 0000h

### 39.5.1 Interrupt Handling and Event Linking

The DOC has a bit to enable or disable interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

## 39.6 Usage Note

### 39.6.1 Module Stop Function Setting

Operation of the data operation circuit can be disabled or enabled using module stop control register B (MSTPCRB). The initial setting is for the data operation circuit to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

## 40. RAM

This MCU has an on-chip high-speed static RAM.

### 40.1 Overview

Table 40.1 lists the specifications of the RAM.

**Table 40.1 RAM Specifications**

Item	Description
RAM capacity	Max. 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>
Low power consumption function	Module stop state can be set.

Note 1. Selectable by the SYSCR1.RAME bit. For details on the SYSCR1 register, refer to section 3.2.2, System Control Register 1 (SYSCR1).

Note 2. The capacity of RAM differs depending on the products.

RAM Capacity	RAM Address
32 Kbytes	RAM0: 0000 0000h to 0000 7FFFh
64 Kbytes	RAM0: 0000 0000h to 0000 FFFFh

### 40.2 Operation

#### 40.2.1 Low Power Consumption Function

Power consumption can be reduced by setting module stop control register C (MSTPCRC) to stop supply of the clock signal to the RAM.

Setting the MSTPCRC.MSTPC0 bit to 1 stops supply of the clock signal to RAM0.

Stopping supply of the clock signal places the RAM0 in the module stop state. The RAM operates after the value is initialized by a reset.

The RAM is not accessible in the module stop state. Do not make a transition to the module stop state while the RAM is being accessed.

For details on the MSTPCRC register, refer to section 11, Low Power Consumption.

## 41. Flash Memory

This MCU has packages with 128, 256, 384, and 512 Kbyte flash memory (ROM) for storing code and 8-Kbyte flash memory (E2 DataFlash) for storing data.

In this section, “PCLK” is used to refer to PCLKB.

### 41.1 Overview

Table 41.1 lists the specifications of the flash memory.

**Table 41.1 Flash Memory Specifications**

Item	Description
Memory space	<ul style="list-style-type: none"> <li>User area: Up to 512 Kbytes</li> <li>Data area: 8 Kbytes</li> </ul>
Software commands	<ul style="list-style-type: none"> <li>The following commands are implemented: Program, blank check, block erase, unique ID read</li> <li>The following commands are implemented for programming the extra area: Start-up area information program, access window information program</li> </ul>
On-board programming	Boot mode (SCI)*1 <ul style="list-style-type: none"> <li>Channel 1 of the serial communications interface (SCI1) is used for asynchronous serial communication.</li> <li>The user area and data area are rewritable.</li> </ul> Boot mode (USB interface)*1 <ul style="list-style-type: none"> <li>Channel 0 of the USB 2.0 function (USB0) module is used.</li> <li>The user area and data area are rewritable.</li> <li>The flash memory can be rewritable in self-powered or bus-powered mode.</li> <li>A personal computer can be connected using only a USB cable.</li> </ul> Self-programming in single-chip mode <ul style="list-style-type: none"> <li>The user area and data area are rewritable using the flash rewrite routine in the user program.</li> </ul>
Off-board programming	The user area and data area are rewritable using a flash programmer compatible with this MCU.
ID code protect	Connection with the flash programmer can be enabled or disabled using ID codes in boot mode.
Start-up program protection	This function is used to safely rewrite block 0 to block 15.
Area protection	This function enables rewriting only the selected blocks in the user area and disables the other blocks during self-programming.
Background Operation (BGO)	Programs on the ROM can be executed while rewriting the E2 DataFlash.

Note 1. Refer to “PG-FP5 Flash Memory Programmer User’s Manual” and “Renesas Flash Programmer Flash memory programming software User’s Manual” for more details.



### 41.2 ROM Area and Block Configuration

A maximum of 512 Kbytes can be configured in the ROM area. The ROM area is divided into blocks according to the ROM capacity. The ROM area is erased in block units. Figure 41.1 shows the ROM Area and Block Configuration.

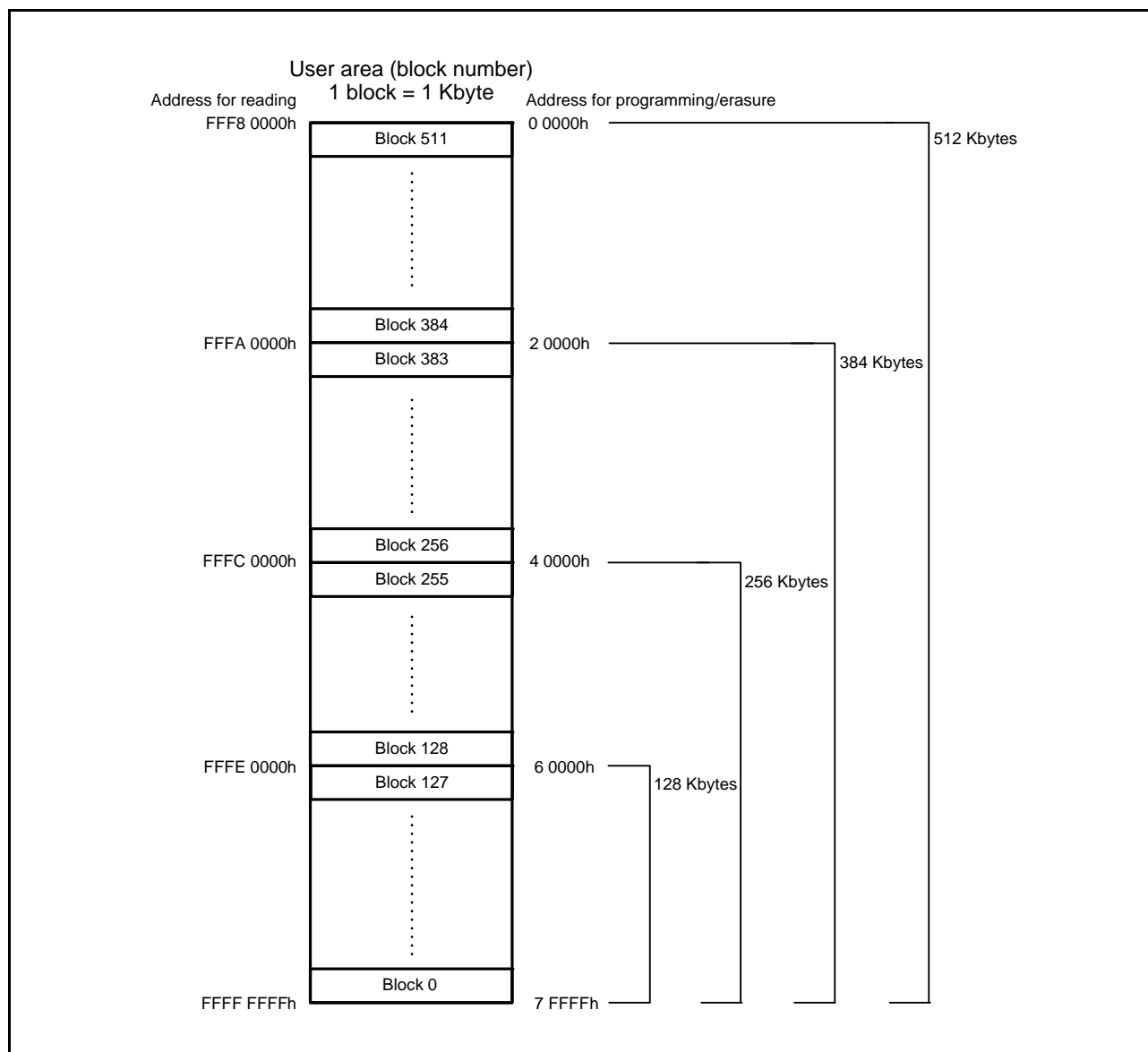


Figure 41.1 ROM Area and Block Configuration

Table 41.2 Correspondence Between ROM Capacity and Addresses

ROM Capacity	Address
512 Kbytes	FFF8 0000h to FFFF FFFh
384 Kbytes	FFFA 0000h to FFFF FFFh
256 Kbytes	FFFC 0000h to FFFF FFFh
128 Kbytes	FFE 0000h to FFFF FFFh

### 41.3 E2 DataFlash Area and Block Configuration

The E2 DataFlash is 8 Kbytes in the MCU. The E2 DataFlash is divided into blocks and erased in block units. Figure 41.2 shows the E2 DataFlash Area and Block Configuration.

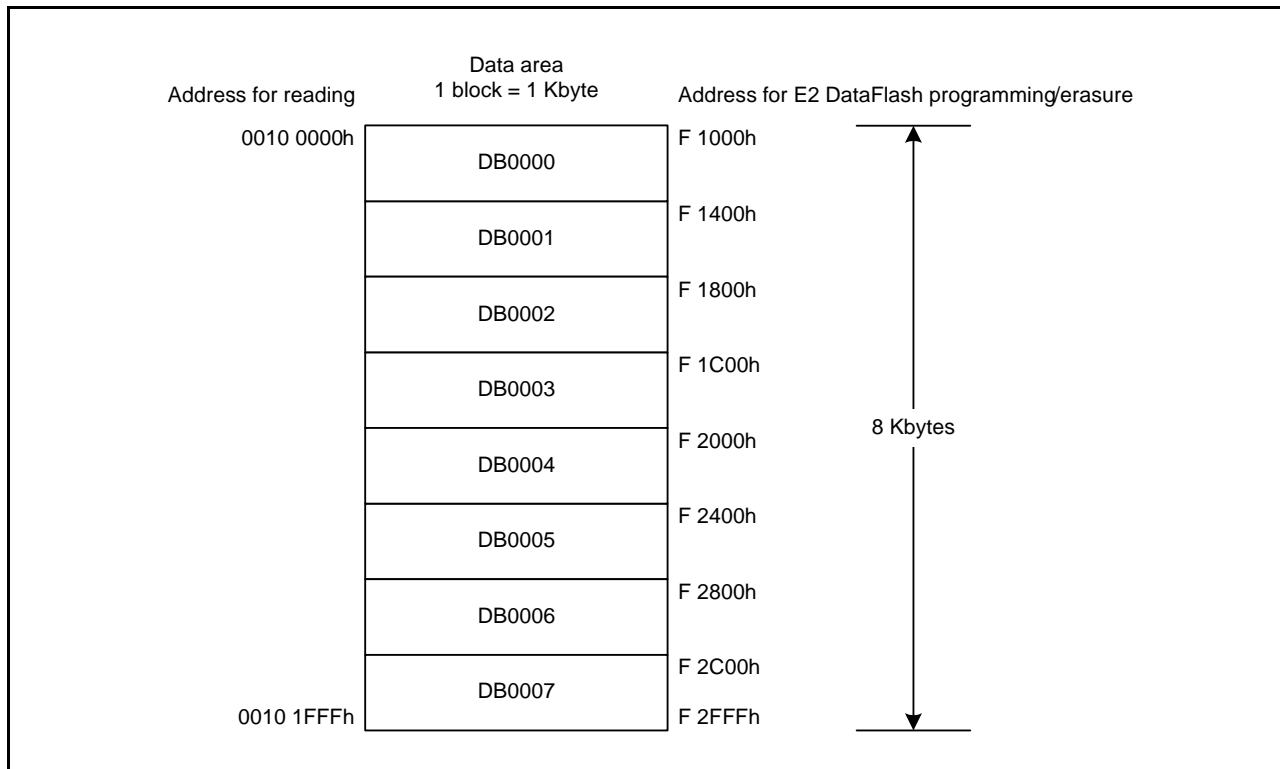
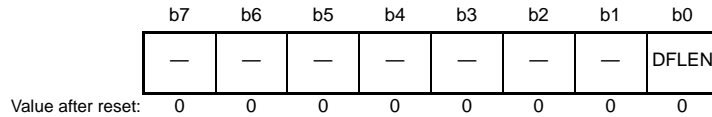


Figure 41.2 E2 DataFlash Area and Block Configuration

## 41.4 Register Descriptions

### 41.4.1 E2 DataFlash Control Register (DFLCTL)

Address(es): 007F C090h



Bit	Symbol	Bit Name	Description	R/W
b0	DFLEN	E2 DataFlash Access Enable	0: Access to E2 DataFlash and access to the extra area in P/E mode*1 disabled 1: Access to E2 DataFlash and access to the extra area in P/E mode*1 enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Unique ID read, start-up area information programming, and access window information program

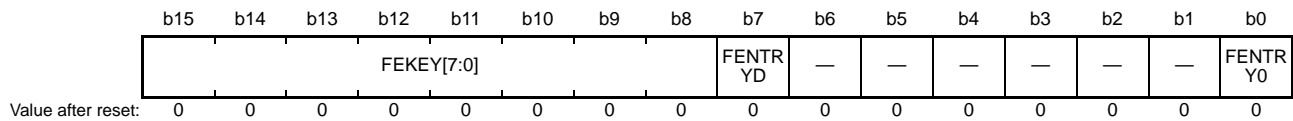
The DFLCTL register is used to enable or disable accessing (reading, programming, and erasing) to the E2 DataFlash and accessing (unique ID read, start-up area information programming, and access window information program) to the extra area in P/E mode.

When reading, programming, and erasing the E2 DataFlash, set the DFLCTL.DFLEN bit to 1 and wait for the E2 DataFlash STOP recovery time (tDSTOP) to elapse before reading the E2 DataFlash and entering E2 DataFlash P/E mode. Do not read the E2 DataFlash or enter E2 DataFlash P/E mode until tDSTOP has elapsed.

Refer to section 41.7.1, Sequencer Modes for details on E2 DataFlash P/E mode. Refer to section 42, Electrical Characteristics for E2 DataFlash STOP recovery time (tDSTOP).

## 41.4.2 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: ROM is in read mode. 1: ROM can be placed in P/E mode.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	FENTRYD	E2 DataFlash P/E Mode Entry	0: E2 DataFlash is in read mode. 1: E2 DataFlash can be placed in P/E mode.	R/W
b15 to b8	FEKEY[7:0]	Key Code	The FEKEY[7:0] bits are used to control rewiring of the FENTRYR register. When rewriting the value of the low-order 8 bits, set the FEKEY[7:0] bits to AAh at the same time (write this register in 16 bits). The FEKEY[7:0] bits are read as 00h.	R/W

To rewrite the ROM or E2 DataFlash, either the FENTRYD or FENTRY0 bit must be set to 1 to place the ROM or E2 DataFlash in P/E mode.

When returning to read mode, set the FENTRYR register and confirm that its value has been rewritten before reading the ROM or E2 DataFlash.

Refer to section 41.7.1, Sequencer Modes for details on P/E mode and read mode.

### FENTRY0 Bit (ROM P/E Mode Entry 0)

This bit is used to place the ROM in P/E mode.

[Setting condition]

- AA01h is written to the FENTRYR register when the FENTRYR register is 0000h.

Note: When entering ROM P/E mode, the instruction fetch address must be transferred to an area other than the ROM so that instruction fetching is not executed to the ROM. Copy necessary instruction code to the internal RAM and jump to the RAM. Note that E2 DataFlash can be rewritten by a program in the ROM.

[Clearing condition]

- AA00h is written to the FENTRYR register.

### FENTRYD Bit (E2 DataFlash P/E Mode Entry)

This bit is used to place the E2 DataFlash in P/E mode.

[Setting condition]

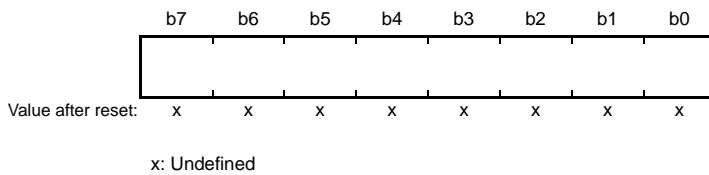
- AA80h is written to the FENTRYR register when the FENTRYR register is 0000h.

[Clearing condition]

- AA00h is written to the FENTRYR register.

### 41.4.3 Protection Unlock Register (FPR)

Address(es): 007F C0C0h



This write-only register is used to protect the FPMCR register from being rewritten inadvertently when the CPU runs out of control. Writing to the FPMCR register is enabled only when the following procedure is used to access the register.

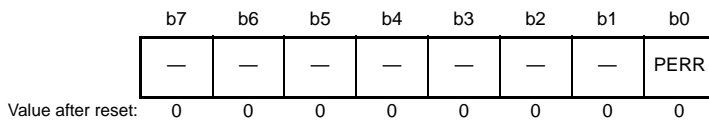
#### Procedure to unlock protection

- (1) Write A5h to the FPR register.
- (2) Write a set value to the FPMCR register.
- (3) Write the inverted set value to the FPMCR register.
- (4) Write a set value to the FPMCR register again.

When a procedure other than the above is used to write data, the FPSR.PERR flag is set to 1.

### 41.4.4 Protection Unlock Status Register (FPSR)

Address(es): 007F C0C1h



Bit	Symbol	Bit Name	Description	R/W
b0	PERR	Protect Error Flag	0: No error 1: An error occurs.	R
b7 to b1	—	Reserved	These bits are read as 0.	R

#### PERR Flag (Protect Error Flag)

When the FPMCR register is not accessed as described in the procedure to unlock protection, data is not written to the register and this flag is set to 1.

[Setting condition]

- The FPMCR register is not accessed as described in the procedure to unlock protection.

[Clearing condition]

- The FPMCR register is accessed according to the procedure to unlock protection described in section 41.4.3, Protection Unlock Register (FPR).

### 41.4.5 Flash P/E Mode Control Register (FPMCR)

Address(es): 007F FF80h

b7	b6	b5	b4	b3	b2	b1	b0
FMS2	LVPE	—	FMS1	RPDIS	—	FMS0	—
0	0	0	0	1	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W																								
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W																								
b1	FMS0	Flash Operating Mode Select 0	<table border="0"> <tr> <td>FMS2</td> <td>FMS1</td> <td>FMS0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0: ROM read mode</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0: E2 DataFlash P/E mode</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1: Discharge mode 1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1: ROM P/E mode</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1: Discharge mode 2</td> </tr> </table> Settings other than above are prohibited.	FMS2	FMS1	FMS0		0	0	0	0: ROM read mode	0	1	0	0: E2 DataFlash P/E mode	0	1	1	1: Discharge mode 1	1	0	1	1: ROM P/E mode	1	1	1	1: Discharge mode 2	R/W
FMS2	FMS1	FMS0																										
0	0	0	0: ROM read mode																									
0	1	0	0: E2 DataFlash P/E mode																									
0	1	1	1: Discharge mode 1																									
1	0	1	1: ROM P/E mode																									
1	1	1	1: Discharge mode 2																									
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W																								
b3	RPDIS	ROM P/E Disable	0: ROM programming/erasure enabled 1: ROM programming/erasure disabled	R/W																								
b4	FMS1	Flash Operating Mode Select 1	See the FMS0 bit.	R/W																								
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W																								
b6	LVPE	Low-Voltage P/E Mode Enable	0: Low-voltage P/E mode disabled 1: Low-voltage P/E mode enabled	R/W																								
b7	FMS2	Flash Operating Mode Select 2	See the FMS0 bit.	R/W																								

The FPMCR register is used to set the operating mode of the flash memory.

This register is protected. Set its value using the procedure to unlock protection. For details, refer to section 41.4.3, Protection Unlock Register (FPR).

When entering discharge mode 2 or ROM P/E mode, or during either of these modes, an instruction must be executed on the RAM.

#### FMS0, FMS1, and FMS2 Bits (Flash Operating Mode Select 0 to Flash Operating Mode Select 2)

These bits are used to set the operating mode of the flash memory.

[Transition from read mode to ROM P/E mode]

Set the FMS2 bit = 0, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.

Wait for ROM mode transition wait time 1 (tDIS, refer to section 42, Electrical Characteristics).

Set the FMS2 bit = 1, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.

Set the FMS2 bit = 1, the FMS1 bit = 0, the FMS0 bit = 1, and the RPDIS bit = 0.

Wait for ROM mode transition wait time 2 (tMS, refer to section 42, Electrical Characteristics).

[Transition from ROM P/E mode to read mode]

Set the FMS2 bit = 1, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.

Wait for ROM mode transition wait time 1 (tDIS, refer to section 42, Electrical Characteristics).

Set the FMS2 bit = 0, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.

Set the FMS2 bit = 0, the FMS1 bit = 0, the FMS0 bit = 0, and the RPDIS bit = 1.

Wait for ROM mode transition wait time 2 (tMS, refer to section 42, Electrical Characteristics).

[Transition from read mode to E2 DataFlash P/E mode]

Set the FMS2 bit = 0, the FMS1 bit = 1, the FMS0 bit = 0, and the RPDIS bit = 0.

[Transition from E2 DataFlash P/E mode to read mode]

Set the FMS2 bit = 0, the FMS1 bit = 0, the FMS0 bit = 0, and the RPDIS bit = 1.

Wait for ROM mode transition wait time 2 (tMS, refer to section 42, Electrical Characteristics).

#### RPDIS Bit (ROM P/E Disable)

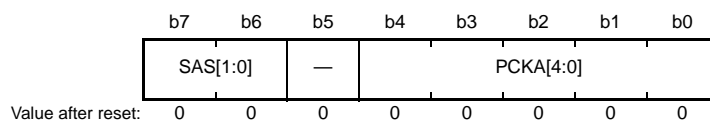
This bit is used to disable the execution of ROM programming/erasure with software.

#### LVPE Bit (Low-Voltage P/E Mode Enable)

Set this bit to 0 for programming/erasure in high-speed mode, and set this bit to 1 for programming/erasure in middle-speed mode.

### 41.4.6 Flash Initial Setting Register (FISR)

Address(es): 007F C0B6h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PCKA[4:0]	Peripheral Clock Notification	These bits are used to set the frequency of the FlashIF clock (FCLK).	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	SAS[1:0]	Start-Up Area Select	b7 b6 0 x: The start-up area is selected according to the start-up area settings of the extra area. 1 0: The start-up area is switched to the default area temporarily. 1 1: The start-up area is switched to the alternate area temporarily.	R/W

x: Don't care

Data can be written to the FISR register in ROM P/E mode or E2 DataFlash P/E mode.

#### PCKA[4:0] Bits (Peripheral Clock Notification)

These bits are used to set the frequency of the FlashIF clock (FCLK) when programming/erasing the ROM/E2 DataFlash.

Set the FCLK frequency in the PCKA[4:0] bits before programming/erasure. Do not change the frequency during programming/erasure of the ROM/E2 DataFlash.

[When FCLK is higher than 4 MHz]

Set a rounded-up value for a non-integer frequency.

For example, set 32 MHz (PCKA bit = 1111b) when the frequency is 31.5 MHz.

[When FCLK is 4 MHz or lower]

Do not use a non-integer frequency.

Use the FCLK at a frequency of 1, 2, 3, or 4 MHz.

Note: When the PCKA[4:0] bits are set to a frequency different from the FCLK, the data in the ROM/E2 DataFlash may be damaged.

**Table 41.3 Example of FlashIF Clock Frequency Settings**

FlashIF Clock Frequency [MHz]	PCKA[4:0] Bit Setting	FlashIF Clock Frequency [MHz]	PCKA[4:0] Bit Setting	FlashIF Clock Frequency [MHz]	PCKA[4:0] Bit Setting
32	11111b	31	11110b	30	11101b
29	11100b	28	11011b	27	11010b
26	11001b	25	11000b	24	10111b
23	10110b	22	10101b	21	10100b
20	10011b	19	10010b	18	10001b
17	10000b	16	01111b	15	01110b
14	01101b	13	01100b	12	01011b
11	01010b	10	01001b	9	01000b
8	00111b	7	00110b	6	00101b
5	00100b	4	00011b	3	00010b
2	00001b	1	00000b	—	—

**SAS[1:0] Bits (Start-Up Area Select)**

These bits are used to select the start-up area. To change the start-up area, the following three methods can be used.

**(1) When selecting the start-up area according to the start-up area settings of the extra area**

With the SAS[1:0] bits set to 00b or 01b, the start-up area is selected according to the start-up area settings of the extra area. The settings are enabled after a reset is released.

**(2) When switching the start-up area to the default area temporarily**

When 10b is written to the SAS[1:0] bits, the start-up area is switched to the default area immediately after data is written to the register, regardless of the start-up area settings of the extra area.

When a reset is generated after this, the area is selected according to the start-up area settings of the extra area.

**(3) When switching the start-up area to the alternative area temporarily**

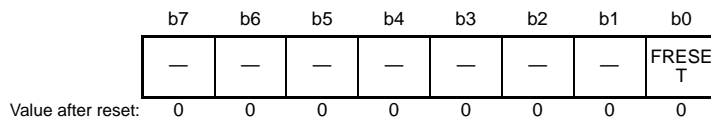
When 11b is written to the SAS[1:0] bits, the start-up area is switched to the alternative area, regardless of the start-up area settings of the extra area.

When a reset is generated after this, the area is selected according to the start-up area settings of the extra area.



### 41.4.7 Flash Reset Register (FRESETR)

Address(es): 007F FF89h



Bit	Symbol	Bit Name	Description	R/W
b0	FRESET	Flash Reset	0: Flash control circuit reset is released. 1: Flash control circuit is reset.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

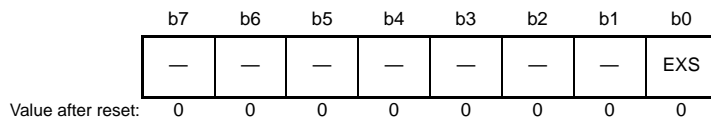
#### FRESET Bit (Flash Reset)

When this bit is set to 1, registers FASR, FSARH, FSARL, FEARH, FEARL, FWBH, FWBL, FCR, and FEXCR are reset. Also, the values of registers FEAMH and FEAML are undefined. Do not access these registers during a reset. To release the reset, set this bit to 0.

Do not write to this register while executing a software command or rewriting the extra area.

### 41.4.8 Flash Area Select Register (FASR)

Address(es): 007F FF81h



Bit	Symbol	Bit Name	Description	R/W
b0	EXS	Extra Area Select	0: User area or data area 1: Extra area	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Data can be written to the FASR register in ROM P/E mode or E2 DataFlash P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1.

Data cannot be written to this register while the FRESETR.FRESET bit is 1.

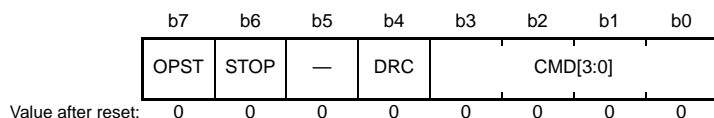
#### EXS Bit (Extra Area Select)

Set this bit to 1 when programming the extra area using the FEXCR register.

Set this bit to 0 when not programming the extra area.

### 41.4.9 Flash Control Register (FCR)

Address(es): 007F FF85h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CMD[3:0]	Software Command Setting	b3    b0 0 0 0 1: Program 0 0 1 1: Blank check 0 1 0 0: Block erase 0 1 0 1: Unique ID read Settings other than above are prohibited.*1	R/W
b4	DRC	Data Read Completion	0: Start data read. 1: Complete data read.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	STOP	Forced Processing Stop	When this bit is set to 1, the processing being executed can be forcibly stopped.	R/W
b7	OPST	Processing Start	0: Processing stops. 1: Processing starts.	R/W

Note 1. This does not include set the FCR register to 00h when the FSTATR1.FRDIY flag is 1.

Data can be written to the FCR register when in ROM P/E mode and the ROM can be programmed/erased or in E2 DataFlash P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.

Note that this register cannot be initialized by the FRESETR.FRESET bit while a software command is being executed. [Products with 384-Kbyte or 512-Kbyte ROM]

- Blank checking cannot be executed across 256-Kbyte boundaries.

#### CMD[3:0] Bits (Software Command Setting)

These bits are used to set a software command (program, block erase, unique ID read, or blank check).

The function of each command is described below.

[Program]

- Write the value set in registers FWBH and FWBL to the address set in registers FSARH and FSARL.

[Blank check]

- Check whether there is data in the area from the address set in registers FSARH and FSARL to the address set in registers FEARH and FEARL. Confirm that data is not programmed in the area. This command does not guarantee whether the area remains erased.

[Block erase]

- Erase the specified block in the flash memory.  
The beginning address of the block to be erased set in registers FSARH and FSARL, and the last address in registers FEARH and FEARL. If a value other than the above is set, erasure may not be executed correctly.

[Unique ID read]

- When executing the unique ID read after setting registers FSARH, FSARL, FEARH, and FEARL to 00h, 0850h, 00h, and 086Fh, respectively, the unique ID is stored in registers FRBH and FRBL sequentially.

**DRC Bit (Data Read Completion)**

This bit is used with the unique ID read command to control the state of the sequencer.

When issuing the unique ID read command with this bit set to 0, data is read from the address set in registers FSARH and FSARL, and the data is stored in registers FRBH and FRBL.

When issuing the unique ID read command with this bit set to 1 after reading data from registers FRBH and FRBL, the sequencer ends the read cycle and enters the wait state.

When issuing the unique ID read command again with this bit set to 0, the internal address of the sequencer is incremented by 4, and the next data is read.

**STOP Bit (Forced Processing Stop)**

This bit is used to forcibly stop the processing (blank check or block erase) being executed.

After setting this bit to 1, wait until the FSTATR1.FRDY flag is 1 (processing completed) before setting the OPST bit to 0.

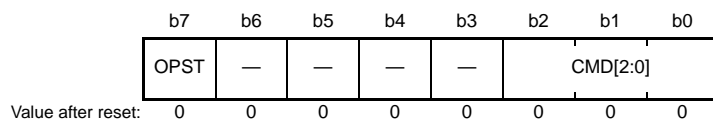
**OPST Bit (Processing Start)**

This bit is used to execute the command set in the CMD[2:0] bits.

This bit is not set to 0 again even when the processing is completed. Confirm that the FSTATR1.FRDY flag is 1 (processing completed) before setting the OPST bit to 0 again. After that, confirm that the FRDY flag is 0 before executing the next processing.

**41.4.10 Flash Extra Area Control Register (FEXCR)**

Address(es): 007F C0B7h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD[2:0]	Software Command Setting	b2 b0 0 0 1: Start-up area information program 0 1 0: Access window information program Settings other than above are prohibited.*1	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OPST	Processing Start	0: Processing stops. 1: Processing starts.	R/W

Note 1. This does not include writing 00h to the FEXCR register when the FSTATR1.EXRDY flag is 1.

Data can be written to the FEXCR register when in ROM P/E mode and the ROM can be programmed/erased.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.

Note that this register cannot be initialized by the FRESETR.FRESET bit while a software command is being executed.

**CMD[2:0] Bits (Software Command Setting)**

This bit is used to set a software command (start-up area information program or access window information program).

The details of each command are described below.

[Start-up area information program]

This command is used to switch the start-up area used for start-up program protection.

- When setting the start-up area to the default area

Set the FWBH register to FFFFh, set the FWBL register to FFFFh, and execute this command.

- When setting the start-up area to the alternative area

Set the FWBH register to FFFFh, set the FWBL register to FEFFh, and execute this command.

When registers FWBH and FWBL are set to values other than the above, do not execute the start-up area information program.

[Access window information program]

This command is used to set the access window used for area protection.

Set the access window in block units.

Specify the access window start address, which is the beginning address of the access window in the FWBL register, specify the access window end address, which is the next address of the last address of the access window in the FWBH register, and issue this command. Set bit 19 to bit 10 of the address for programming/erasure in each register.

If the same value is set as the start address and end address, all areas can be accessed. Do not set the start address to a value larger than the value of the end address.

### OPST Bit (Processing Start)

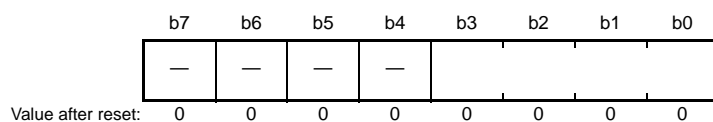
This bit is used to execute the command set in the CMD[2:0] bits.

This bit is not set to 0 again even when the processing is completed. Confirm that the FSTATR1.EXRDY flag is 1 (processing completed) before setting the OPST bit to 0 again. After that, confirm that the FSTATR1.EXRDY flag is 0 before executing the next processing.

Writing to the extra area is started by writing 1 to the OPST bit. Do not write to the CMD[2:0] bits while a software command is being executed.

## 41.4.11 Flash Processing Start Address Register H (FSARH)

Address(es): 007F FF84h



The FSARH register is used to set the target processing address or the start address of the target processing range in the flash memory when a software command is executed.

Set bit 19 to bit 16 of the flash memory address for programming/erasure in this register.

Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.

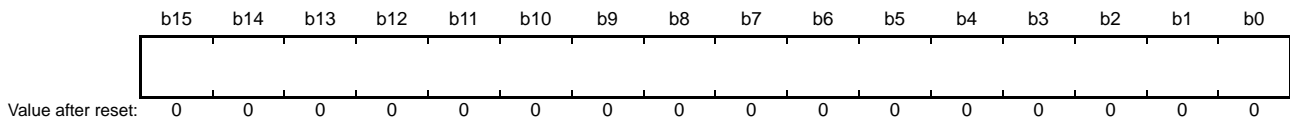
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.

If this register is read while executing a software command set by the FEXCR register, an undefined value is read.

Refer to Figure 41.1 and Figure 41.2 for details on the addresses of the flash memory.

### 41.4.12 Flash Processing Start Address Register L (FSARL)

Address(es): 007F FF82h



The FSARL register is used to set the target processing address or the start address of the target processing range in the flash memory when a software command is executed.

Set bit 15 to bit 0 of the address for programming/erasure in this register.

To set the ROM area, set bit 1 and bit 0 to 00b.

Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.

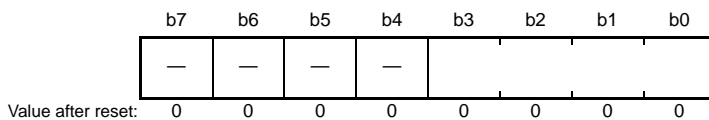
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.

If this register is read while executing a software command set by the FEXCR register, an undefined value is read.

Refer to Figure 41.1 and Figure 41.2 for details on the addresses of the flash memory.

### 41.4.13 Flash Processing End Address Register H (FEARH)

Address(es): 007F FF88h



The FEARH register is used to set the end address of the target processing range in the flash memory when a software command is executed.

Set bit 19 to bit 16 of the flash memory address for programming/erasure in this register.

Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.

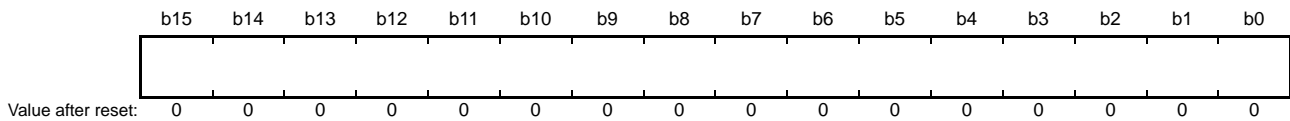
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.

If this register is read while executing a software command set by the FEXCR register, an undefined value is read.

Refer to Figure 41.1 and Figure 41.2 for details on the addresses of the flash memory.

#### 41.4.14 Flash Processing End Address Register L (FEARL)

Address(es): 007F FF86h



The FEARH register is used to set the end address of the target range for processing when a software command is executed.

Set bit 15 to bit 0 of the flash memory address for programming/erasure in this register.

Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.

If this register is read while executing a software command set by the FEXCR register, an undefined value is read.

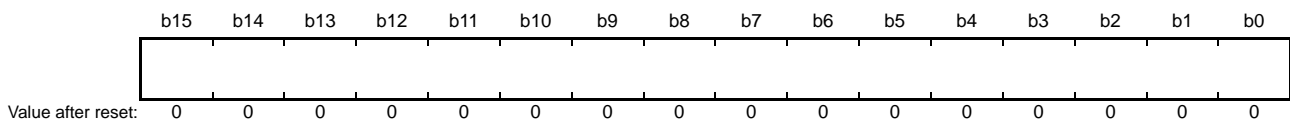
Refer to Figure 41.1 and Figure 41.2 for details on the addresses of the flash memory.

[Products with 384-Kbyte or 512-Kbyte ROM]

- The range specified by registers FSARH and FSARL and registers FEARH and FEARL cannot be set to a range that spans across 256-Kbyte boundaries.

#### 41.4.15 Flash Read Buffer Register H (FRBH)

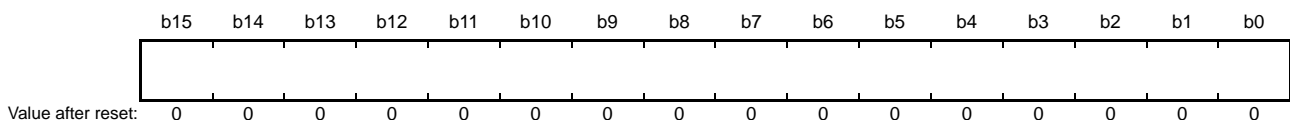
Address(es): 007F C0C4h



This register is used to store the upper 2 bytes of the 4-byte data (part of the unique ID) that is read from the extra area when unique ID read is executed.

#### 41.4.16 Flash Read Buffer Register L (FRBL)

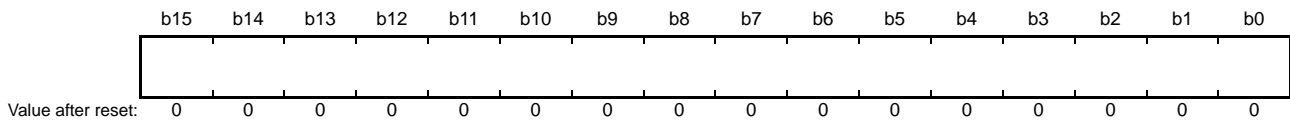
Address(es): 007F C0C2h



This register is used to store the lower 2 bytes of the 4-byte data (part of the unique ID) that is read from the extra area when unique ID read is executed.

### 41.4.17 Flash Write Buffer Register H (FWBH)

Address(es): 007F FF8Eh



This register is used to set the high-order 16 bits of the data for programming the ROM.

Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.

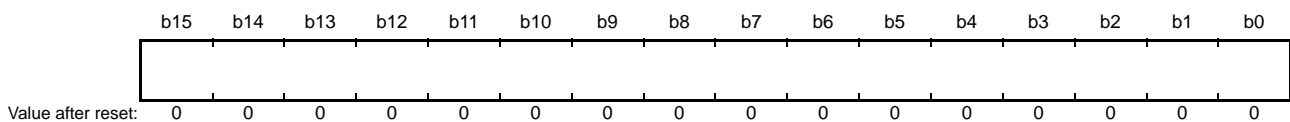
The value read from this register is undefined while a software command is being executed.

The data for programming the E2 DataFlash should be specified in the low-order 8 bits in the FWBL register.

If this register is read while executing a software command set by the FEXCR register, an undefined value is read.

### 41.4.18 Flash Write Buffer Register L (FWBL)

Address(es): 007F FF8Ch



This register is used to set the low-order 16 bits of the data for programming the ROM or the data for programming the E2 DataFlash.

The data for programming the E2 DataFlash should be set in bit 7 to bit 0.

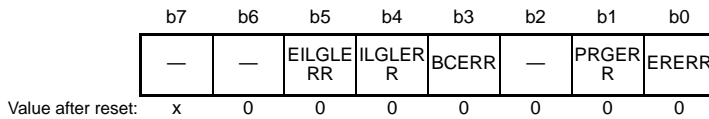
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.

If this register is read while executing a software command set by the FEXCR register, an undefined value is read.

### 41.4.19 Flash Status Register 0 (FSTATR0)

Address(es): 007F FF8Ah



Bit	Symbol	Bit Name	Description	R/W
b0	ERERR	Erase Error Flag	0: Erasure terminates normally. 1: An error occurs during erasure.	R
b1	PRGERR	Program Error Flag	0: Programming terminates normally. 1: An error occurs during programming.	R
b2	—	Reserved	The read value is undefined.	R
b3	BCERR	Blank Check Error Flag	0: Blank checking terminates normally. 1: An error occurs during blank checking.	R
b4	ILGLERR	Illegal Command Error Flag	0: No illegal software command or illegal access is detected. 1: An illegal command or illegal access is detected.	R
b5	EILGLERR	Extra Area Illegal Command Error Flag	0: No illegal command or illegal access to the extra area is detected. 1: An illegal command or illegal access to the extra area is detected.	R
b7, b6	—	Reserved	The read value is undefined.	R

This register is a status register used to confirm the result of executing a software command. Each error flag is set to 0 when the next software command is executed.

#### ERERR Flag (Erase Error Flag)

This bit indicates the result of the erase processing for the ROM/E2 DataFlash.

[Setting condition]

- An error occurs during erasure.

[Clearing condition]

- The next software command is executed.

The value read from this flag is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during erasure.

#### PRGERR Flag (Program Error Flag)

This bit indicates the result of the program processing for the ROM/E2 DataFlash.

[Setting condition]

- An error occurs during programming.

[Clearing condition]

- The software command is executed.

#### BCERR Flag (Blank Check Error Flag)

This bit indicates the result of the blank check processing for the ROM/E2 DataFlash.

[Setting condition]

- An error occurs during blank checking.



[Clearing condition]

- The next software command is executed.

The value read from this flag is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during blank checking.

### **ILGLERR Flag (Illegal Command Error Flag)**

This bit indicates the result of executing a software command.

[Setting conditions]

- Programming/erasure is executed to an area other than the access window range.
- Either the blank check or block erase command is executed when the set value of registers FSARH and FSARL is larger than the set value of registers FEARH and FEARL.
- Program and block erase commands are executed when the FASR.EXS bit is 1.
- The E2 DataFlash address is set in registers FSARH and FSARL and a software command is executed when the ROM is in P/E mode.
- The ROM address is set in registers FSARH and FSARL and a software command is executed when the E2 DataFlash is in P/E mode.
- The ROM and E2 DataFlash are set to P/E mode and a software command is executed.

[Clearing condition]

- The next software command is executed.

### **EILGLERR Flag (Extra Area Illegal Command Error Flag)**

This flag indicates the result of executing a software command for the extra area.

[Setting condition]

- A software command for the extra area is executed when the FASR.EXS bit is 0.

[Clearing condition]

- The next software command is executed.

### 41.4.20 Flash Status Register 1 (FSTATR1)

Address(es): 007F FF8Bh

b7	b6	b5	b4	b3	b2	b1	b0
EXRDY	FRDY	—	—	—	—	DRRDY	—
Value after reset:	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0.	R
b1	DRRDY	Data Read Ready Flag	0: No valid data in registers FRBH and FRBL 1: Valid data in registers FRBH and FRBL	R
b2	—	Reserved	This bit is read as 1.	R
b5 to b3	—	Reserved	These bits are read as 0.	R
b6	FRDY	Flash Ready Flag	0: Other than below 1: 00h can be written to the FCR register (processing to complete the software command).	R
b7	EXRDY	Extra Area Ready Flag	0: Other than below 1: 00h can be written to the FEXCR register (processing to complete the software command).	R

This register is a status register used to confirm the result of executing a software command. Each flag is set to 0 when the next software command is executed.

#### DRRDY Flag (Data Read Ready Flag)

This flag is used to check if the valid read data is stored in registers FRBH and FRBL.

When the sequencer stores data read from the flash memory to registers FRBH and FRBL, the DRRDY flag becomes 1. When issuing the unique ID command with the FCR.DRC bit set to 1, the sequencer ends the read cycle, and the DRRDY flag becomes 0.

Note that, even if issuing the unique ID command with the FCR.DRC bit set to 0 after reading data from the address set in registers FEARH and FEARL, the DRRDY flag does not become 1, but the FRDY flag becomes 1.

#### FRDY Flag (Flash Ready Flag)

This flag is used to confirm whether a software command is executed.

This flag becomes 1 when processing of the executed software command or the forced stop processing is completed, and this flag becomes 0 when setting the FCR.OPST bit to 0.

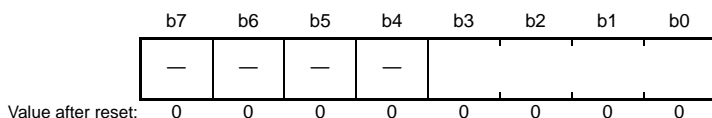
#### EXRDY Flag (Extra Area Ready Flag)

This flag is used to confirm whether a software command for the extra area is executed.

This flag is set to 1 when processing of the executed software command is completed, and 0 when the FEXCR.OPST bit is set to 0.

### 41.4.21 Flash Error Address Monitor Register H (FEAMH)

Address(es): 007F C0BAh



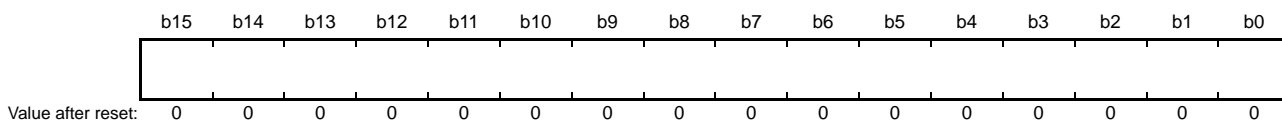
This register is used to confirm the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 19 to bit 16 of the address where the error has occurred for the program command or blank check command, or it stores bit 19 to bit 16 of the beginning address of the area where the error has occurred for the block erase command.

Since this register value becomes undefined if setting the FRESETR.FRESET bit to 1, read the value before error processing. If the software command terminates normally, this register stores bit 19 to bit 16 of the end address at execution of the command.

Refer to Figure 41.1 and Figure 41.2 for details on the addresses of the flash memory.

### 41.4.22 Flash Error Address Monitor Register L (FEAML)

Address(es): 007F C0B8h



This register is used to confirm the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 15 to bit 0 of the address where the error has occurred for the program command or blank check command, or it stores bit 15 to bit 0 of the beginning address where the error has occurred for the block erase command.

Since this register value becomes undefined if setting the FRESETR.FRESET bit to 1, read the value before error processing.

When the software command is normally completed, this register stores bit 15 to bit 0 of the last address at execution of the command.

When executing a software command for the ROM or the unique ID read command, low-order 2 bits become 00b.

Refer to Figure 41.1 and Figure 41.2 for details on the addresses of the flash memory.

### 41.4.23 Flash Start-Up Setting Monitor Register (FSCMR)

Address(es): 007F C0B0h

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

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0.	R
b8	SASMF	Start-Up Area Setting Monitor Flag	0: Setting to start up using the alternative area 1: Setting to start up using the default area	R
b14 to b9	—	Reserved	These bits are read as 1. Writing to these bits has no effect.	R
b15	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

Note 1. The value of the blank product is 1. It is set to the same value set in bit 8 in the FWBL register after the start-up area information program command is executed.

#### SASMF Flag (Start-Up Area Setting Monitor Flag)

This flag is used to confirm the settings of the start-up area.

When this flag is 0, the user program is set to start up using the alternative area.

When this flag is 1, the user program is set to start up using the default area.

### 41.4.24 Flash Access Window Start Address Monitor Register (FAWSMR)

Address(es): 007F C0B2h

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

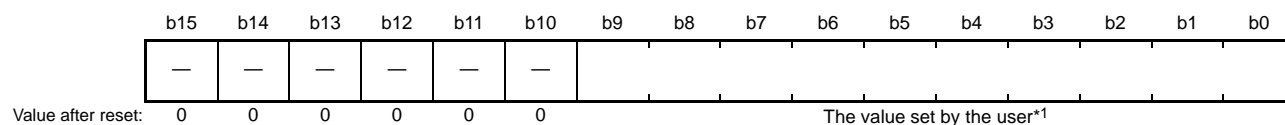
The value set by the user\*1

Note 1. The value of the blank product is 1. It is set to the same value set in bit 9 to bit 0 the FWBL register after the access window information program command is executed.

This register is used to confirm the set value of the access window start address used for area protection.

### 41.4.25 Flash Access Window End Address Monitor Register (FAWEMR)

Address(es): 007F C0B4h

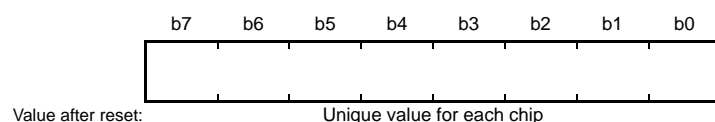


Note 1. The value of the blank product is 1. It is set to the same value set in bit 9 to bit 0 in the FWBH register after the access window information program command is executed.

This register is used to confirm the set value of the access window end address used for area protection.

### 41.4.26 Unique ID Register n (UIDRn) (n = 0 to 31)

Address(es): 0850h to 086Fh (extra area)



The UIDRn register stores a 32-byte ID code (unique ID) for identifying the individual MCU.

The unique ID is stored in the extra area of the flash memory and cannot be rewritten by the user. Issue the unique ID read command to read the register value.

### 41.5 Start-Up Program Protection

When rewriting the start-up program\*1 by self-programming, if the rewrite operation is interrupted due to temporary blackout, the start-up program may not be successfully programmed and the user program may not start properly.

This problem can be avoided by rewriting the start-up program without erasing the existing start-up program using the start-up program protection. This function is available in products with a 32-Kbyte or larger ROM.

Figure 41.3 shows the Overview of the Start-Up Program Protection. In this figure, the default area indicates block 0 to block 15, and the alternate area indicates block 16 to block 31.

Note 1. Program to perform operation to start the user program. It includes the fixed vector table.

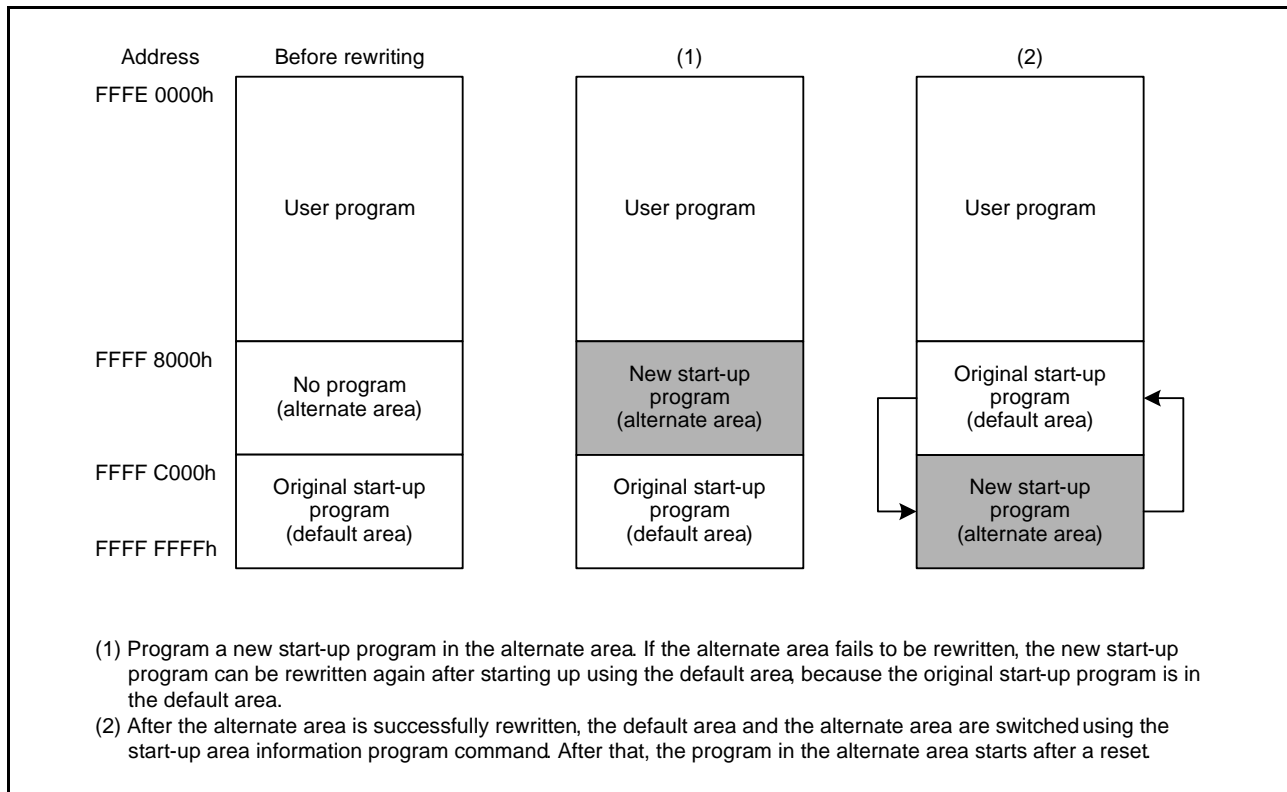


Figure 41.3 Overview of the Start-Up Program Protection

### 41.6 Area Protection

Area protection enables rewriting only the selected blocks (access window) in the user area and disables rewriting the other blocks during self-programming. The access window cannot be set in the data area.

Specify the start address and end address to set the access window. While the access window can be set in boot mode or by self-programming, area protection is enabled only during self-programming in single-chip mode.

Figure 41.4 shows the Area Protection Overview (Set Blocks 4 to 6 as the Access Window).

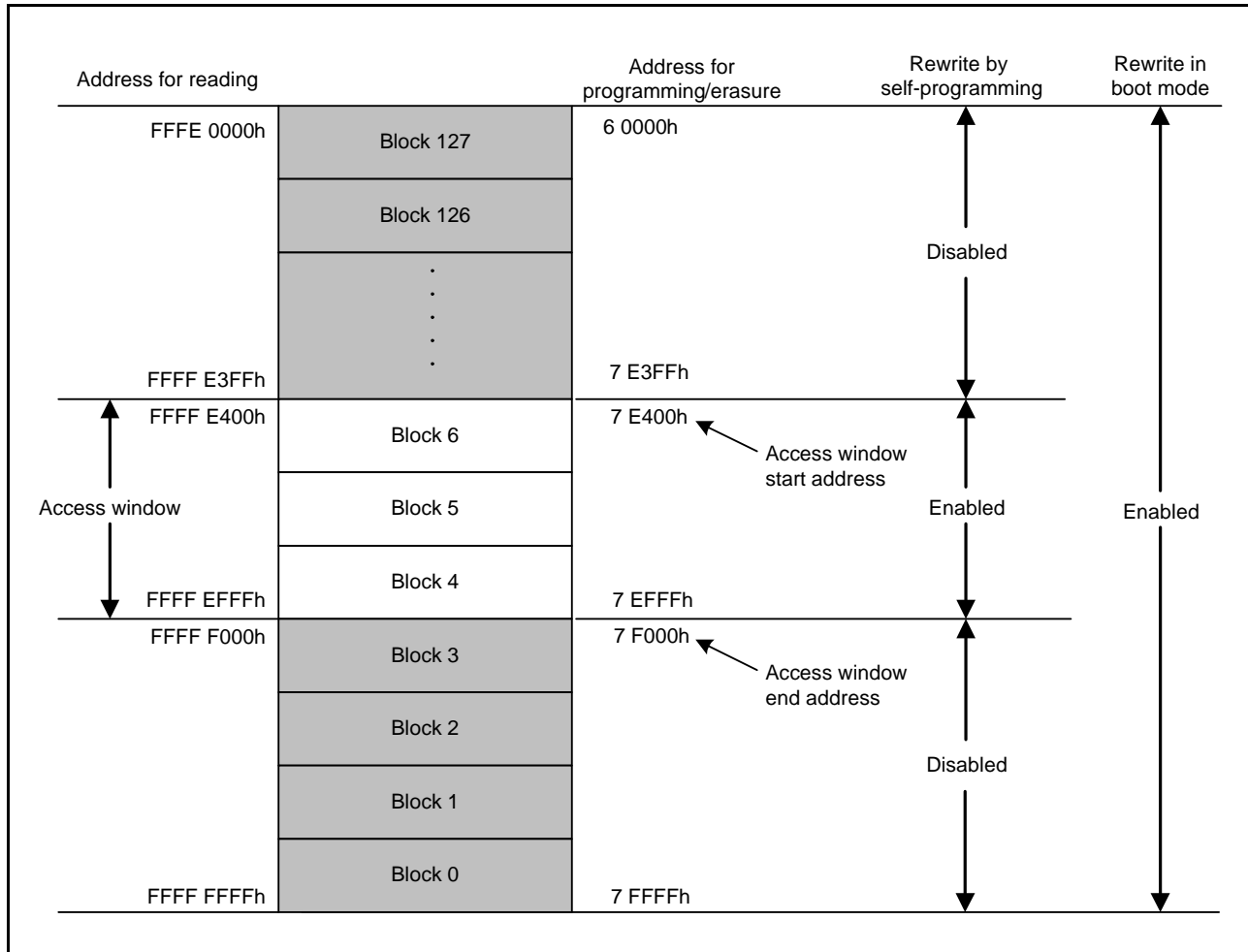


Figure 41.4 Area Protection Overview (Set Blocks 4 to 6 as the Access Window)

## 41.7 Programming and Erasure

The ROM and E2 DataFlash can be programmed and erased by changing the mode of the dedicated sequencer for programming and erasure, and by issuing commands for programming and erasure.

The mode transitions and commands required to program or erase the ROM and E2 DataFlash are described below. The descriptions apply in common to boot mode and single-chip mode.

### 41.7.1 Sequencer Modes

The sequencer has four modes. Transitions between modes are caused by writing to the DFLCTL and FENTRYR registers and setting the FPMCR register. Figure 41.5 is a diagram of mode transitions of the flash memory.

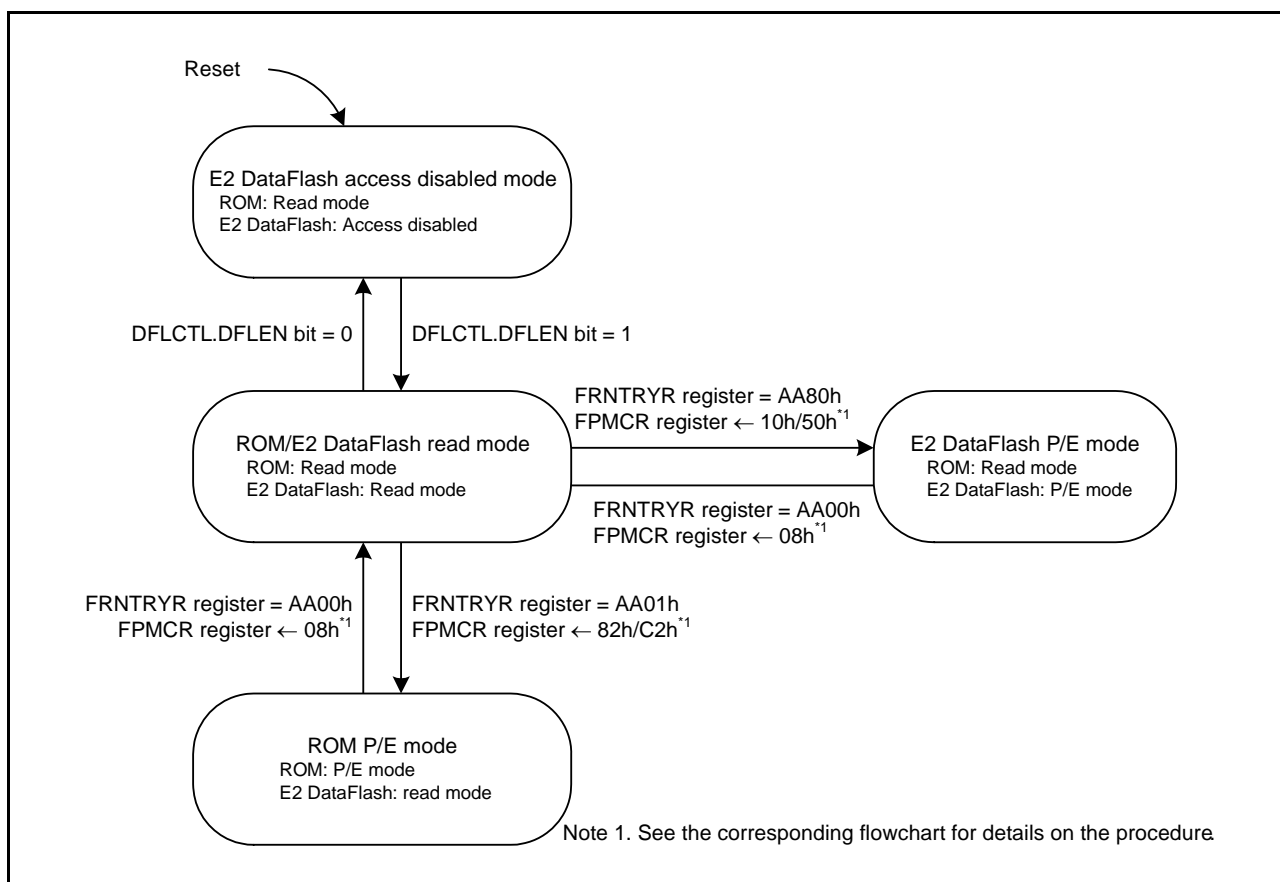


Figure 41.5 Mode Transitions of the Flash Memory

#### 41.7.1.1 E2 DataFlash Access Disabled Mode

In E2 DataFlash access disabled mode, access to the E2 DataFlash is disabled. After a reset, the sequencer enters this mode.

When setting the DFLCTL.DFLEN bit to 1, the E2 DataFlash is placed in read mode.

#### 41.7.1.2 Read Mode

Read mode is for high-speed reading of the ROM/E2 DataFlash. Reading from a ROM address for reading can be accomplished in one ICLK clock.



#### (1) ROM/E2 DataFlash Read Mode

In this mode, the ROM and E2 DataFlash are in read mode. The sequencer enters this mode when setting the FPMCR register to 08h, setting the FENTRYR.FENTRYD bit to 0, and setting the FENTRYR.FENTRY0 bit to 0.

### 41.7.1.3 P/E Modes

The P/E mode is for programming and erasure of the ROM/E2 DataFlash.

#### (1) ROM P/E Mode

In this mode, the ROM is in P/E mode, and the E2 DataFlash is in read mode. The sequencer enters this mode when the setting the FENTRYR.FENTRYD to 0, setting the FENTRYR.FENTRY0 bit to 1, and setting the FPMCR register 82h or C2h.

#### (2) E2 DataFlash P/E Mode

In this mode, the ROM is in read mode, and the E2 DataFlash is in P/E mode. The sequencer enters this mode when the setting the FENTRYR.FENTRYD to 1, setting the FENTRYR.FENTRY0 bit to 0, and setting the FPMCR register 10h or 50h.

## 41.7.2 Software Commands

Software commands consist of commands for programming and erasure and commands for programming start-up program area information and access window information. Table 41.4 lists the software commands for use with the flash memory.

**Table 41.4 Software Commands**

Command	Function
Program	<ul style="list-style-type: none"> <li>ROM programming (4 bytes)</li> <li>E2 DataFlash programming (1 byte)</li> </ul>
Block erase	ROM/E2 DataFlash erasure
Unique ID read	Read the unique ID in the extra area.
Blank check	Check whether the specified area is blank. Confirm that data is not programmed in the area. This command does not guarantee whether the area remains erased.
Start-up area information program	Rewrite the start-up area switching information used for start-up program protection.
Access window information program	Set the access window used for area protection.

## 41.7.3 Software Command Usage

The following describes each software command.

#### (1) Switching to ROM P/E Mode

A transition to ROM P/E mode is required before executing a software command for the ROM.

Setting the FENTRYR.FENTRY0 bit to 1 causes a transition to ROM P/E mode.

Figure 41.6 shows the Procedure for Transition to ROM P/E Mode.

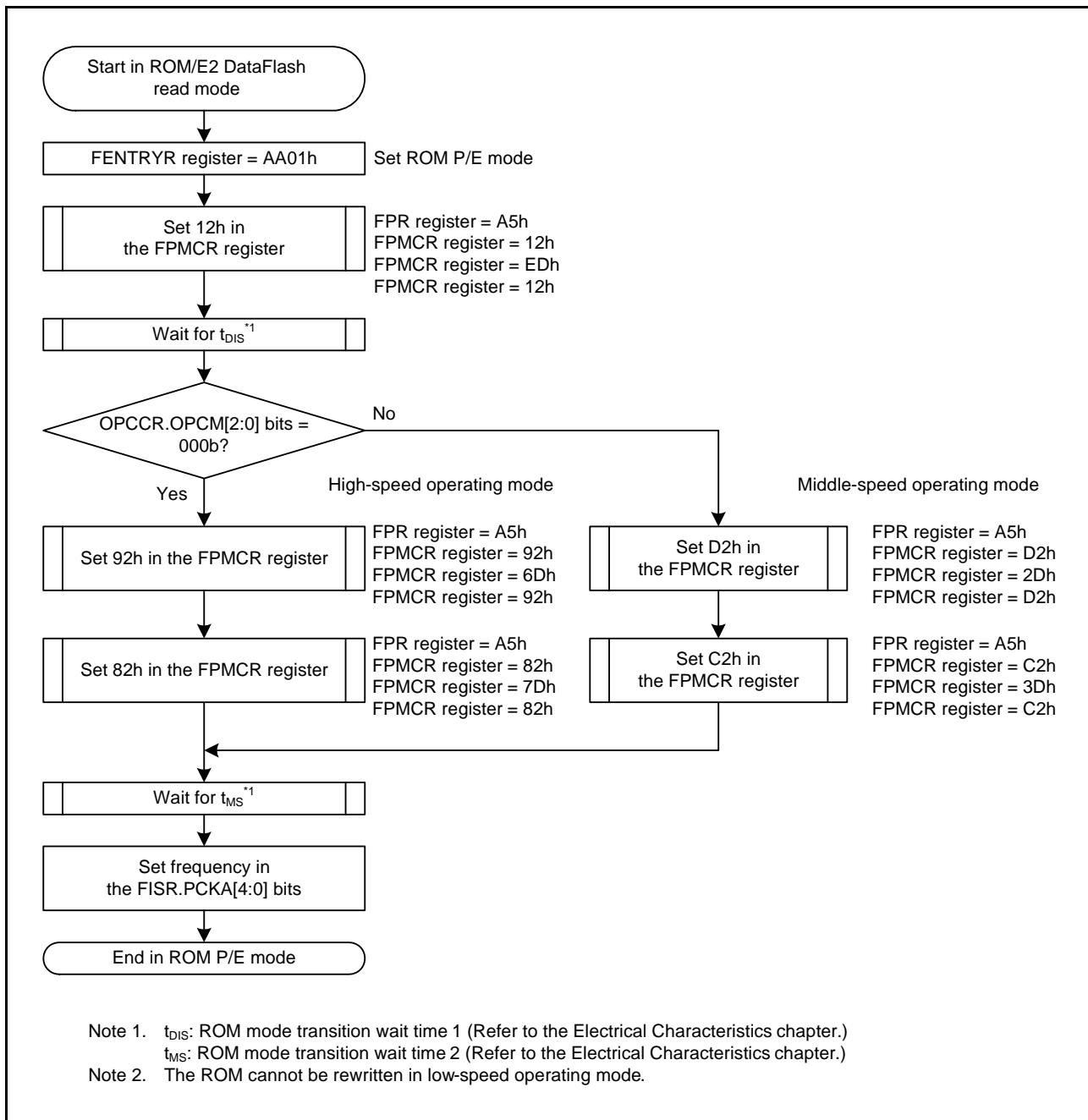


Figure 41.6 Procedure for Transition to ROM P/E Mode

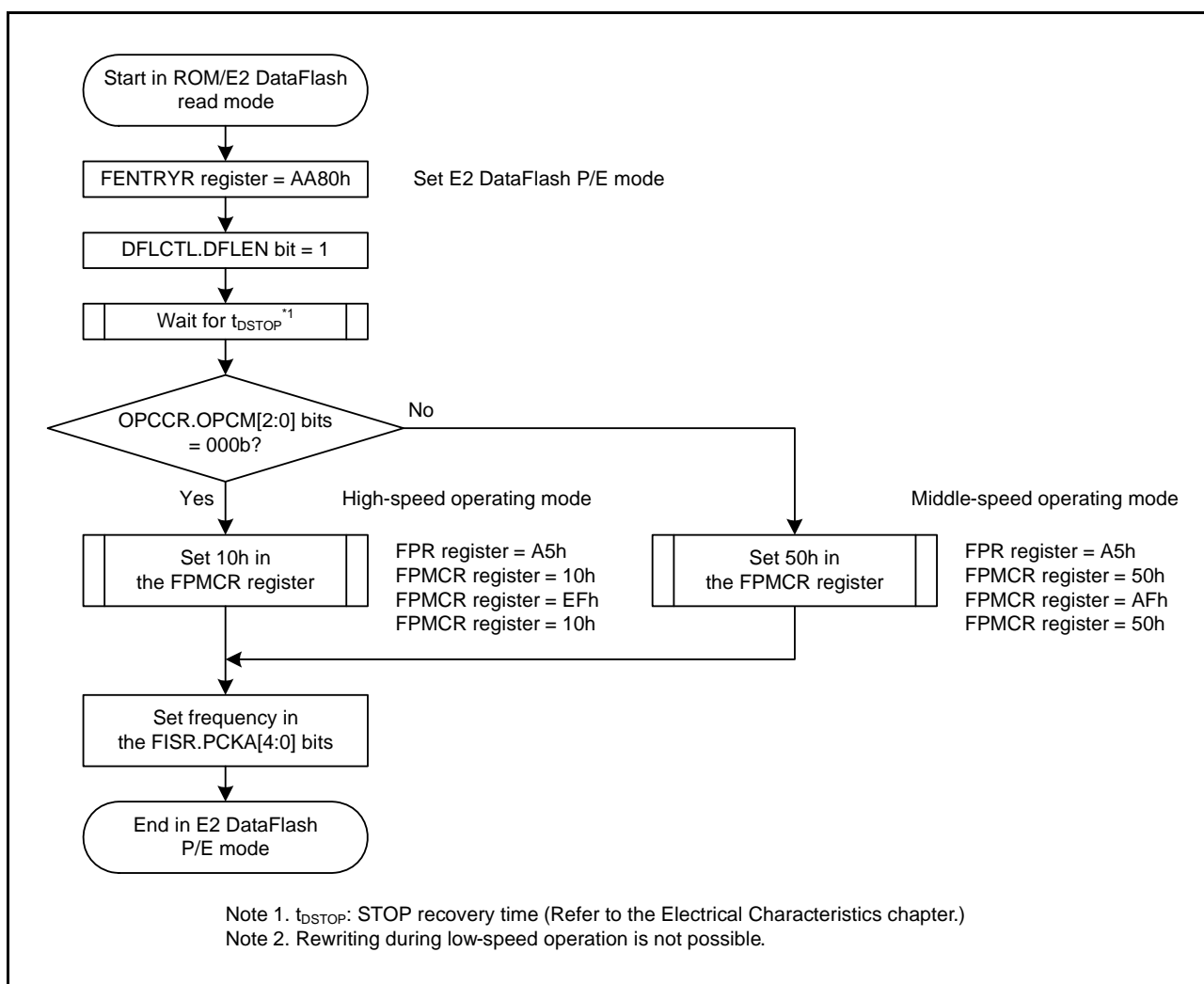


Figure 41.7 Procedure for Transition from ROM/E2 DataFlash Read Mode to E2 DataFlash P/E Mode

(2) Switching to ROM/E2 DataFlash Read Mode

High-speed reading of the ROM requires setting of the FENTRYR.FENTRY0 bit to 0, which places the sequencer in ROM/E2 DataFlash read mode.

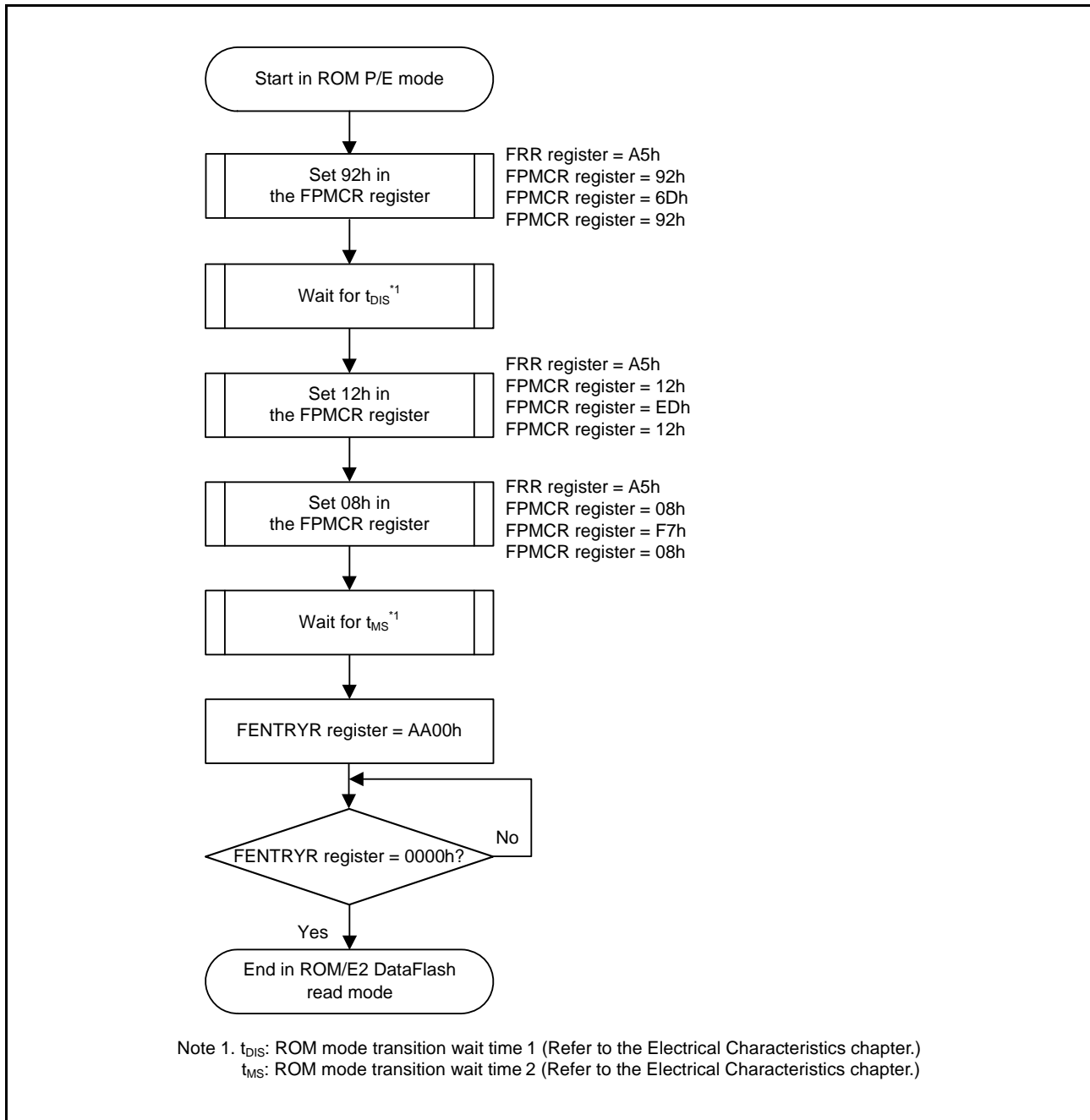


Figure 41.8 Procedure for Transition to ROM/E2 DataFlash Read Mode

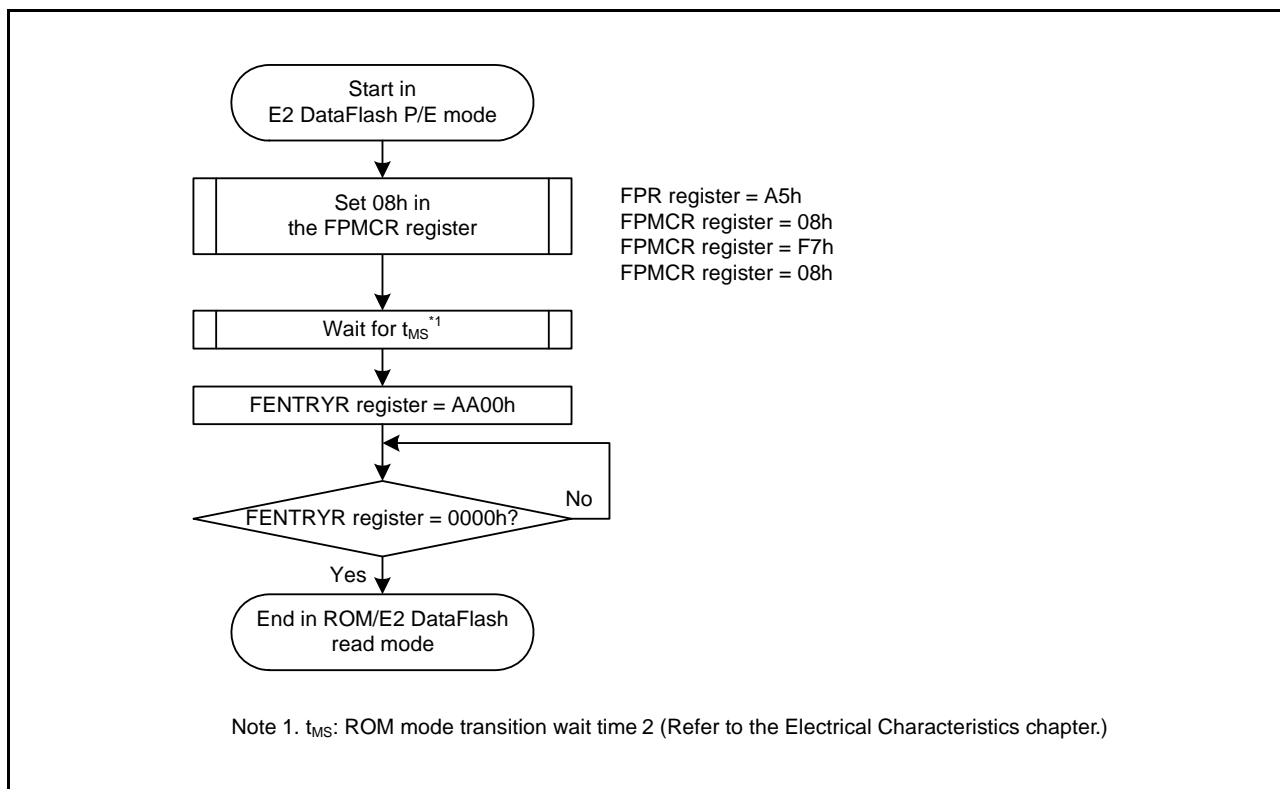


Figure 41.9 Procedure for Transition from E2 DataFlash P/E Mode to ROM/E2 DataFlash Read Mode

(3) Programming and Erasure Procedures

The following describes the flow of procedures for programming or erasing the ROM.

Figure 41.10 to Figure 41.15 are simple flowcharts of the procedure for programming the ROM.

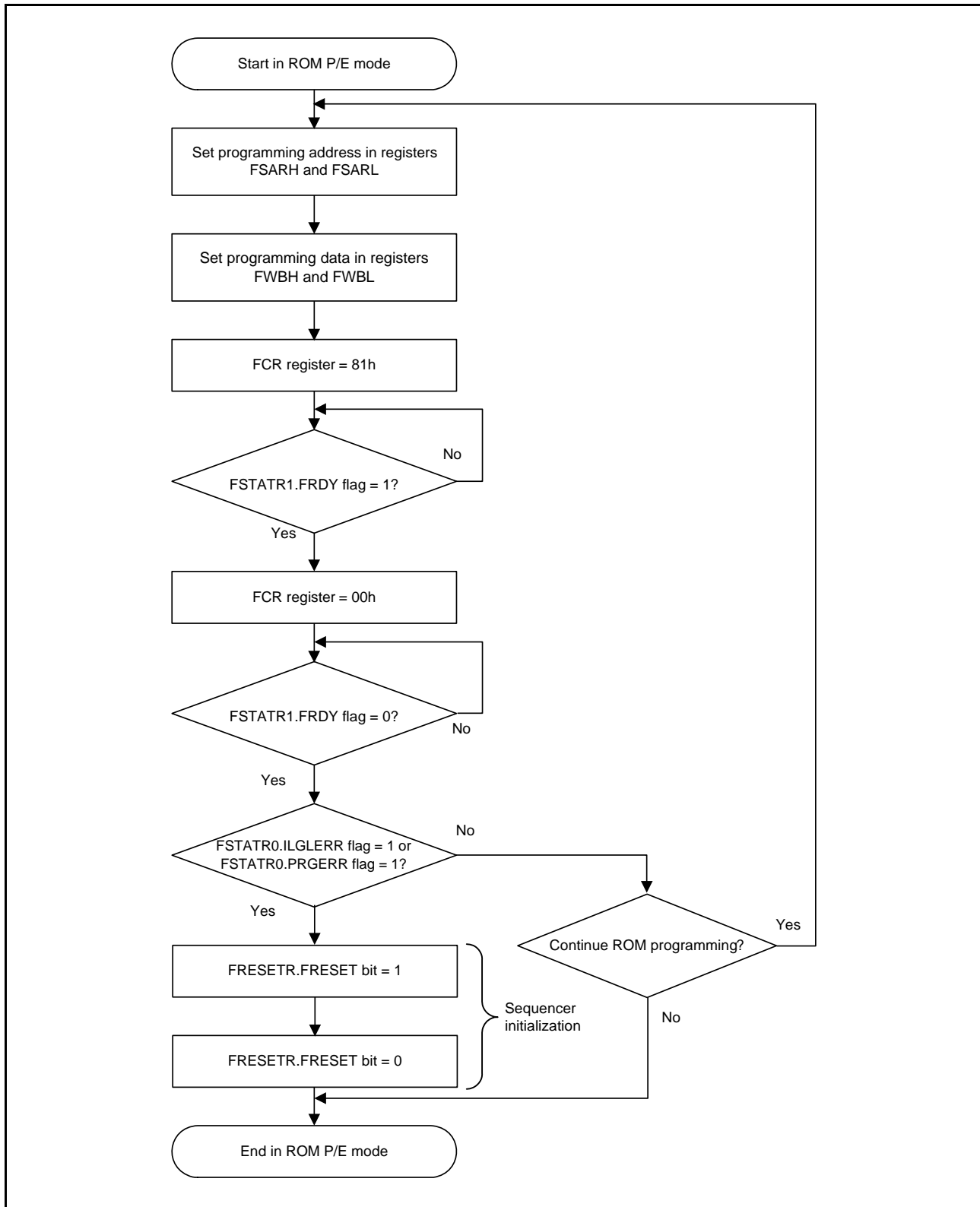


Figure 41.10 ROM Programming Procedure

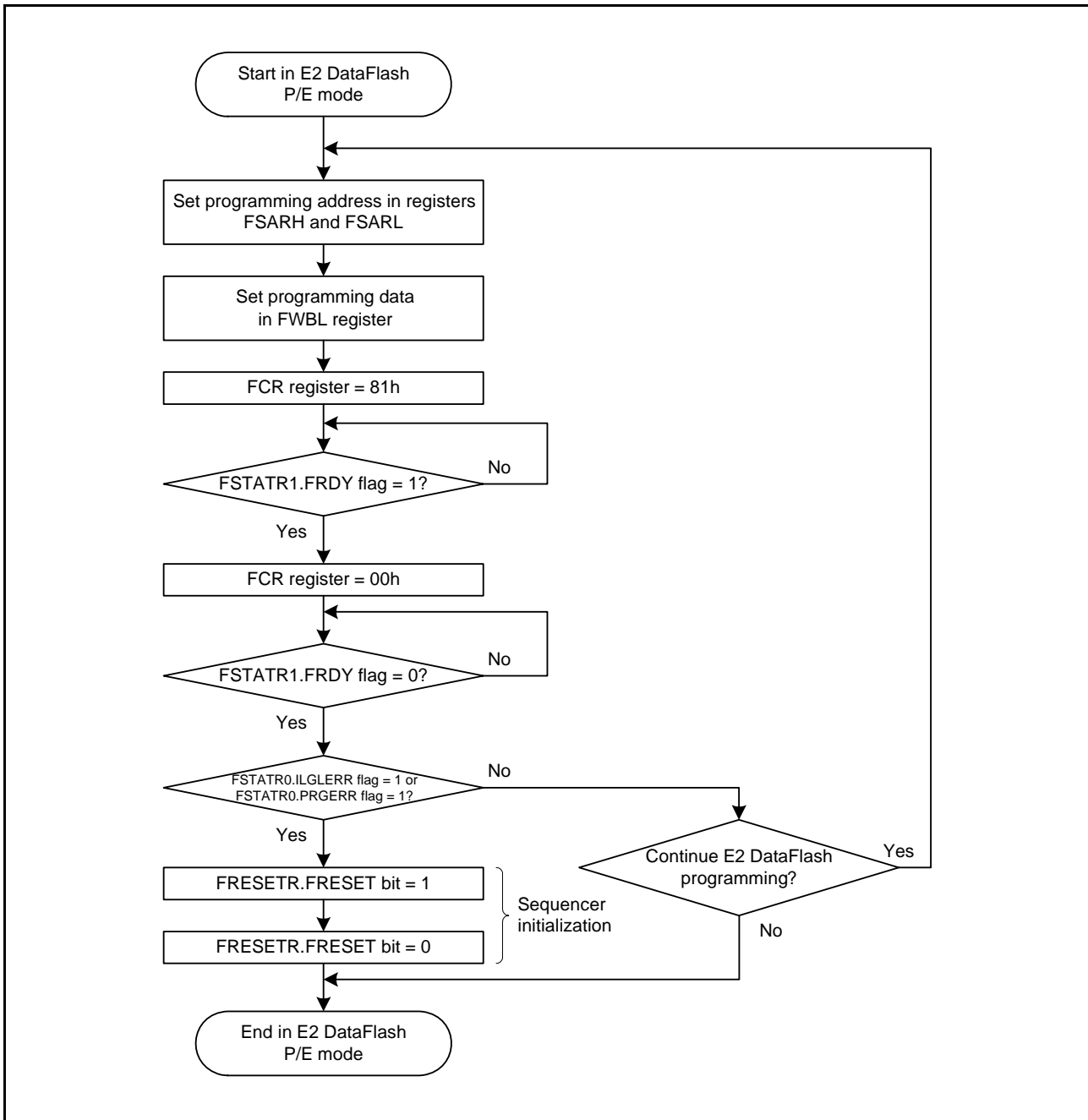


Figure 41.11 E2 DataFlash Programming Procedure

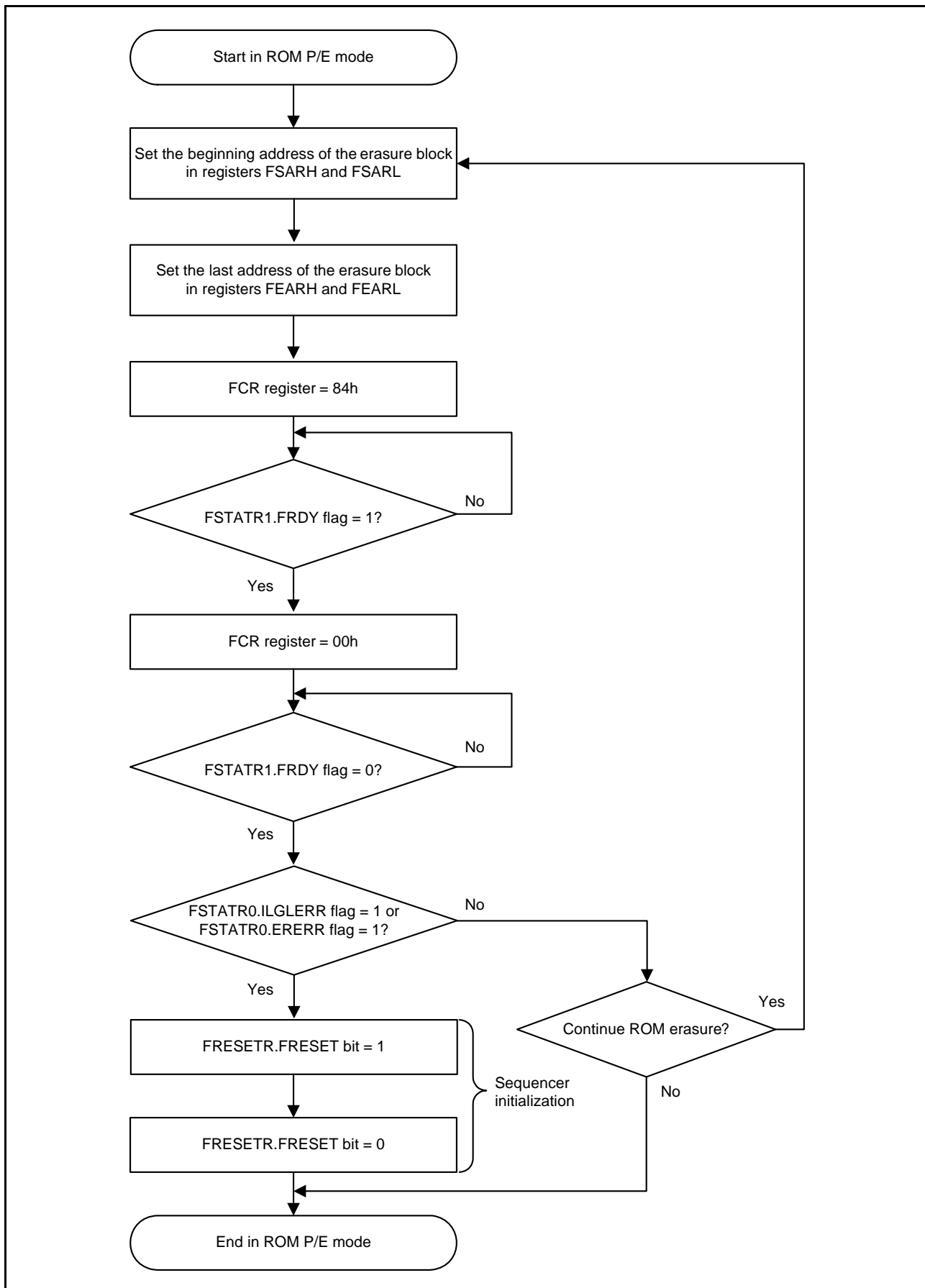


Figure 41.12 ROM Block Erase Procedure



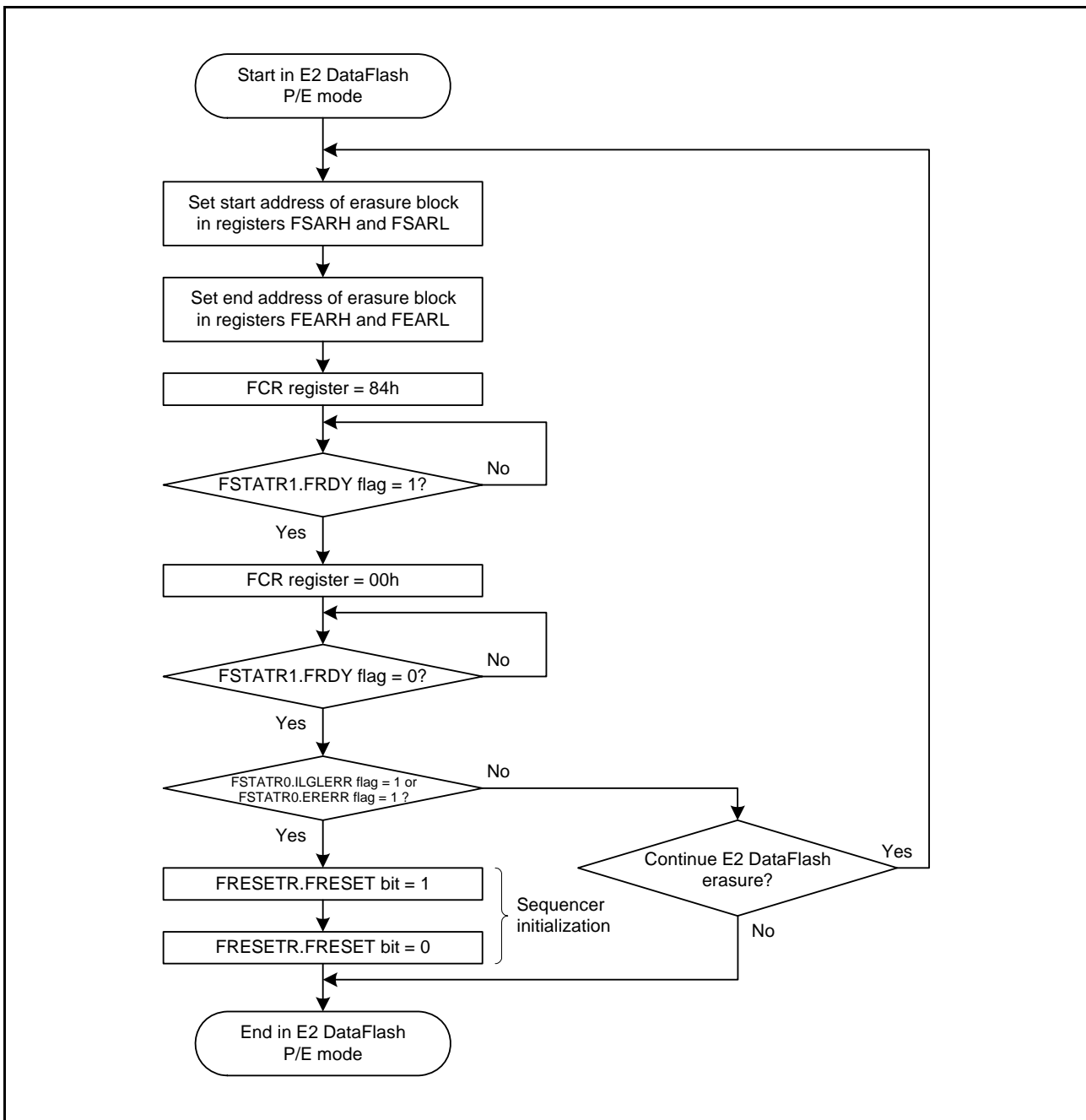


Figure 41.13 E2 DataFlash Block Erase Procedure

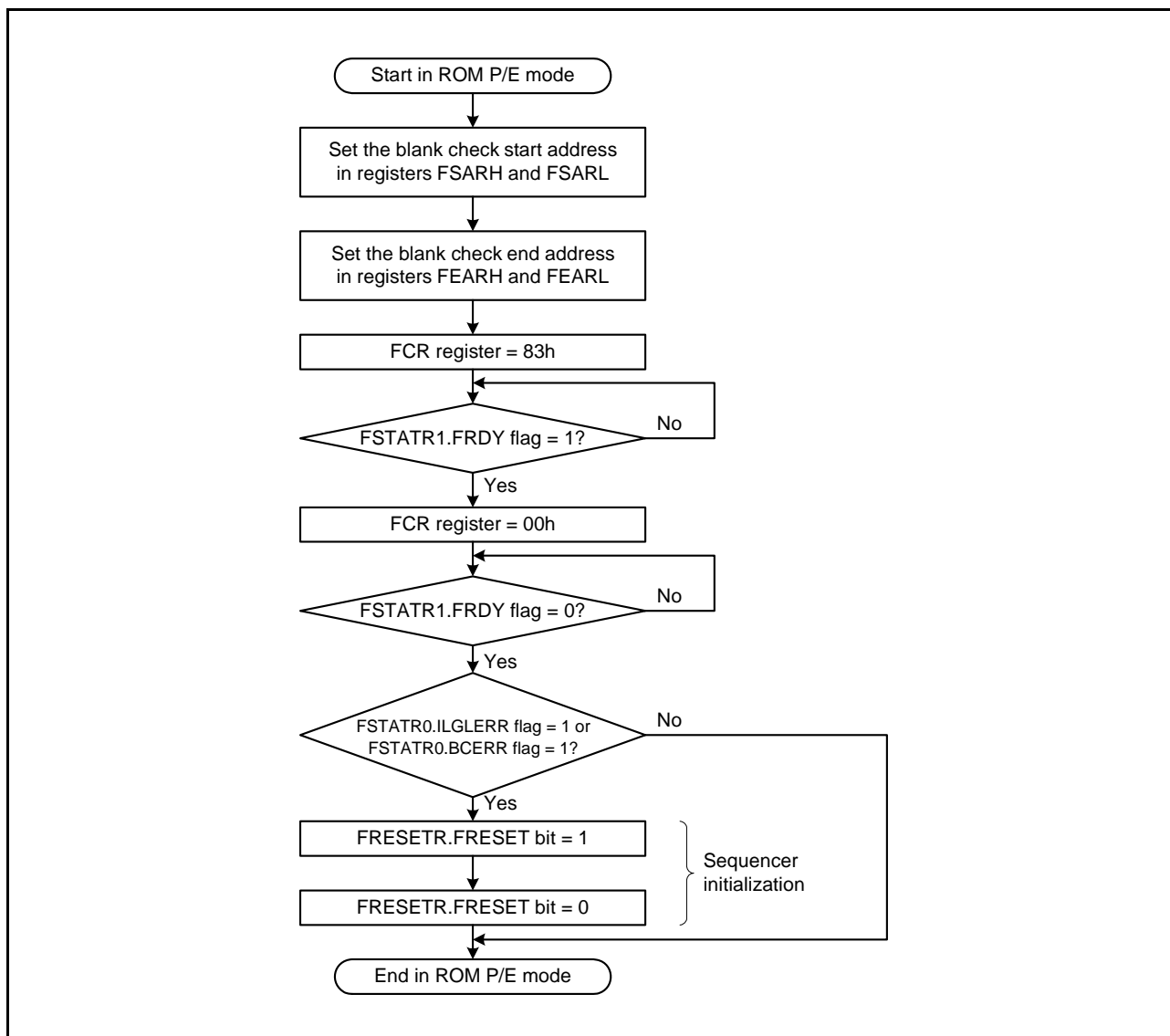


Figure 41.14 ROM Blank Check Procedure

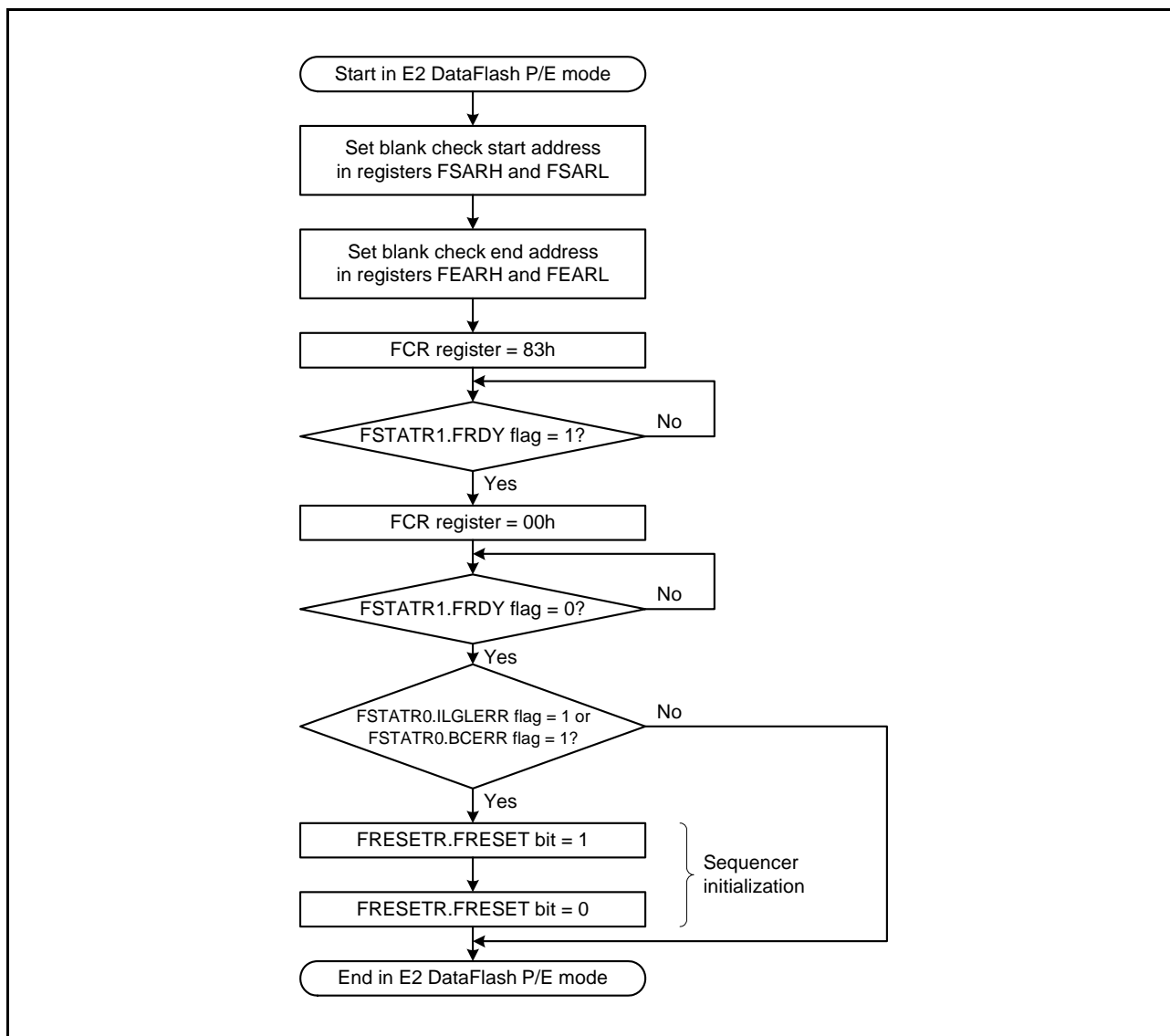


Figure 41.15 E2 DataFlash Blank Check Procedure

(4) Start-Up Area Information Program/Access Window Information Program

Figure 41.16 is a simple flowchart of the procedure for the start-up area information program/access window information program.

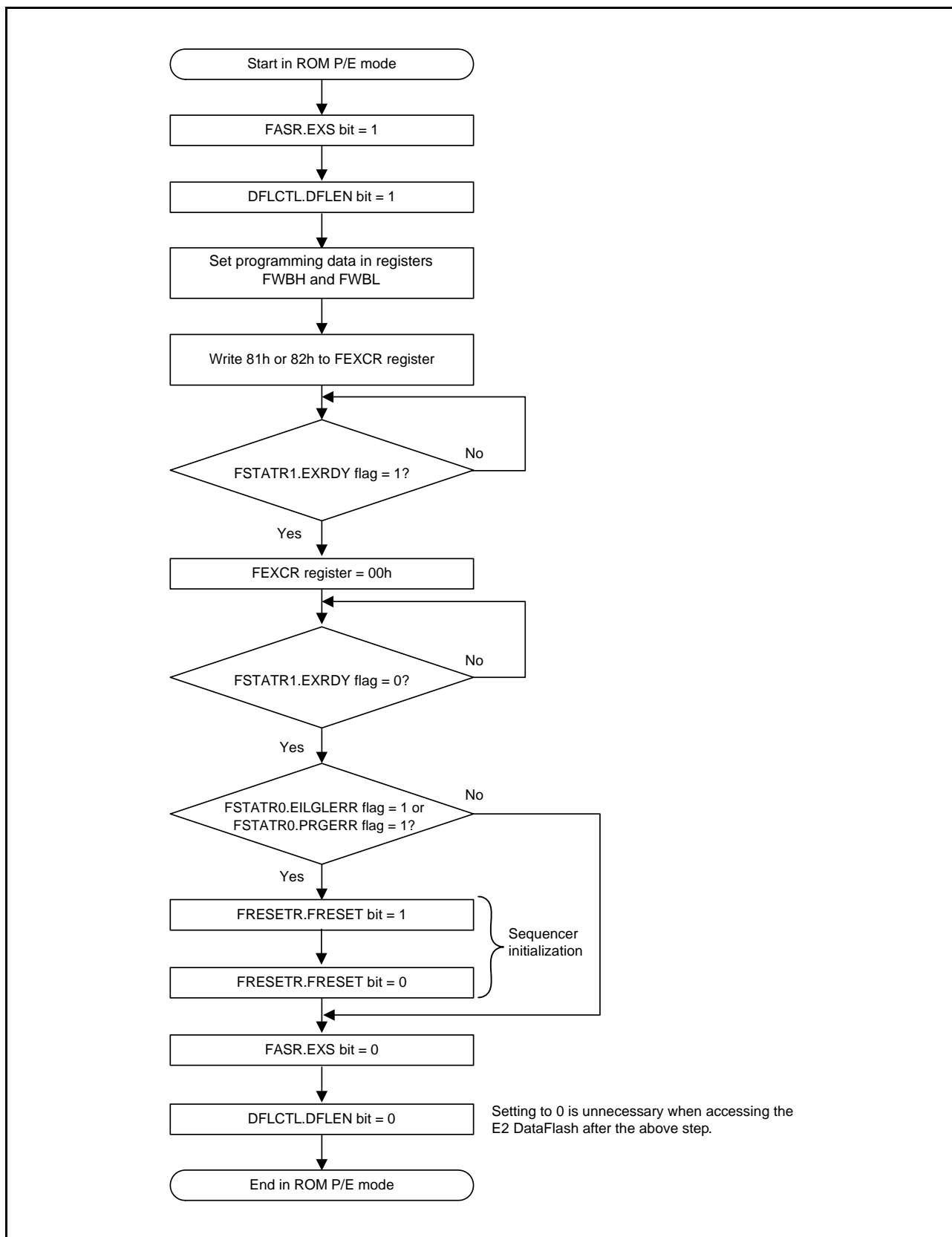


Figure 41.16 Procedure for the Start-Up Area Information Program/Access Window Information Program

(5) Unique ID Read

Figure 41.17 is a simple flowchart for the procedure for unique ID read.

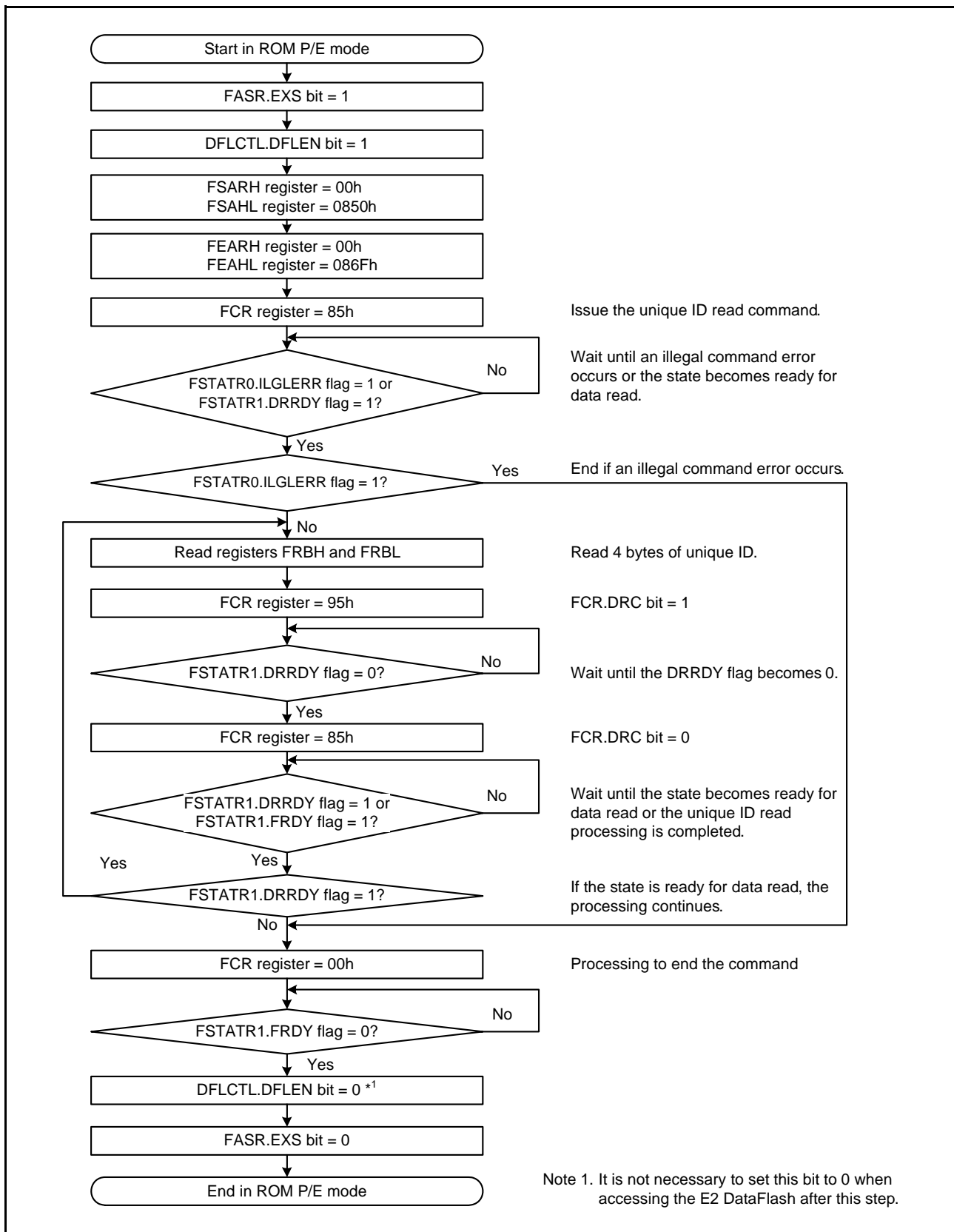


Figure 41.17 Simple Flowchart of the Procedure for Unique ID Read

## (6) Forced Stop of Software Commands

Perform the procedure shown in Figure 41.18 to forcibly stop the blank check command or block erase command. When the command processing is forcibly stopped, registers FEAMH and FEAML store the address at the time of the forced stop. For blank check, the stopped processing can be continued by copying the FEAMH and FEAML register values to registers FSARH and FSARL.

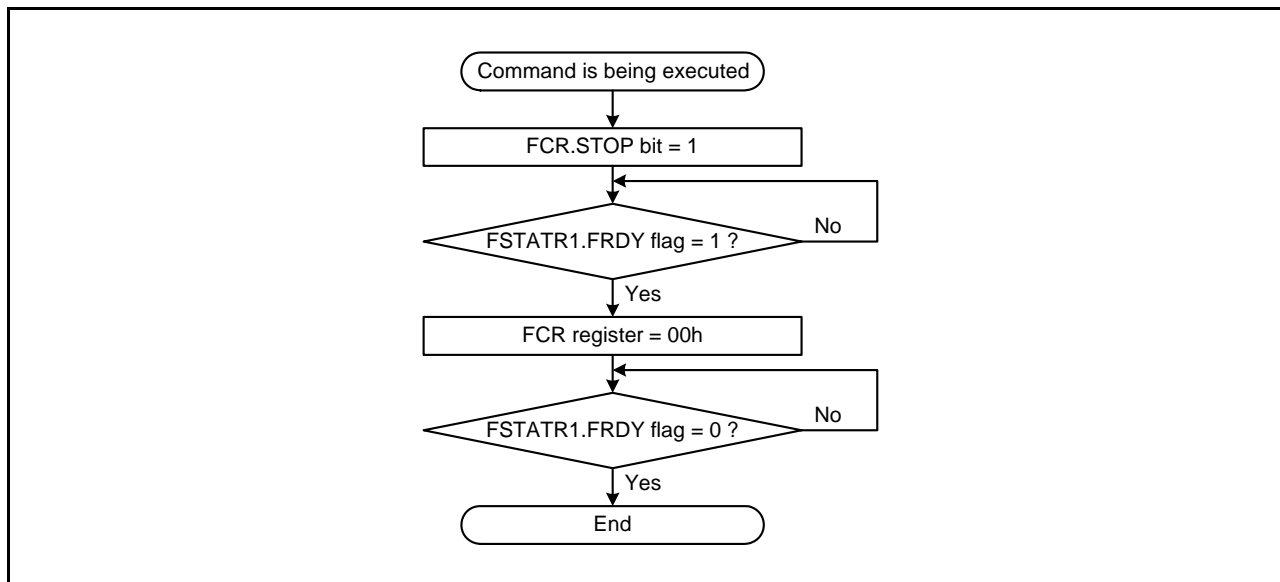


Figure 41.18 Procedure for Forced Stop of Software Commands

## 41.8 Boot Mode

The USB interface or SCI is used in boot mode.

Table 41.5 lists the Areas and Communication Functions Used in Boot Mode. Table 41.6 lists the I/O Pins Used in Boot Mode.

**Table 41.5 Areas and Communication Functions Used in Boot Mode**

Item	Boot Mode	
	USB Interface	SCI
Programmable and erasable areas	User area	User area
	Data area	Data area
Communication function	USB0	SCI1 (asynchronous serial communication)

**Table 41.6 I/O Pins Used in Boot Mode**

Pin Name	I/O	Mode	Description
P14/UB#*1	Input	Boot mode	Select operating mode (refer to section 3., Operating Modes).
MD	Input		Select operating mode (refer to section 3., Operating Modes).
USB0_DP, USB0_DM	I/O	USB interface	USB data I/O
P16/USB0_VBUS	Input		Detect USB cable connection/disconnection
P35	Input		Set bus-powered mode or self-powered mode
P15/RXD1	Input	Boot mode (SCI)	Receive data through SCI1 for programmer communication*2
P16/TXD1	Output		Transmit data through SCI1 for programmer communication*2

Note 1. This pin becomes high for a certain time by the on-chip input pull-up resistor after starting up in boot mode.

Note 2. When using the SCI, connect (pull up) this pin to VCC via a resistor.

### 41.8.1 Boot Mode (USB Interface)

The user area and data area can be programmed and erased using USB communication in boot mode (USB interface). Self power or bus power can be selected in boot mode (USB interface) in accordance of the state of the UPSEL pin.

When a reset is released while pins MD, UB#, and UPSEL are low, self power is selected. When a reset is released while pins MD and UB# are low and UPSEL is high, bus power is selected.

Contact the manufacturer for details on the dedicated flash memory programmer (USB programmer).

#### 41.8.1.1 System Configuration in Boot Mode (USB Interface)

USB0 is used for communication with the dedicated flash memory programmer in boot mode (USB interface).

Prepare tools for transmitting and receiving control commands and state via the USB and data for programming.

6, 8, 12, or 16 MHz can be used as the frequency input to the main clock oscillator. The operating voltage range is 3.0 to 3.6 V.

Connect the UB# pin to VSS on the dedicated flash memory programmer, or connect to VSS via a resistor (pull down).

For pull down, connect a resistor of up to 2.5 k $\Omega$

Figure 41.19 shows the Example of Pin Connections in Boot Mode (USB Interface) When Self-Powered. Table 41.7 lists Pin Handling in Boot Mode (USB Interface) When Self-Powered. Figure 41.20 shows the Example of Pin Connections in Boot Mode (USB Interface) When Bus-Powered. Table 41.8 lists Pin Handling in Boot Mode (USB Interface) in Bus-Powered Mode.

Pins in Figure 41.19 show only those pins associated with boot mode (USB interface). Operations are not guaranteed in all systems.

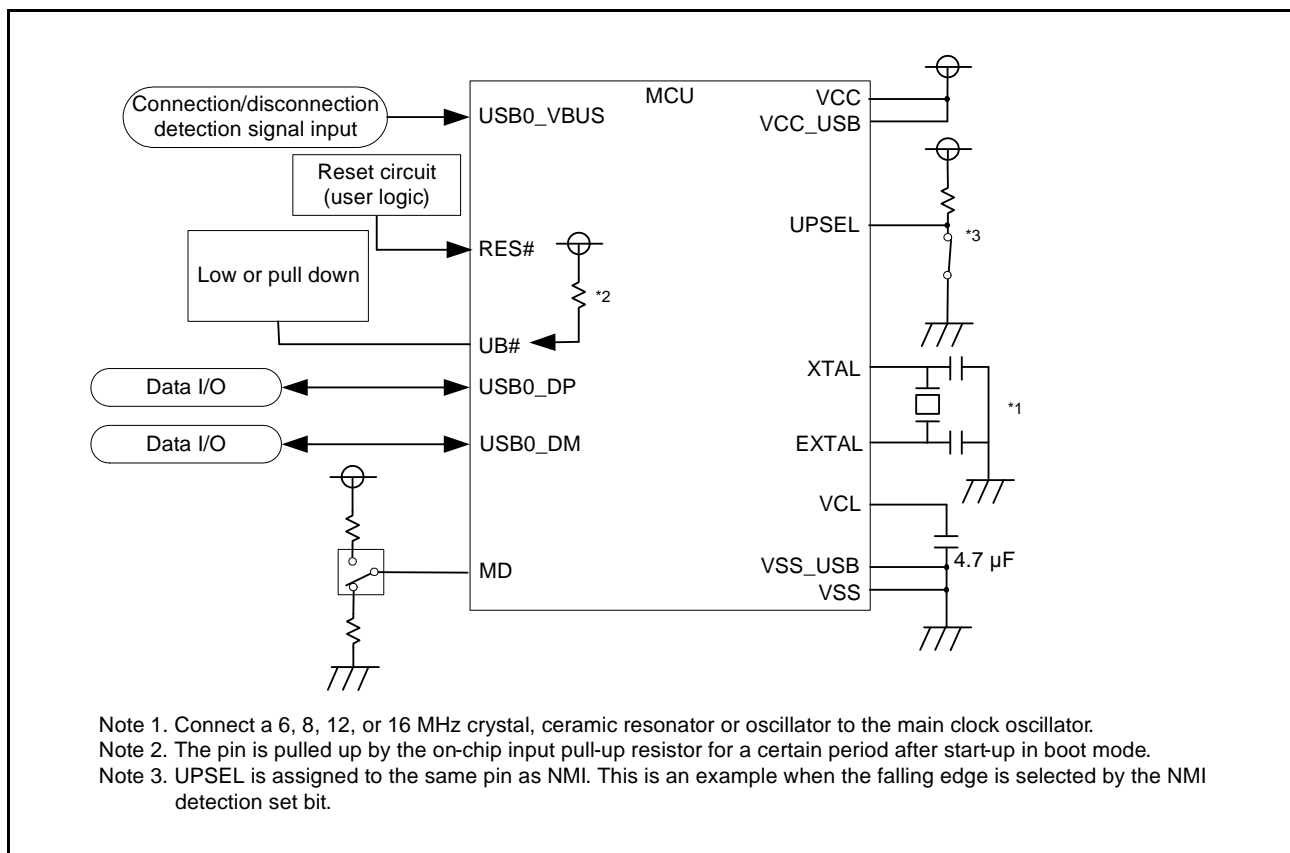


Figure 41.19 Example of Pin Connections in Boot Mode (USB Interface) When Self-Powered



**Table 41.7 Pin Handling in Boot Mode (USB Interface) When Self-Powered**

Pin Name	Name	I/O	Function
VCC, VSS	Power supply input	Input	Input the guaranteed voltage for program/erase to the VCC pin. Input 0 V to the VSS pin.
VCC_USB, VSS_USB	USB power supply input	Input	Connect the VCC_USB pin to the VCC pin. Connect the VSS_USB pin to the VSS pin.
AVCC0, AVSS0	12-bit A/D converter power supply input	Input	Connect the AVCC0 pin to the VCC pin. Connect the AVSS0 pin to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor (4.7 $\mu$ F) for stabilizing the internal voltage.
XTAL, EXTAL	Main clock I/O pin	I/O	Connect a 6 MHz, 8 MHz, 12 MHz, or 16 MHz crystal or ceramic resonator or oscillator.
XCIN, XCOUT	Sub-clock I/O pin	I/O	Input high or low, or leave open.
MD	Operating mode control	Input	Connect the VSS pin, or connect the VSS pin via a resistor (pull down).
P14/UB#	Operating mode control	Input	Connect the VCC pin, or connect the VCC pin via a resistor (pull up). For pull up, connect a resistor of up to 2.5 k $\Omega$ .
P35/UPSEL	USB power mode control	Input	Connect the VSS pin, or connect the VSS pin via a resistor (pull down).
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.
USB0_DP	USB on-chip transceiver D+ I/O pin	I/O	Connect the circuit described in section 26, USB 2.0 Host/Function Module (USBc).
USB0_DM	USB on-chip transceiver D- I/O pin	I/O	Connect the circuit described in section 26, USB 2.0 Host/Function Module (USBc).
P16/USB0_VBUS	USB cable connection monitor pin	Input	Connect the circuit described in section 26, USB 2.0 Host/Function Module (USBc).
Ports other than the above	Input port	Input	Input high or low, or leave open.

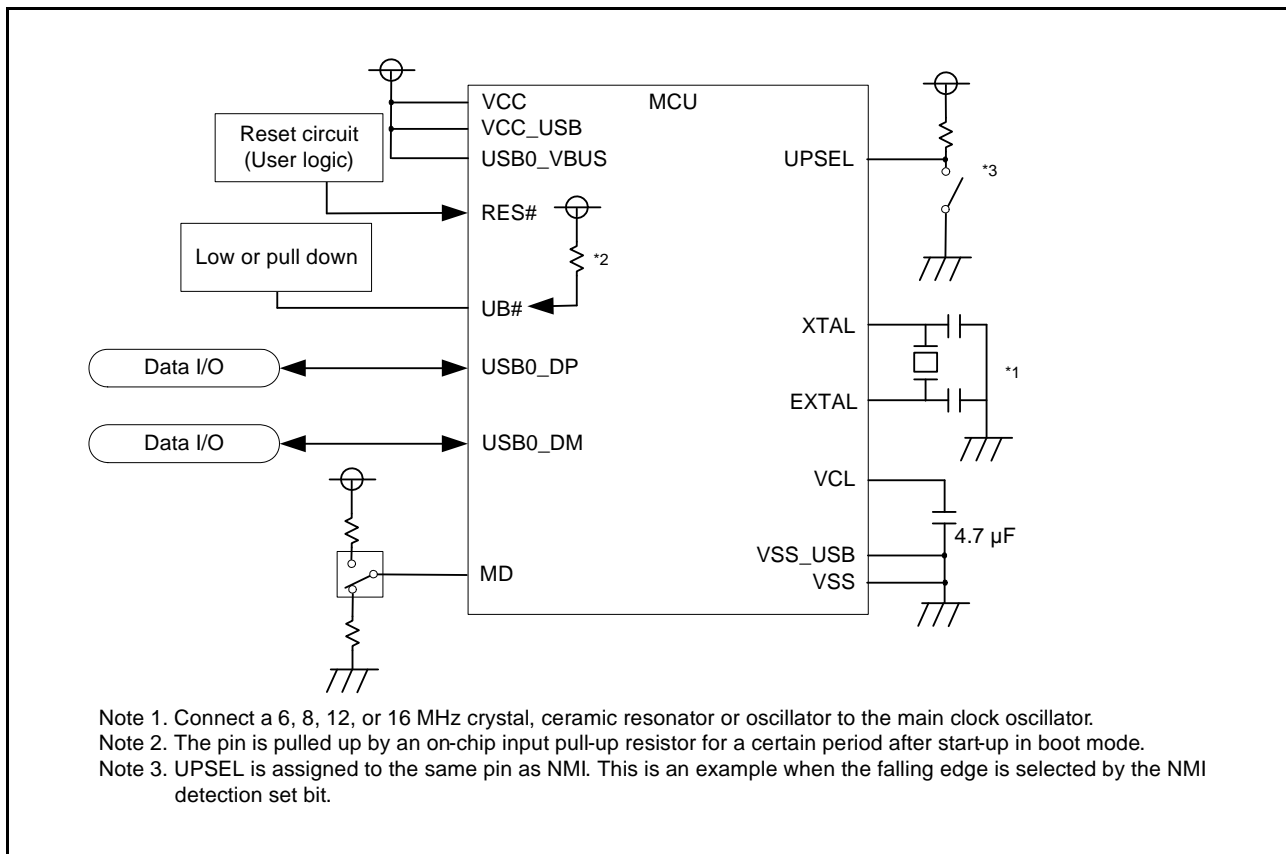


Figure 41.20 Example of Pin Connections in Boot Mode (USB Interface) When Bus-Powered

**Table 41.8 Pin Handling in Boot Mode (USB Interface) in Bus-Powered Mode**

Pin Name	Name	I/O	Function
VCC, VSS	Power supply input	Input	Input the guaranteed voltage for program/erase to the VCC pin. Input 0 V to the VSS pin.
VCC_USB, VSS_USB	USB power supply input	Input	Connect the VCC_USB pin to the VCC pin. Connect the VSS_USB pin to the VSS pin.
AVCC0, AVSS0	12-bit A/D converter power supply input	Input	Connect the AVCC0 pin to the VCC pin. Connect the AVSS0 pin to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor (4.7 $\mu$ F) for stabilizing the internal voltage.
XTAL, EXTAL	Main clock I/O pin	I/O	Connect a 6, 8, 12, or 16 MHz crystal or ceramic resonator or oscillator.
XCIN, XCOU	Sub-clock I/O pin	I/O	Input high or low, or leave open.
MD	Operating mode control	Input	Connect the VSS pin, or connect the VSS pin via a resistor (pull down).
P14/UB#	Operating mode control	Input	Connect the VCC pin, or connect the VCC pin via a resistor (pull up). For pull down, connect a resistor of up to 2.5 k $\Omega$ .
P35/UPSEL	USB power mode control	Input	Connect the VCC pin, or connect the VCC pin via a resistor (pull up).
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.
USB0_DP	USB on-chip transceiver D+ I/O pin	I/O	Connect the circuit described in section 26, USB 2.0 Host/Function Module (USBc).
USB0_DM	USB on-chip transceiver D- I/O pin	I/O	Connect the circuit described in section 26, USB 2.0 Host/Function Module (USBc).
P16/USB0_VBUS	USB cable connection monitor pin	Input	Connect the USB0_VBUS pin to the VCC pin.
Ports other than the above	Input port	Input	Input high or low, or leave open.

## 41.8.2 Boot Mode (SCI)

The user area and data area can be programmed and erased using asynchronous serial communication in boot mode (SCI). Contact the manufacturer for details on the dedicated flash memory programmer (SCI programmer).

### 41.8.2.1 System Configuration in Boot Mode (SCI)

SCI1 is used to communicate with the programmer in boot mode (SCI).

Prepare tools for transmitting/receiving control commands and status via asynchronous serial communication, and data for programming in the dedicated flash memory programmer. Figure 41.21 shows the Example of Pin Connections in Boot Mode (SCI). Table 41.9 lists Pin Handling in Boot Mode (SCI).

Pins in Figure 41.21 show only those pins associated with boot mode (SCI). Operations are not guaranteed in all systems.

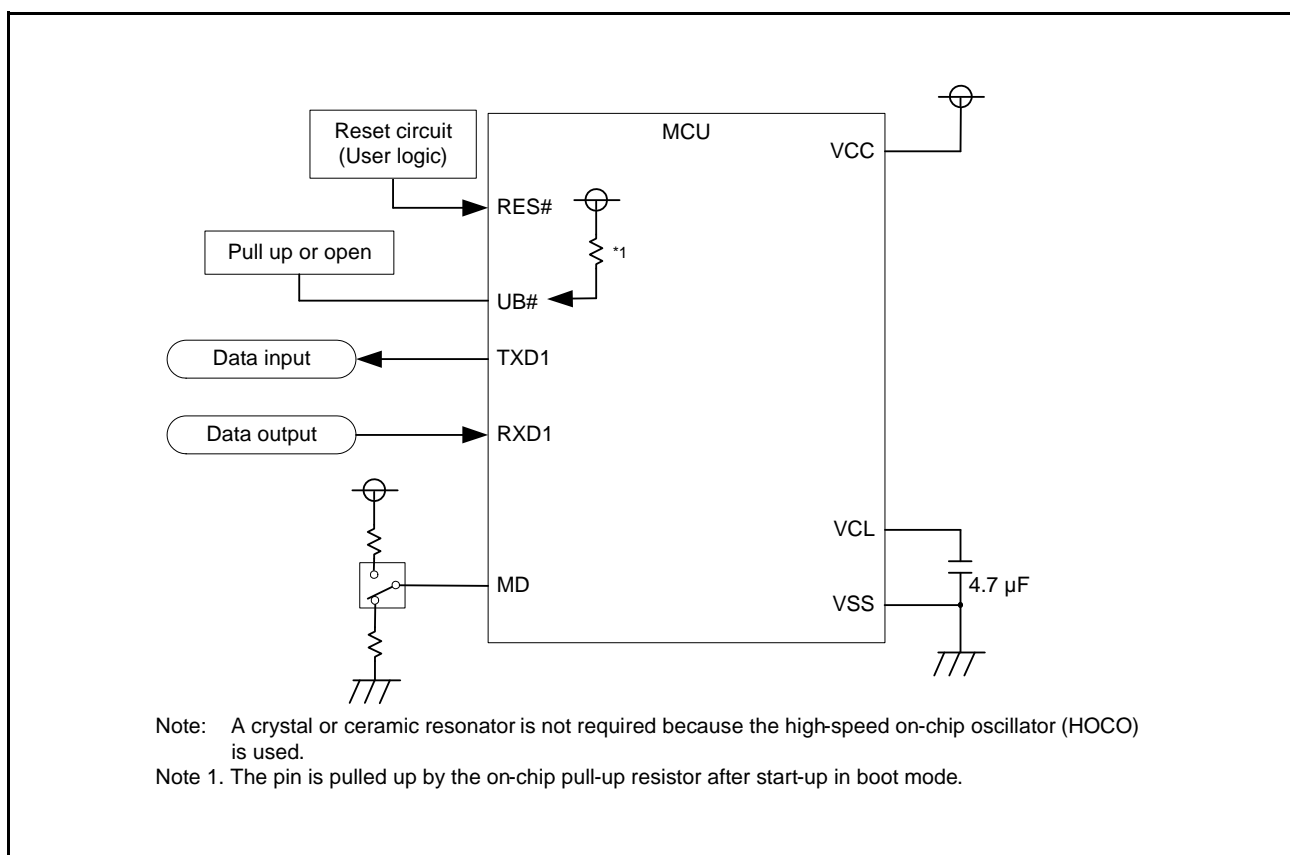
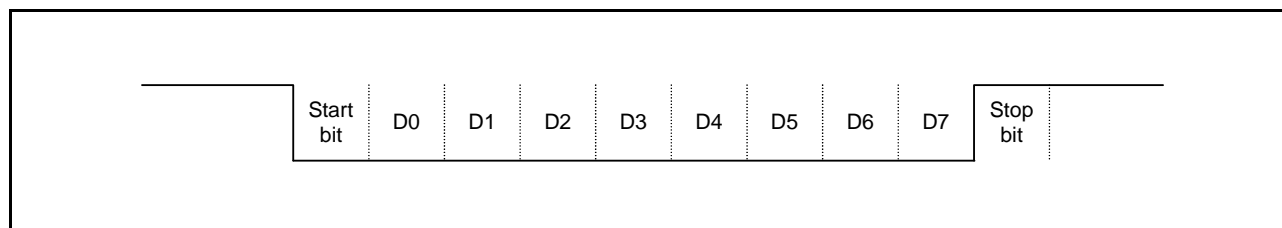


Figure 41.21 Example of Pin Connections in Boot Mode (SCI)

**Table 41.9 Pin Handling in Boot Mode (SCI)**

Pin Name	Name	I/O	Function
VCC, VSS	Power supply input	Input	Input the guaranteed voltage for program/erase to the VCC pin. Input 0 V to the VSS pin.
VCC_USB, VSS_USB	USB power supply input	Input	Connect the VCC_USB pin to the VCC pin. Connect the VSS_USB pin to the VSS pin.
AVCC0, AVSS0	12-bit A/D converter power supply input	Input	Connect the AVCC0 pin to the VCC pin. Connect the AVSS0 pin to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor (4.7 $\mu$ F) for stabilizing the internal voltage.
XTAL, EXTAL	Main clock I/O pin	I/O	Input high or low, or leave open.
XCIN, XCOU	Sub-clock I/O pin	I/O	Input high or low, or leave open.
MD	Operating mode control	Input	Connect the VSS pin, or connect the VSS pin via a resistor (pull down).
P14/UB#	Operating mode control	Input	Connect the VCC pin via a resistor (pull up), or leave open.
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.
P15/RXD1	Operating mode control	Input	Input pin for Input serial data
P6/TXD1	Operating mode control	Input	Output pin for output serial data
USB0_DP	USB on-chip transceiver D+ I/O pin	I/O	Leave the pin open.
USB0_DM	USB on-chip transceiver D- I/O pin	I/O	Leave the pin open.
Ports other than the above	Input port	Input	Input high or low, or leave the pin open.

As shown in Figure 41.22, set the format to 8-bit data, 1 stop bit, no parity, and LSB first to communicate with the dedicated flash memory programmer.

**Figure 41.22 Communication Format**

Communication with the programmer is performed at 9,600 bps or 19,200 bps. The communication bit rate can be changed after the MCU is connected with the programmer.

Table 41.10 lists the maximum communication bit rates for communication in boot mode (SCI).

**Table 41.10 Conditions for Communication**

Operating Voltage	Maximum Communication Bit Rate
1.8 V or higher, and lower than 3.0 V	500 kbps
3.0 V or higher, and 3.6 V or lower	2 Mbps

### 41.8.2.2 Starting Up in Boot Mode (SCI)

To start up in boot mode (SCI), release the reset (drive the RES# pin high from low) while the MD pin is low and the UB# pin is high or open. After starting up in boot mode (SCI), wait at least 400 ms holding the RES# pin high until communication is enabled in boot mode (SCI).

Figure 41.23 shows the pin states until communication is enabled in boot mode (SCI). Use resets according to the range described in section 42.3.2, Reset Timing.

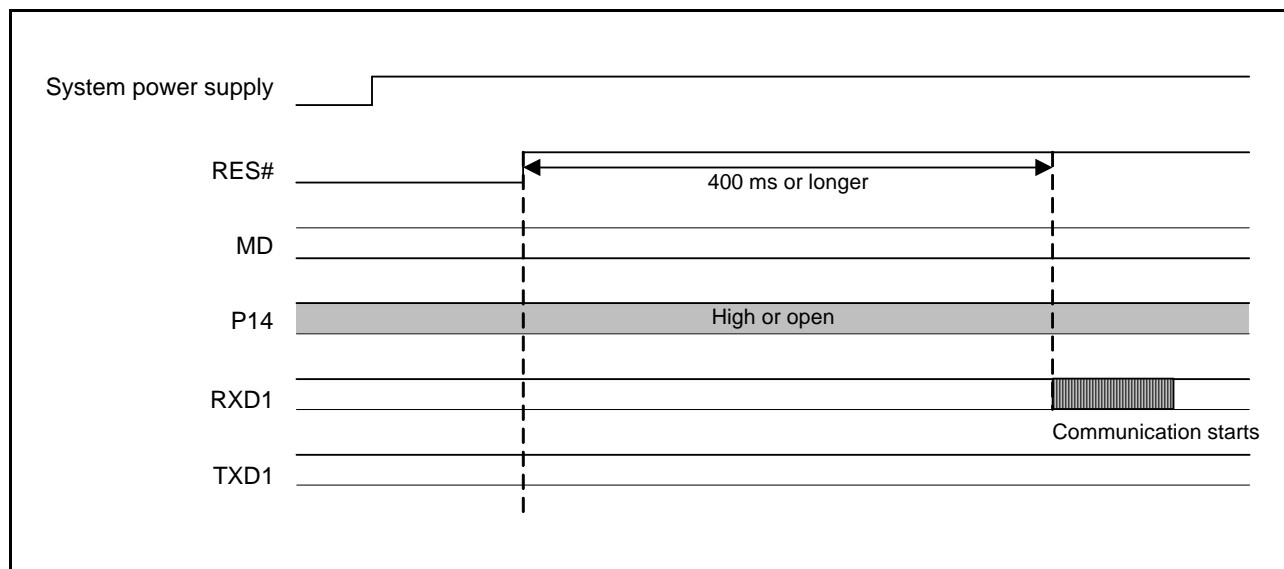


Figure 41.23 Pin States until Communication Becomes Possible in Boot Mode (SCI)

## 41.9 Flash Memory Access Disable Function

The flash memory access disable function disables reading and programming of the flash memory. The boot mode ID code protection is for boot mode, and the on-chip debugging emulator ID code protection is for the on-chip debugging emulator. Details are below.

### 41.9.1 ID Code

ID codes are used for boot mode ID code protection in boot mode, and on-chip debugging emulator ID code protection in the on-chip debugging emulator.

ID codes consist of the control code and ID code 1 to ID code 15. Set ID codes in 32-bit units. Figure 41.24 shows the ID Code Configuration.

	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 41.24 ID Code Configuration**

The following shows a program example for setting ID codes

This is an example when setting the control code to 45h and setting ID codes to 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, and 0Fh (from the ID code 1 field to the ID code 15 field).

C language:

```
#pragma address ID_CODE = 0xFFFFF0A0
const unsigned long ID_CODE [4] = {0x45010203, 0x04050607, 0x08090A0B, 0x0C0D0E0F};
```

Assembly language:

```
.SECTION ID_CODE, CODE
.ORG 0xFFFFF0A0
.LWORD 45010203h
.LWORD 04050607h
.LWORD 08090A0Bh
.LWORD 0C0D0E0Fh
```

### 41.9.1.1 Boot Mode ID Code Protection

Boot mode ID code protection disables reading and programming of the user area and data area.

When the control code indicates that the boot mode ID code protection is disabled while the user area and data area are blank, the user area and data area can be read and programmed.

When the control code indicates that the boot mode ID code protection is disabled while the user area and data area have data, the MCU enters the erase ready state so the user area and data area can be erased. After all blocks in the user area and data area are erased in the erase ready state, the user area and data area can be read and programmed.

When the control code indicates that boot mode ID code protection is enabled, the MCU compares ID codes sent from the programmer with the control code and ID code 1 to ID code 15 in the user area. According to the comparison result, reading and programming the user area and data area are disabled.

#### (1) Control Code

The control code determines whether protection is enabled or disabled and the method of authentication with the programmer. Table 41.11 lists the protection specifications and Figure 41.25 shows the protection authentication flow.

**Table 41.11 Boot Mode ID Code Protection Specifications**

ID Code		Protection	ID Code Matching Result	Content of User Area and Data Area	Operation
Control Code	ID Code 1 to ID Code 15				
45h	Any desired value	Enabled	Matched	—	Exit the boot mode ID code authentication state and enter the program/erase state.
			Not matched	—	Continues the boot mode ID code authentication state.
			Not matched three times consecutively	Not blank	Enter the erase ready state for erasing the user area and data area.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., and FFh (8 bytes are all FFh)	Enabled	N/A	—	Determine that the codes do not match regardless of the codes sent from the serial programmer, and disable reading or rewriting of the flash memory.
	Other than above		Matched	—	Exit the boot mode ID code authentication state and enter the program/erase state.
	Other than above		Not matched	—	Continues the boot mode ID code authentication state.
Other than above	Any desired value	Disabled	N/A	Blank	Enter the program/erase state.
				Not blank	Enter the erase ready state for erasing the user area and data area.



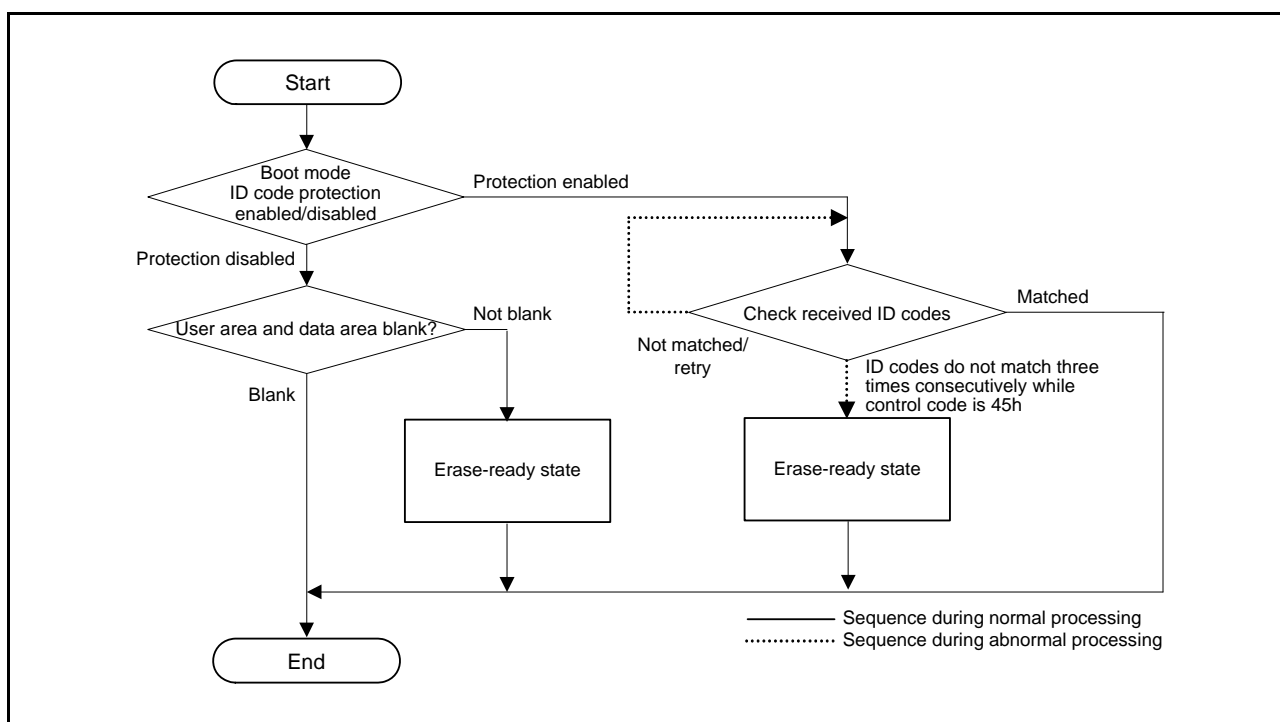


Figure 41.25 Authentication for Boot Mode ID Code Protection

(2) ID Code 1 to ID Code 15

ID code 1 to ID code 15 can be set to any desired value.

However, only when disabling connection with the serial programmer, the ID codes must be set to 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, FFh, FFh, FFh, FFh, FFh, FFh, and FFh (from the ID code 1 field to the ID code 15 field).

### 41.9.1.2 On-Chip Debugging Emulator ID Code Protection

On-chip debugging emulator ID code protection enables or disables connection with the on-chip debugging emulator. When the on-chip debugging emulator ID code protection is disabled, connection with the on-chip debugging emulator is enabled. When ID codes sent from the on-chip debugging emulator and ID codes in the user area match while on-chip debugging emulator ID code protection is enabled, connection with the on-chip debugging emulator is also enabled. The ID code configuration shown in Figure 41.24 is used for the on-chip debugging emulator ID code protection. Table 41.12 lists the protection specifications.

**Table 41.12 On-Chip Debugging Emulator ID Code Protection Specifications**

ID Code		Protection	ID Code Matching Result	Operation
Control Code	ID Code 1 to ID Code 15			
FFh	FFh, ..., and FFh (15 bytes are all FFh)	Disabled	N/A	Enable connection with the on-chip debugging emulator.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, and 74h + any 8 bytes	Enabled	N/A	Determine that the codes do not match regardless of the codes sent from the on-chip debugging emulator, and disable connection with the on-chip debugging emulator.
Other than above	Other than above	Enabled	Matched	Connection with the on-chip debugging emulator is enabled.
			Not matched	Continue the ID code wait state.

### 41.10 Communication Protocol

This section describes the protocol used in boot mode. When developing a programmer, control with this communication protocol.

#### 41.10.1 State Transition in Boot Mode (SCI)

Figure 41.26 shows the Boot Mode (SCI) State Transition. Descriptions for numbers in parenthesis are on the following page.

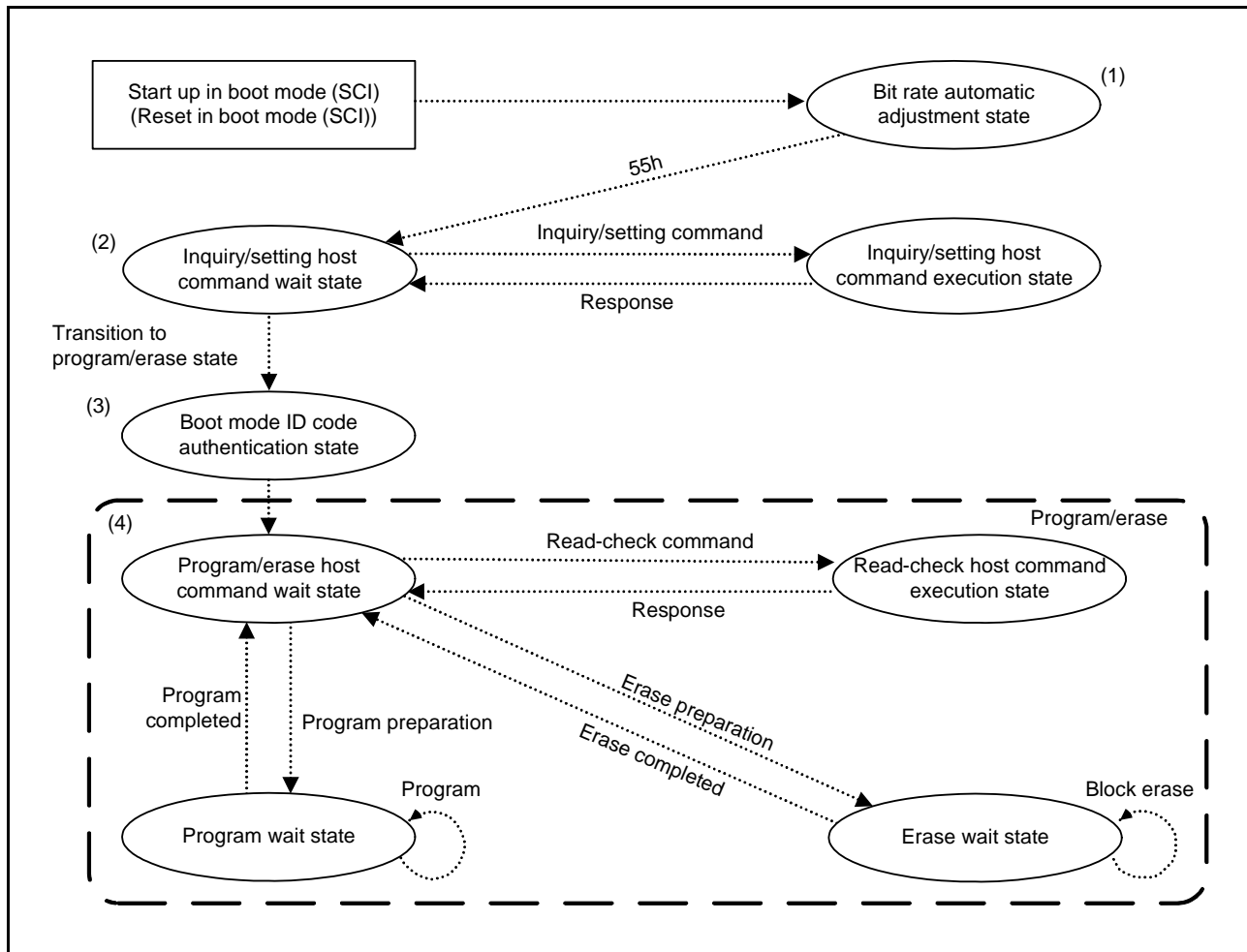


Figure 41.26 Boot Mode (SCI) State Transition

### (1) Bit rate automatic adjustment state

In this state, the bit rate is automatically adjusted for communication with the host.

When the bit rate adjustment is completed, the MCU sends 00h to the host. After that, when the MCU receives 55h sent from the host, the MCU sends E6h to the host, and the MCU enters the inquiry/setting host command wait state.

The host must not send data until 400 ms elapse after a reset of the MCU is released.

### (2) Inquiry/setting host command wait state

In this state, the host can make inquiries for the MCU information including area configuration, size, and addresses, and select a device and bit rate.

When the host sends a program/erase state transition command, the MCU enters the boot mode ID code authentication state.

Refer to section 41.10.4, Inquiry Commands and section 41.10.5, Setting Commands for details on inquiry/setting commands.

### (3) Boot mode ID code authentication state

In this state, the MCU compares ID codes in boot mode ID code protection.

When the user area and data area are blank while boot mode ID code protection is disabled, the MCU enters the program/erase state. When the user area and data area are not blank while boot mode ID code protection is disabled, the MCU enters the erase ready state to erase the user area and data area. When all blocks in the user area and data area are erased in the erase ready state, the MCU enters the program/erase state. When boot mode ID codes do not match, the MCU remains in the boot mode ID code authentication state.

Refer to section 41.9.1.1, Boot Mode ID Code Protection for details on boot mode ID code protection. Refer to section 41.10.6, ID Code Authentication Command for details on the ID code authentication command.

### (4) Program/erase state

In this state, the MCU executes program/erase or read-check commands according to commands sent from the host.

Refer to section 41.10.7, Program/Erase Commands for details on program/erase commands. Refer to section 41.10.8, Read-Check Commands for details on read-check commands.

## 41.10.2 Command and Response Configuration

The communication protocol is composed of a “Command” sent from the host to the MCU and a “Response” sent from the MCU to the host. Commands include 1-byte commands and multiple-byte commands. Responses include 1-byte responses, multiple-byte responses, and error responses.

A multiple-byte command and multiple-byte response have “Size” for informing the number of transmit/receive data bytes and “SUM” for detecting communication errors.

“Size” indicates the number of transmit/receive data bytes excluding Command (the first byte), Size, SUM.

“SUM” indicates byte data that is calculated so the total bytes of Command or Response becomes 00h.

The flash memory addresses for reading are used as the program address for the program command, the block start address for the block erase command, the AW start address for the access window information program command and access window read command, and the AW end address.

For details on the flash memory addresses for reading, see Figure 41.1 and Figure 41.2.

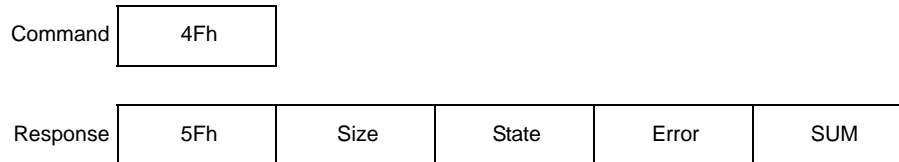
When the host sends an undefined command, the MCU sends a command error as a response. The contents of the response is shown below. The command in the error response stores the first byte of the command sent from the MCU.

Error response	80h	Command
----------------	-----	---------

### 41.10.3 Boot Mode Status Inquiry

This command is used to check the current state and the previous error of the boot program. The MCU returns a code from Table 41.13 and Table 41.14 as the current state and the previous error.

The boot mode status inquiry command can be used in the inquiry/setting host command wait state and program/erase state.



Size (1 byte): Total bytes of "State" and "Error" (the value is always 2)

State (1 byte): MCU state (see Table 41.13)

Error (1 byte): Information about the error occurred in the MCU (see Table 41.14)

SUM (1 byte): Value that is calculated so the sum of response data is 00h

**Table 41.13 Information Regarding the States**

Code	State*1	Description
11h	Inquiry/setting host command wait state	Device selection wait state
12h/ 13h		Clock mode selection wait state
1Fh		Program/erase state transition command wait state
3Fh	Program/erase state	Program/erase host command wait state
4Fh		Program wait state
5Fh		Erase wait state

Note 1. Refer to Figure 41.26 for details on the states.

**Table 41.14 Error Information**

Code	Description
00h	No error
11h	SUM error
21h	Device code error
24h	Bit rate selection error
29h	Block address error
2Ah	Address error
2Bh	Data length error
51h	Erase error
52h	Not blank
53h	Program error
80h	Command error
FFh	Bit rate automatic adjustment error

## 41.10.4 Inquiry Commands

Inquiry commands are used to obtain necessary information for sending setting commands, program/erase commands, and read-check commands. Table 41.15 lists the inquiry commands. These commands can only be used in the inquiry/setting host command wait state.

**Table 41.15 Inquiry Commands**

Command	Description
Supported device inquiry	Inquiry for the device code and series name
Data area availability inquiry	Inquiry for the availability of the data area
User area information inquiry	Inquiry for the number of user areas, and the start and end addresses of the user area
Data area information inquiry	Inquiry for the number of data areas, and the start and end addresses of the data area
Block information inquiry	Inquiry for the start and end addresses of the user areas and data areas, the block size, and the number of blocks

### 41.10.4.1 Supported Device Inquiry

When the host sends this command, the MCU sends information about a device for little endian data and a device for big endian data in this order.

Command	20h		
Response	30h	Size	Number of devices
	Number of characters	Device code for little endian	
	Number of characters	Device code for big endian	
	SUM		
	Series name for little endian		
	Series name for big endian		

Size (1 byte): Total bytes of Number of Devices, Characters, Device code, and Series name  
 Number of devices (1 byte): Number of endian types of program data (the value is always 02h)  
 Number of characters (1 byte): Number of characters for the device code and device name  
 Device code (4 bytes): Identification code indicating the endian of program data  
 Series name (n bytes): ASCII code of the series name of the supported device  
 SUM (1 byte): Value that is calculated so the sum of response data is 00h

### 41.10.4.2 Data Area Availability Inquiry

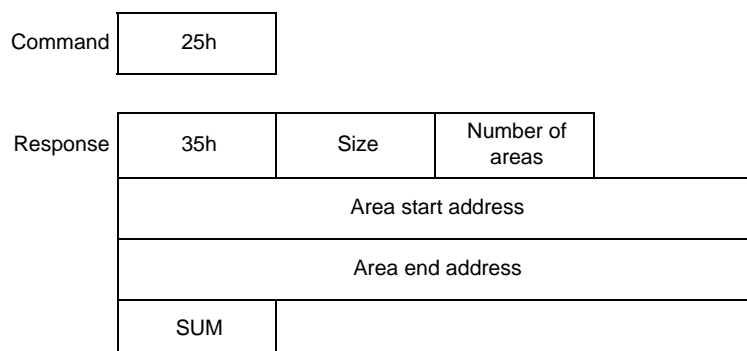
When the host sends this command, the MCU sends data indicating that the data area is available and area protection can be used.

Command	2Ah			
Response	3Ah	Size	Availability	SUM

Size (1 byte): Number of characters of Availability (the value is always 01h)  
 Availability (1 byte): Availability of the data area (the value is always 19h)  
 19h represents the data area is available and area protection can be used.  
 SUM (1 byte): Value that is calculated so the sum of response data is 00h (the value is always ACh)

### 41.10.4.3 User Area Information Inquiry

When the host sends this command, the MCU sends the number of user areas and addresses.



Size (1 byte): Total bytes of Number of areas, Area start address, and Area end address (the value is always 09h)

Number of areas (1 byte): Number of user areas (the value is always 01h)

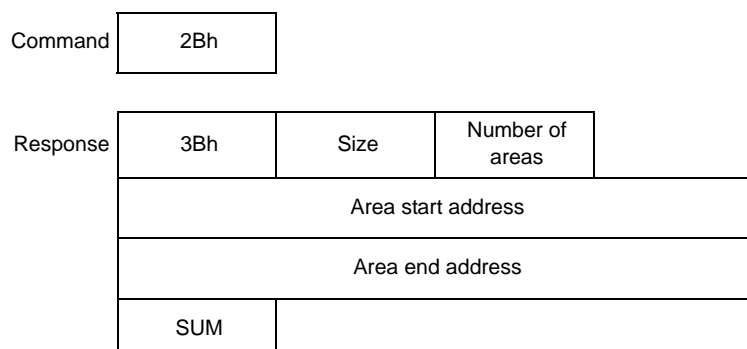
Area start address (4 bytes): Start address of the user area

Area end address (4 bytes): End address of the user area

SUM (1 byte): Value that is calculated so the sum of the response data is 00h

### 41.10.4.4 Data Area Information Inquiry

When the host sends this command, the MCU sends the number of data areas and addresses.



Size (1 byte): Total bytes of data of Number of areas, Area start address, and Area end address (the value is always 09h)

Number of areas (1 byte): Number of areas in the data area (the value is always 01h)

Area start address (4 bytes): Start address of the data area (the value is always 0010 0000h)

Area end address (4 bytes): End address of the data area (the value is always 0010 1FFFh)

SUM (1 byte): Value that is calculated so the sum of the response data is 00h (the value is always 7Dh)

### 41.10.4.5 Block Information Inquiry

When the host sends this command, the MCU sends the start address, the size of one block, and the number of blocks in the user area and data area.

Command	26h		
Response	36h	Size	DDh
	Start address of the user area		
	Block size of one block for the user area		
	Number of blocks of the user area		
	Start address of the data area		
	Block size of one block for the data area		
	Number of blocks of the data area		
	SUM		

Size (2 bytes): Total bytes of data from DDh to Number of blocks of the data area (the value is always 00 19h)

Start address of the user area (4 bytes): Start address of the user area

Block size of one block for the user area (4 bytes): Memory size of one block (the value is always 00 00 04 00h)

Number of blocks of the user area (4 bytes): Number of blocks in the user area

Start address of the data area (4 bytes): Start address of the data area (the value is always 00 10 00 00h)

Block size of one block for the data area (4 bytes): Memory size of one block (the value is always 00 00 04 00h)

Number of blocks of the data area (4 bytes): Number of blocks in the data area (the value is always 00 00 04 00h)

SUM (1 byte): Value that is calculated so the sum of response data is 00h



## 41.10.5 Setting Commands

Setting commands are used to configure the settings necessary to execute program/erase commands in the MCU.

Table 41.16 lists Setting Commands. These commands can be used only in the inquiry/setting host command wait state.

**Table 41.16 Setting Commands**

Command	Function
Device select	Select a device code.
Operating frequency select	Change the bit rate for communication.
Program/erase state transition	Enter the erase ready state.

### 41.10.5.1 Device Select

To send a device code from the host, select a device code in the response to the support device inquiry command, and send it using the device select command. Select the device code corresponding to the endian of program data.

When the device is supported, the MCU sends a response. When the device is not supported or the received command is invalid, the MCU sends an error response.

Command	10h	Size	Device code	SUM
---------	-----	------	-------------	-----

Size (1 byte): Number of characters of the device code (the value is always 04h)

Device code (4 bytes): Identification code indicating the device  
(code in the response to the support device inquiry command)

SUM (1 byte): Value that is calculated so the sum of command data is 00h

Response	46h
----------	-----

Error response	90h	Error
----------------	-----	-------

Error (1 byte): Error code

11h: SUM error

21h: Device code error

### 41.10.5.2 Operating Frequency Select

When the host sends a command to set the bit rate, select 16 MHz input clock and a bit rate with error of less than 4%. When the settings are supported, the MCU sends a response. When the settings are not supported or the transmitted command is invalid, the MCU sends an error response.

After a response is received, wait for a 1-bit period and then change the bit rate to the selected value from the host. When the MCU successfully receives communication confirmation data, the MCU sends a response. When the MCU fails to receive the communication confirmation data, the MCU sends an error response.

Command	3Fh	Size	Bit rate		Input frequency
	Number of Clocks	Multiplier 1	Multiplier 2		
	SUM				

Size (1 byte): Total bytes of data of Bit rate, Input frequency, Number of clocks, and Multiplier (the value is always 07h)

Bit rate (2 bytes): New bit rate (e.g. 00C0h: 19,200 bps)

The value is calculated by dividing the bit rate by 100.

Input frequency (2 bytes): Input frequency of the MCU (the value is always 0640h: 16 MHz)

Number of clocks (1 byte): Types of clocks (the value is always 02h)

Multiplier 1 (1 byte): Multiplier of the input frequency for the system clock (ICLK) (the value is always 01h)

Multiplier 2 (1 byte): Multiplier of the input frequency for the peripheral module clock (PCLK) (the value is always 01h)

SUM (1 byte): Value that is calculated so the sum of command data is 00h

Response	06h
----------	-----

Error response	BFh	Error
----------------	-----	-------

Error (1 byte): Error code

11h: SUM error

24h: Bit rate selection error

Communication Confirmation	06h
----------------------------	-----

Response	06h
----------	-----

Error response	FFh
----------------	-----

- Bit rate selection error

A bit rate selection error occurs when the bit rate specified with the operating frequency select command cannot be set to a value with error of less than 4%. When the bit rate selected by the operating frequency select command is B, and the value of the SCI bit rate register (BRR) is N, the bit rate error is calculated by the following formula:

$$\text{Error (\%)} = \left\{ \left[ \frac{16 \times 10^6}{B \times 64 \times 2^{-1} \times (N + 1)} \right] - 1 \right\} \times 100$$

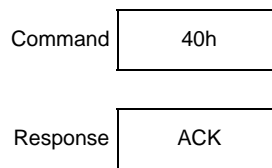
### 41.10.5.3 Program/Erase State Transition

When the host sends this command, the MCU performs processing for the boot mode ID code protection.

When boot mode ID code protection is disabled while the user area and data area are blank, the MCU sends a response (06h) indicating that ID code protection is disabled and the MCU enters the program/erase state.

When boot mode ID code protection is disabled while the user area and data area are not blank, the MCU sends a response (56h) indicating that ID code protection is disabled and the MCU enters the erase ready state.

When boot mode ID code protection is enabled, the MCU sends a response (16h) indicating that ID code protection is enabled.



ACK (1 byte): ACK code

06h: ID code protection is disabled. The user area and data area are blank.\*1

56h: ID code protection is disabled.

16h: ID code protection is enabled.

Note 1. Erase the block that is programmed before sending a program command.

### 41.10.6 ID Code Authentication Command

The ID code authentication command is used to send data from the host to compare with the control code and ID code 1 to ID code 15 on the ROM when the boot mode ID code protection is enabled.

Table 41.17 lists ID code authentication command. This command can be used only in the boot mode ID code authentication state.

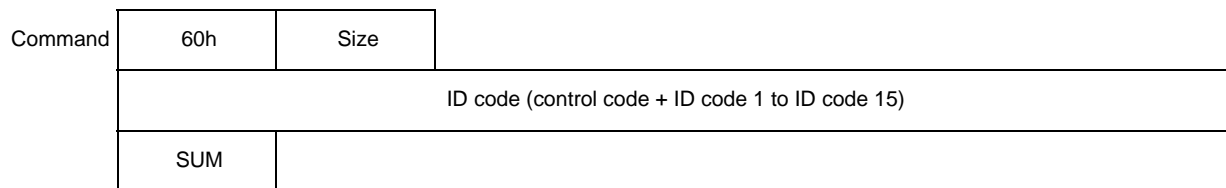
**Table 41.17 ID Code Authentication Command**

Command	Function
ID code check	Compare data with the control code and ID code 1 to ID code 15 using boot mode ID code protection.

#### 41.10.6.1 ID Code Check

Send the same data as the control code and ID code 1 to ID code 15 in the user area from the host. When the control code and ID code 1 to ID code 15 match the boot mode ID codes, the MCU enters the program/erase state and sends a response (06h).

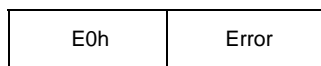
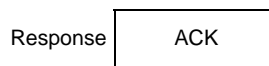
When ID codes do not match three times consecutively while the control code is 45h, the MCU enters the erase ready state and sends a response (56h). When they do not match or when the MCU fails to receive data, the MCU sends an error response.



Size (1 byte): Number of bytes of ID codes (the value is always 16)

ID code (16 bytes): Control code (1 byte) + ID code 1 to ID code 15 (15 bytes)

SUM (1 byte): Value that is calculated so the sum of the command data is 00h



ACK (1 byte): ACK code

06h: The MCU enters the program/erase state.

56h: The MCU enters the erase ready state.

Error (1 byte): Error code

11h: SUM error

61h: ID codes do not match

### 41.10.6.2 Erase Ready

The erase ready is a part of the boot mode ID code protection to disable reading data stored in the user area and data area when protection is disabled while the user area and data area are not blank or when ID codes do not match three times while the control code is 45h.

Only the erase preparation command and block erase command can be accepted in the erase ready state.

Table 41.18 lists ID Code Authentication Command.

**Table 41.18 Commands Used for Erase Ready**

Command	Function
Erase preparation	Enter the erase wait state.
Block erase	Erase the selected block, or enter the program/erase state (end of erase).

Note: Refer to section 41.10.8, Read-Check Commands for details on the erase preparation command and block erase command.

### 41.10.7 Program/Eraser Commands

Program/erase commands are used to program or erase the user area or data area based on the response to inquiry commands. Table 41.19 lists commands used in the program/erase command wait state, program wait state, and erase wait state. Table 41.20 lists commands that cannot be accepted in each state.

When a command that cannot be accepted is received in the state listed in Table 41.20, the MCU sends a command error response.

**Table 41.19 Program/Eraser Commands**

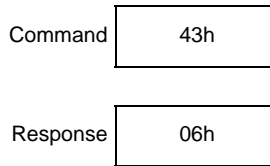
Command	Function
User/data area program preparation	Select the user area or data area to program, and enter the program wait state.
Program	Program the selected area, or enter the program erase state (end of program).
Erase preparation	Enter the erase wait state.
Block erase	Erase the selected block, or enter the program erase state (end of erase).

**Table 41.20 Commands That Cannot Be Accepted**

State	Command That Cannot Be Accepted
Program/erase host command wait state	Commands other than the program command and block erase command
Program wait state	Commands other than the program command
Erase wait state	Commands other than the block erase command

### 41.10.7.1 User/Data Area Program Preparation

When the host sends this command, the MCU recognizes that an instruction to prepare for the program command is issued from the host, enters the program wait state, where only the program command to the user area or data area can be accepted, and sends a response.

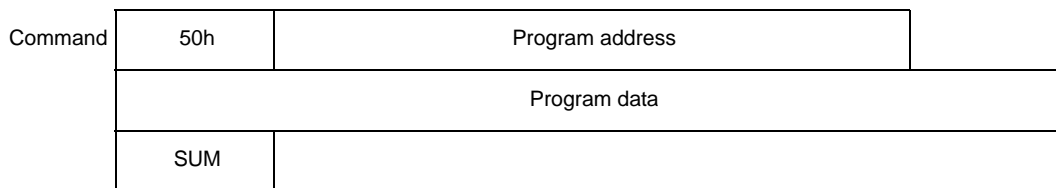


### 41.10.7.2 Program

Set the program address sent from the host aligned on a 256-byte boundary. When the data length is shorter than 256 bytes, the data cannot be programmed. Fill the gaps with FFh to send.

When the program from the selected address is successfully completed, the MCU sends a response. When an error occurs during a program operation, the MCU sends an error response.

When entering program/erase host command wait state, send 50h FFh FFh FFh FFh B4h from the host. The MCU enters the program/erase host command wait state and sends a response.



Program address (4 bytes): Address for program destination

Address aligned on the program data length

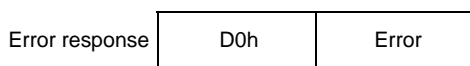
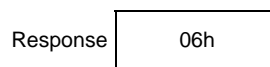
Set FFFF FFFFh for end of program

Program data (n bytes): Program data (n = 256 in boot mode, 0 for end of program)

When the program is less than 256 bytes, set FFh for the missing data.

No data for the end of program

SUM (1 byte): Value that is calculated so the sum of command data is 00h.



Error (1 byte): Error code

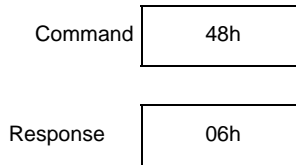
11h: SUM error

2Ah: Address error (the address is not in the selected area.)

53h: Program error (the program cannot be written.)

### 41.10.7.3 Erase Preparation

When the host sends this command, the MCU recognizes that an instruction to prepare for the erase command is issued from the host, enters the erase wait state, where only the block erase command to the user area or data area can be accepted, and sends a response.

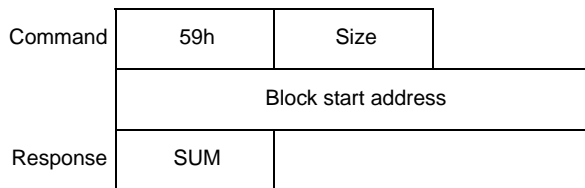


### 41.10.7.4 Block Erase

Send block addresses from the host based on the response to the block information inquiry command.

When the block selected in addresses is successfully erased, the MCU sends a response. When an error occurs during an erase operation, the MCU sends an error response.

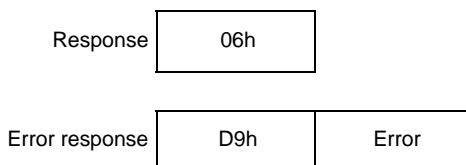
When the MCU enters the program/erase host command wait state after the erase operation ends, send 59h 04h FFh FFh FFh FFh A7h from the host. The MCU enters the program/erase host command wait state and sends a response.



Size (1 byte): Total bytes of Block start address (the value is always 04h)

Block start address (4 bytes): Start address of the block that is erased  
Set FFFF FFFFh for end of erase

SUM (1 byte): Value that is calculated so the sum of response data is 00h



Error (1 byte): Error code

11h: SUM error

29h: Block address error (the block address is not correct)

51h: Erase error (the selected block cannot be erased)

### 41.10.8 Read-Check Commands

Read-check commands are used to read or check the user area or data area in the MCU based on the response to inquiry commands.

Table 41.21 lists read-check commands used in the program/erase command wait state.

**Table 41.21 Read-Check Commands**

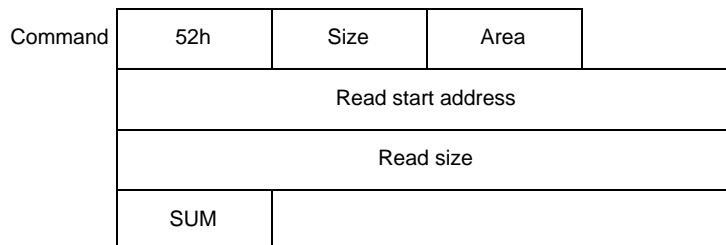
Command	Function
Memory read	Read data from the user area. Read data from the data area.
User area checksum	Checksum of the user area.
Data area checksum	Checksum of the data area.
User area blank check	Check whether data is programmed in the user area.
Data area blank check	Check whether data is programmed in the data area.
Access window information program	Set the access window.
Access window read	Read the settings of the access window.

#### 41.10.8.1 Memory Read

For a read start address sent from the host, set an address within the range from the start address to the end address received in the response to the user area information inquiry command or the data area information inquiry command.

For a read size sent from the host, set a value so the sum of the read start address and the read size is within the range from the start address to the end address received in the response to the user area information inquiry command or the data area information inquiry command.

When the MCU performs a read successfully, the MCU sends data for the read size from the read start address. When the MCU fails to read the flash memory, the MCU sends an error response.



Size (1 byte): Total bytes for Read start address and Read size

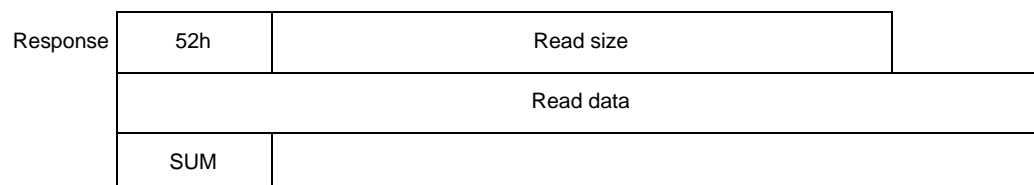
Area (1 byte): Area that is read

01h: User area or data area

Read start address (4 bytes): Start address of the area that is read

Read size (4 bytes): Size of data that is read (in bytes)

SUM (1 byte): Value that is calculated so the sum of response data is 00h

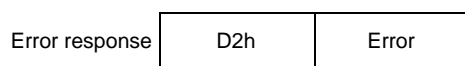


Read size (4 bytes): Size of Data that is read (in bytes)

Read data (read size): Data read from the selected address

SUM (1 byte): Value that is calculated so the sum of response data is 00h





Error (1 byte): Error codes

11h: SUM error

2Ah: Address error

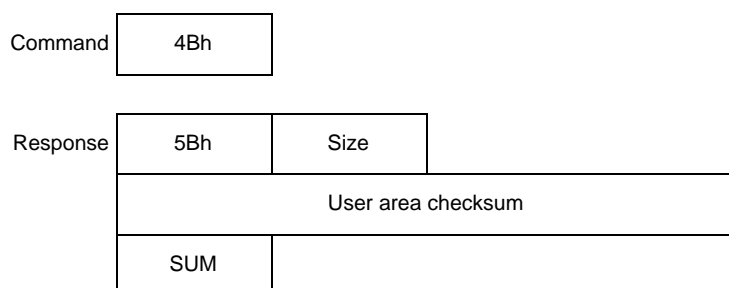
- A value other than 00h and 01h is set for area.
- The read start address is not in the selected area.

2Bh: Size error

- The read size is set to 00h.
- The read size exceeds the area size
- The address calculated from the read start address and read size is not in the selected area

### 41.10.8.2 User Area Checksum

When the host sends this command, the MCU adds data from the start address and the end address in bytes in the response to the user area information inquiry command, and sends the calculated result (checksum) as a response.



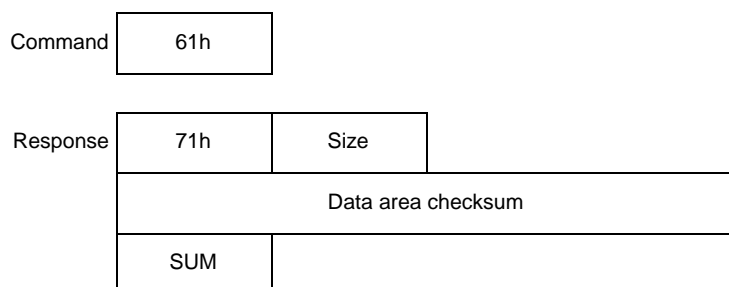
Size (1 byte): Number of bytes for checksum of the user area (the value is always 04h)

User area checksum (4 bytes): Checksum result of the user area

SUM (1 byte): Value that is calculated so the sum of response data is 00h

### 41.10.8.3 Data Area Checksum

When the host sends this command, the MCU adds data from the start address and the end address in bytes in the response to the data area information inquiry command, and sends the calculated result (checksum) as a response.



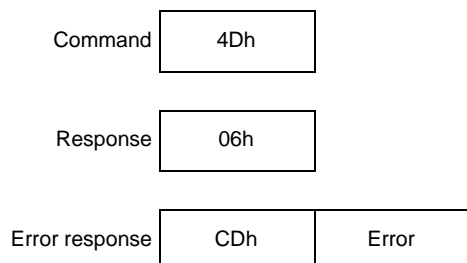
Size (1 byte): Number of bytes for checksum of the data area (the value is always 04h)

Data area checksum (4 bytes): Checksum result of the data area

SUM (1 byte): Value that is calculated so the sum of response data is 00h

#### 41.10.8.4 User Area Blank Check

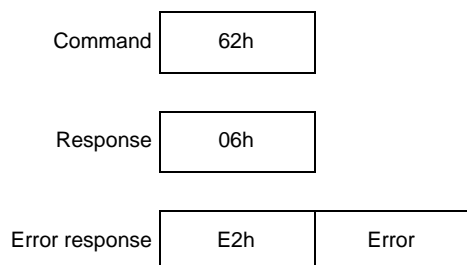
When the host sends this command, the MCU sends a response when there is no data in the area from the start address to the end address received in the response to the user area information inquiry command. When there is at least 1 byte of data, the MCU sends an error response.



Error (1 byte): Error code  
52h: Not blank

#### 41.10.8.5 Data Area Blank Check

When the host sends this command, the MCU sends a response when there is no data in the area from the start address to the end address received in the response to the data area information inquiry command. When there is at least 1 byte of data, the MCU sends an error response.



Error (1 byte): Error code  
52h: Not blank

### 41.10.8.6 Access Window Information Program

For the access window start address sent from the host, set the block start address of the user area. For the access window end address, set the block end address of the user area.

When the program of selected address is successfully completed, the MCU sends a response. When an error occurs during a program operation, the MCU sends an error response.

The access window settings can be overwritten because it is a part of protection.

Command	74h	05h	Access window	
	Access window start address LH	Access window start address HL	Access window end address LH	Access window end address HL
	SUM			

Access window (1 byte): Select the access window or clear the access window settings  
Set 00h to select the access window  
Set FFh to clear the access window settings

Access window start address LH (1 byte): Start address of the access window (D15 to D8)  
Set D15 to D8 of the block start address.  
Set FFh to clear the access window settings

Access window start address HL (1 byte): Start address of the access window (D23 to D16)  
Set D23 to D16 of the block start address.  
Set FFh to clear the access window settings

Access window end address LH (1 byte): End address of the access window (D15 to D8)  
Set D15 to D8 of the block end address.  
Set FFh to clear the access window settings

Access window end address HL (1 byte): End address of the access window (D23 to D16)  
Set D23 to D16 of the block end address.  
Set FFh to clear the access window settings

SUM (1 byte): Value that is calculated so the sum of response data is 00h

Response	06h
----------	-----

Error response	F4h	Error
----------------	-----	-------

Error (1 byte): Error code  
11h: SUM error  
2Ah: Address error (address is not in the selected area)  
53h: Program error (program cannot be written)

### 41.10.8.7 Access Window Read

Send command 73h 01h FFh 8Dh from the host.

When the MCU successfully reads the access window settings, the MCU sends the access window start address and end address that the MCU read.

When the MCU fails to read the access window settings, the MCU sends an error response.

Command	73h	01h	FFh	8Dh
---------	-----	-----	-----	-----

Response	73h	05h		
	Access window start address LH	Access window start address HL	Access window end address LH	Access window end address HL
	FFh			
	SUM			

- Access window start address LH (1 byte): Start address of the access window (D15 to D8)
- Access window start address HL (1 byte): Start address of the access window (D23 to D16)
- Access window end address LH (1 byte): End address of the access window (D15 to D8)
- Access window end address HL (1 byte): End address of the access window (D23 to D16)
- SUM (1 byte): Value that is calculated so the sum of response data is 00h

Error response	F3h	Error
----------------	-----	-------

Error (1 byte): Error code  
 11h: SUM error

### 41.10.9 Programmer Operation in Boot Mode (SCI)

The following describes the procedure for the programmer to access the user area and data area in boot mode (SCI).

1. Automatically adjust the bit rate
2. Receive the MCU information\*<sup>1</sup>
3. Select the device and change the bit rate
4. Enter the program/erase state
5. Disable boot mode ID code protection
6. Perform erase ready processing
7. Erase the user area and data area\*<sup>2</sup>
8. Program the user area and data area\*<sup>2</sup>
9. Check data in the user area\*<sup>2</sup>
10. Check data in the data area\*<sup>2</sup>
11. Set the access window in the user area\*<sup>3</sup>
12. Reset the MCU

Note 1. If the necessary information has been already received, step 2 can be skipped.

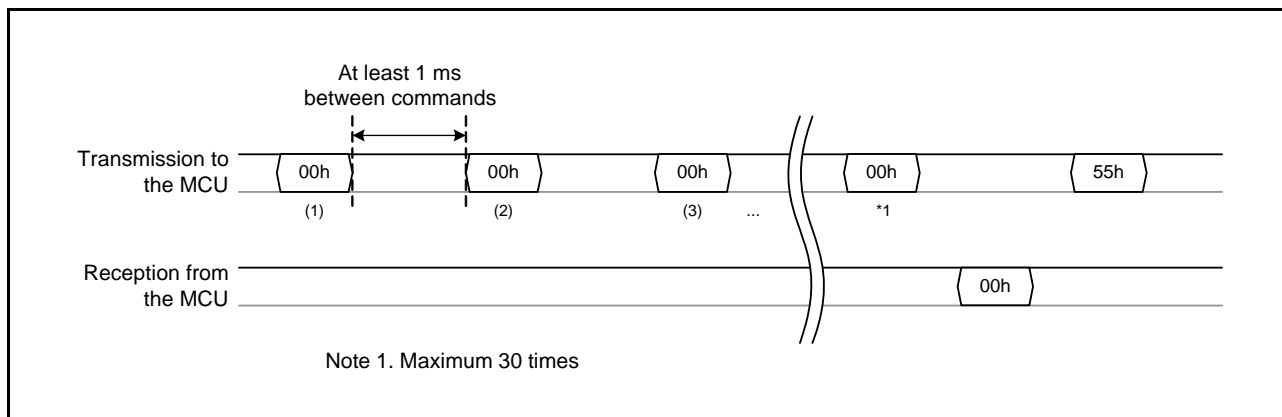
Note 2. Any step from 7 to 11 can be skipped, and their order can be changed.

Note 3. When a timeout occurs or invalid response data is received, stop the operation and perform step 12 (reset the MCU).

Refer to section 41.10.9.1, Bit Rate Automatic Adjustment Procedure to section 41.10.9.11, Set the Access Window in the User Area for details on the procedure above. Refer to section 41.10.4, Inquiry Commands, section 41.10.5, Setting Commands, section 41.10.6, ID Code Authentication Command, section 41.10.7, Program/Erase Commands, and section 41.10.8, Read-Check Commands for details on commands.

### 41.10.9.1 Bit Rate Automatic Adjustment Procedure

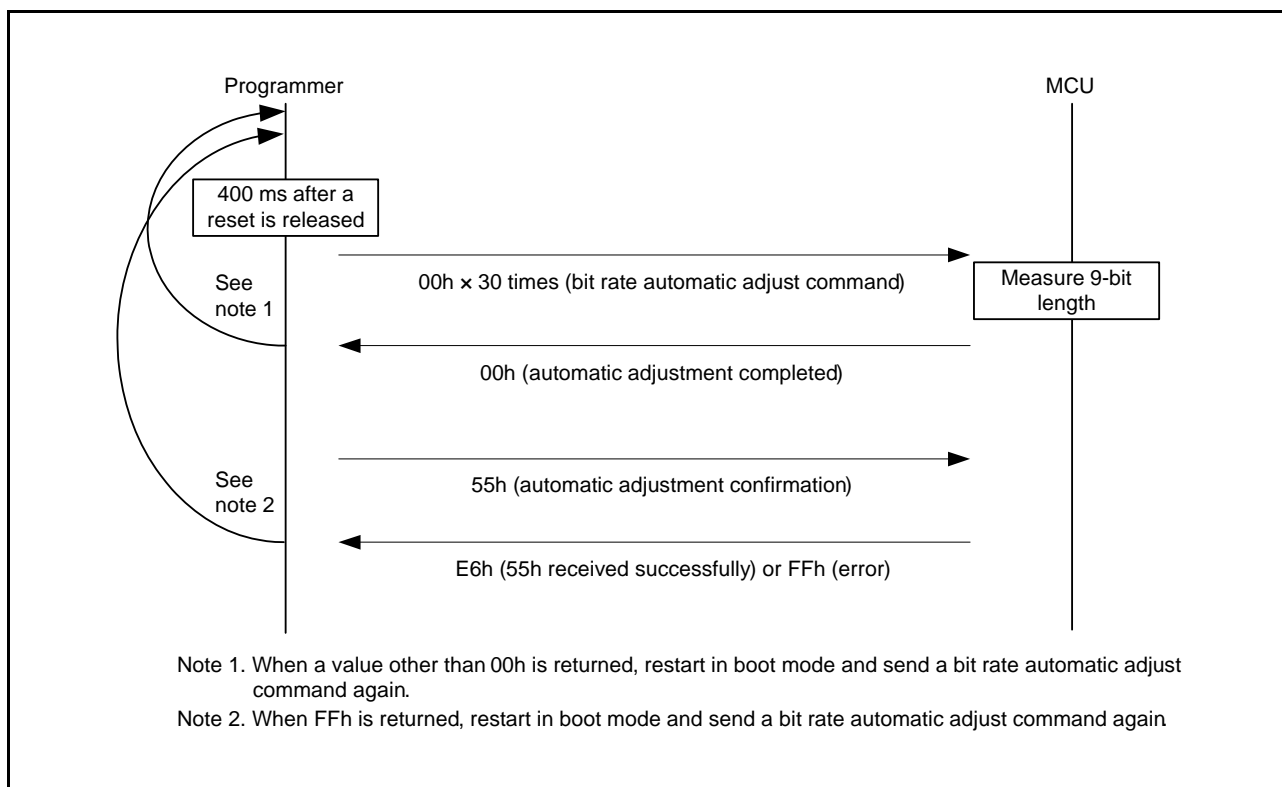
The MCU measures the low width of data 00h to adjust the bit rate.



**Figure 41.27 Transmit/Receive Data for Bit Rate Automatic Adjustment**

After starting up in boot mode, wait for 400 ms and then send 00h to the MCU from the programmer. The programmer can send 00h to the MCU up to 30 times. When the bit rate adjustment is completed, the MCU sends 00h to the programmer. When the programmer receives 00h, send 55h to the MCU from the programmer. When the programmer fails to receive 00h, restart the MCU in boot mode, and adjust the bit rate again.

When the MCU receives 55h, the MCU sends E6h and enters the inquiry/setting command wait state. When the MCU fails to receive 55h, the MCU sends FFh. When the programmer receives FFh, restart the MCU in boot mode, and adjust the bit rate again.



**Figure 41.28 Bit Rate Automatic Adjustment Procedure**

### 41.10.9.2 Procedure to Receive the MCU Information

Send inquiry commands, and receive the information necessary to send setting commands, program/erase commands, and read-check commands.

- (1) Send a support device inquiry command (20h) to check which device to connect. The MCU returns the device code and series name.
- (2) Send a data area availability inquiry command (2Ah) to check the availability of data area and area protection. The MCU returns the availability of data area and area protection.
- (3) Send a user area information inquiry command (25h) to check the start and end addresses of the user area. The MCU returns the start and end addresses of the user area.
- (4) Send a block information inquiry command (26h) to check the block configuration. The MCU returns the start address, the size of one block, and the number of blocks for the user area and data area.
- (5) Send a data area information inquiry command (2Bh) to check the start and end addresses in the data area. The MCU returns the start and end addresses of the data area.

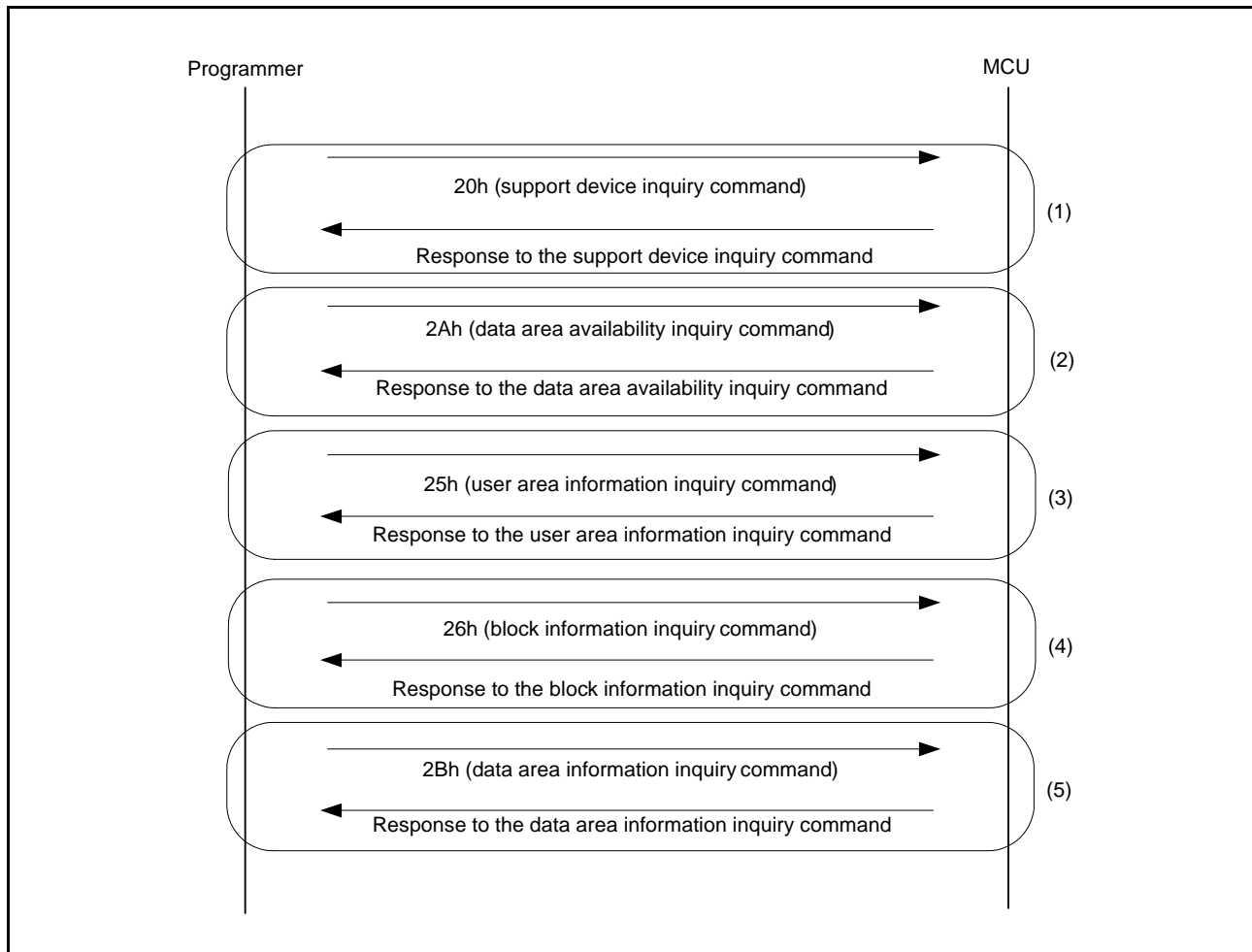


Figure 41.29 Procedure to Send Inquiry Commands

### 41.10.9.3 Procedure to Select the Device and Change the Bit Rate

Set the device to connect with the programmer and change the bit rate for communication.

- (1) Send the device select command (10h) to select the device to connect with the programmer and the endian of data that is programmed. When the program data is little endian, select the same device code as that for little endian in the response to the support device inquiry command. When the program data is big endian, select the same device code as that for big endian in the response to the support device inquiry command. When the device is selected successfully, the MCU sends a response (46h). When the MCU fails to receive, the MCU sends an error response (90h).
- (2) Send the operating frequency select command (3Fh) to change the bit rate for communication. When the bit rate is set successfully, the MCU sends a response (06h). When the bit rate cannot be changed, or when the MCU fails to receive, the MCU sends an error response (BFh).
- (3) When the programmer receives a response (06h), the MCU waits for 1-bit period at the bit rate for sending the operating frequency select command, and then set the bit rate of the programmer to the changed value. After that, the MCU sends communication confirmation data (06h) at the changed bit rate. When the MCU receives the command successfully, the programmer sends a response (06h) of the communication confirmation data.

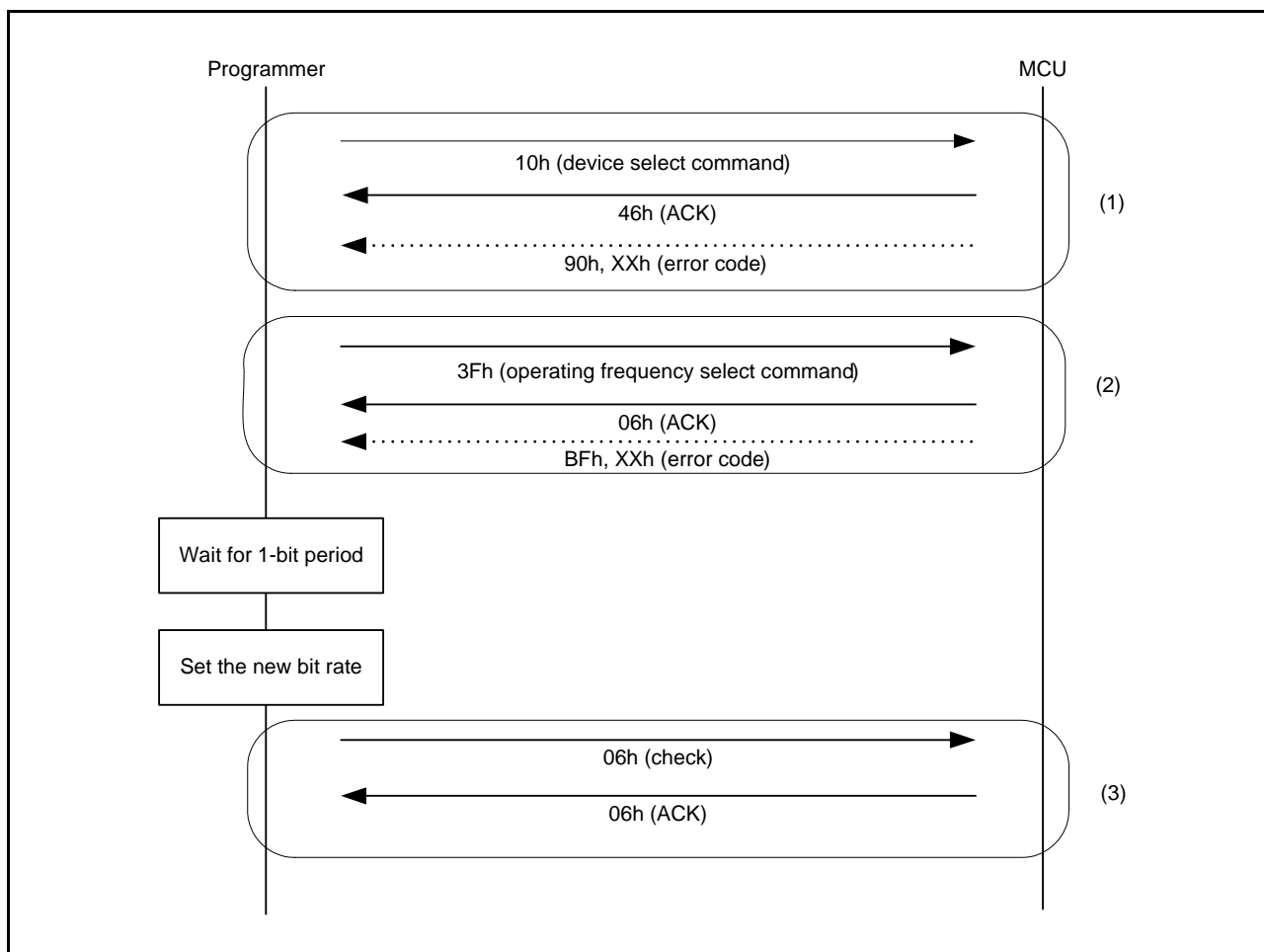


Figure 41.30 Procedure to Select the Device and Change the Bit Rate



#### 41.10.9.4 Transition to the Program/Erase State

The MCU needs to enter the program/erase state to perform program/erase operations.

Send the program/erase state transition command (40h). The MCU responds according to ID codes and the state of the user area and data area.

- (1) When boot mode ID code protection is disabled while the user area and data area are blank, the MCU sends a response (06h). After the MCU responds, the MCU is in the program/erase state and performs operations described in section 41.10.9.7, Erase the User Area and Data Area to section 41.10.9.11, Set the Access Window in the User Area. Perform an erase operation described in section 41.10.9.7, Erase the User Area and Data Area before programming.
- (2) When the boot mode ID code protection is disabled while the user area and data area are not blank, the MCU sends a response (56h). After the MCU responds, the MCU is in the erase ready wait state and performs operation described in section 41.10.9.6, Erase Ready Operation.
- (3) When the boot mode ID code protection is enabled, the MCU sends a response (16h). After the MCU responds, the MCU is in the ID code authentication wait state and performs operation described in section 41.10.9.5, Disable Boot Mode ID Code Protection.

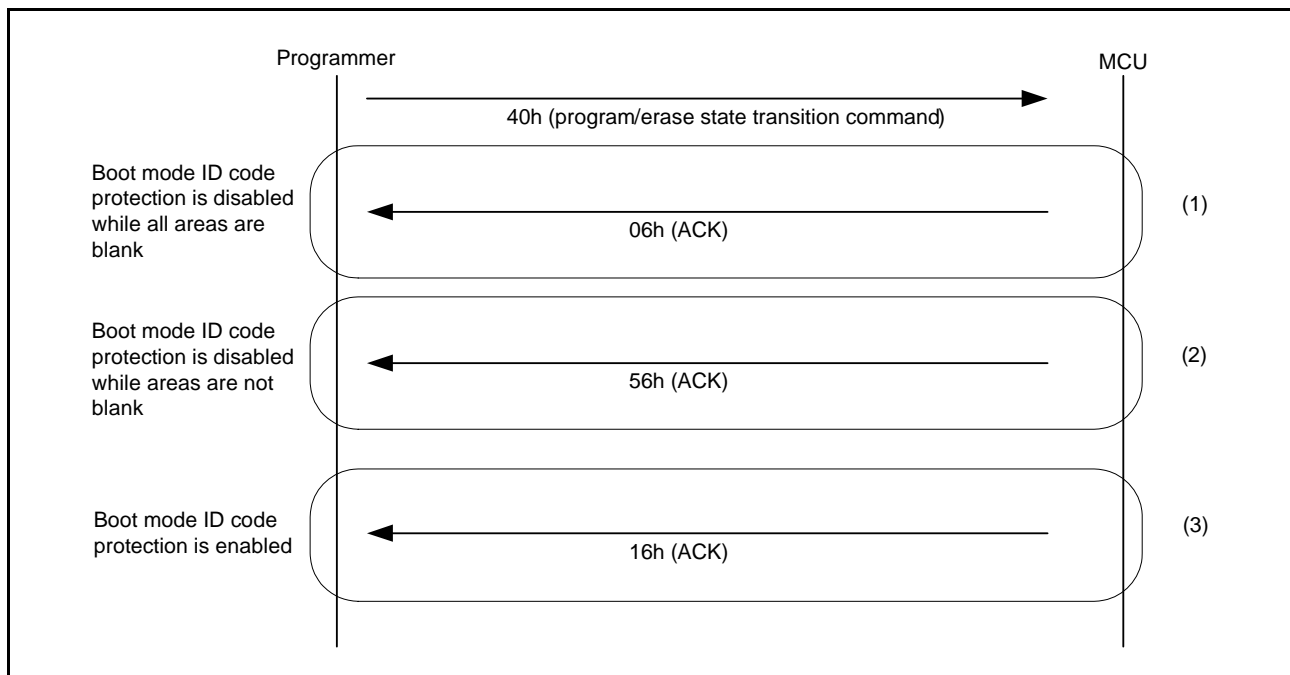


Figure 41.31 Procedure to Enter the Program/Erase State

### 41.10.9.5 Disable Boot Mode ID Code Protection

After the MCU is connected with the programmer, the boot mode ID code protection is enabled so program/read, and read-check operations cannot be performed. Disable this boot mode ID code protection.

Send the ID code check command (60h). When the MCU compares the received ID codes with ID codes in the user area and responds according to the comparison result.

- (1) When ID codes match, the MCU sends a response (06h). After the MCU responds, the MCU is in the program/erase state and performs operations described in section 41.10.9.7, Erase the User Area and Data Area to section 41.10.9.11, Set the Access Window in the User Area. Data in the user area and data area are not erased. Perform an erase operation described in section 41.10.9.7, Erase the User Area and Data Area before programming.  
When ID codes do not match, the MCU sends an error response (E0h). After the MCU responds, the MCU remains in the ID code authentication wait state.
- (2) If ID codes do not match three times consecutively while the control code is 45h, the MCU sends a response (56h). After the MCU responds, the MCU is in the erase ready wait state and performs operation described in section 41.10.9.6, Erase Ready Operation.

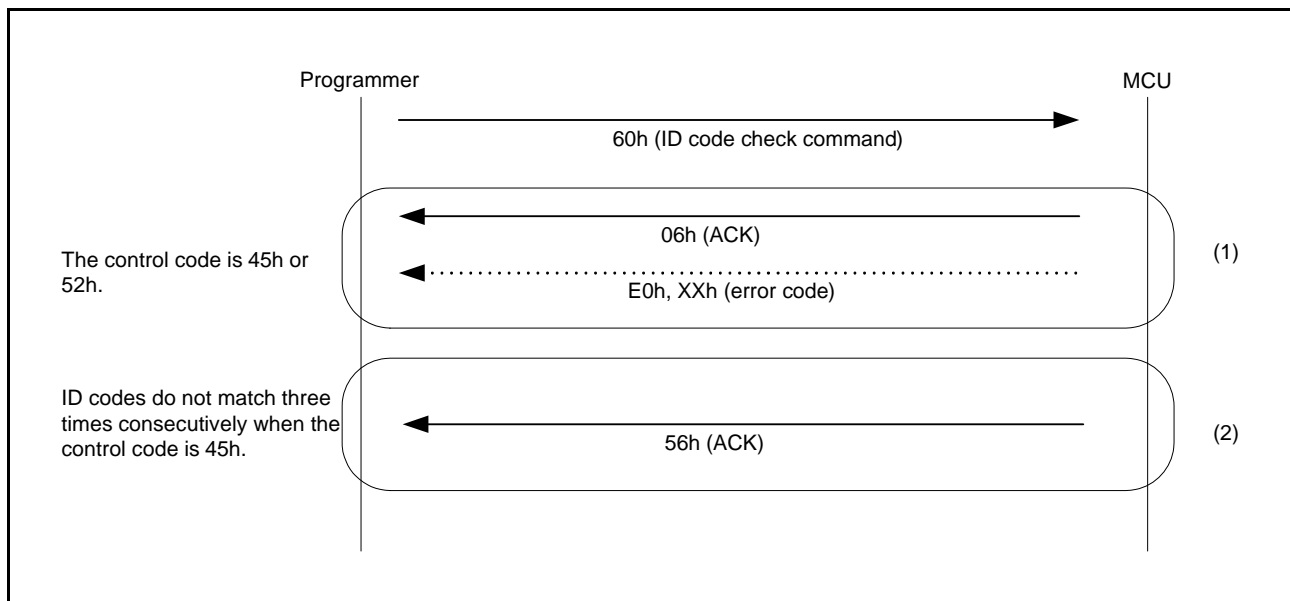


Figure 41.32 Procedure to Check ID Codes

### 41.10.9.6 Erase Ready Operation

Erase the user area and data area in the MCU.

- (1) Send the erase preparation command (48h) to place the MCU in the erase wait state. The MCU enters the erase wait state and sends a response (06h).
- (2) Send a block erase command (59h) to erase blocks in the MCU. When blocks are erased successfully, the MCU sends a response (06h). When the MCU fails to receive, the MCU sends an error response (D9h). Send block erase commands repeatedly until a block erase command has been sent for the total number of blocks. The total number of blocks is the sum of the user area blocks and data area blocks that are obtained in advance using the block information inquiry command. If the operation ends before all the block erase commands are sent, a command error may occur even when a correct command is sent in the program/erase state.
- (3) Send a block erase command for end of erase (59h 04h FFh FFh FFh A7h). The MCU sends a response (06h).
- (4) To confirm whether erase ready operation has ended, send a boot mode status inquiry command (4Fh). The MCU sends a response to the boot mode status inquiry command when erase ready operation has ended. If erase ready operation has not ended, the MCU sends an error response (80h 4Fh). If the programmer receives an error response, restart the MCU in boot mode, and start again from section 41.10.9.1, Bit Rate Automatic Adjustment Procedure.

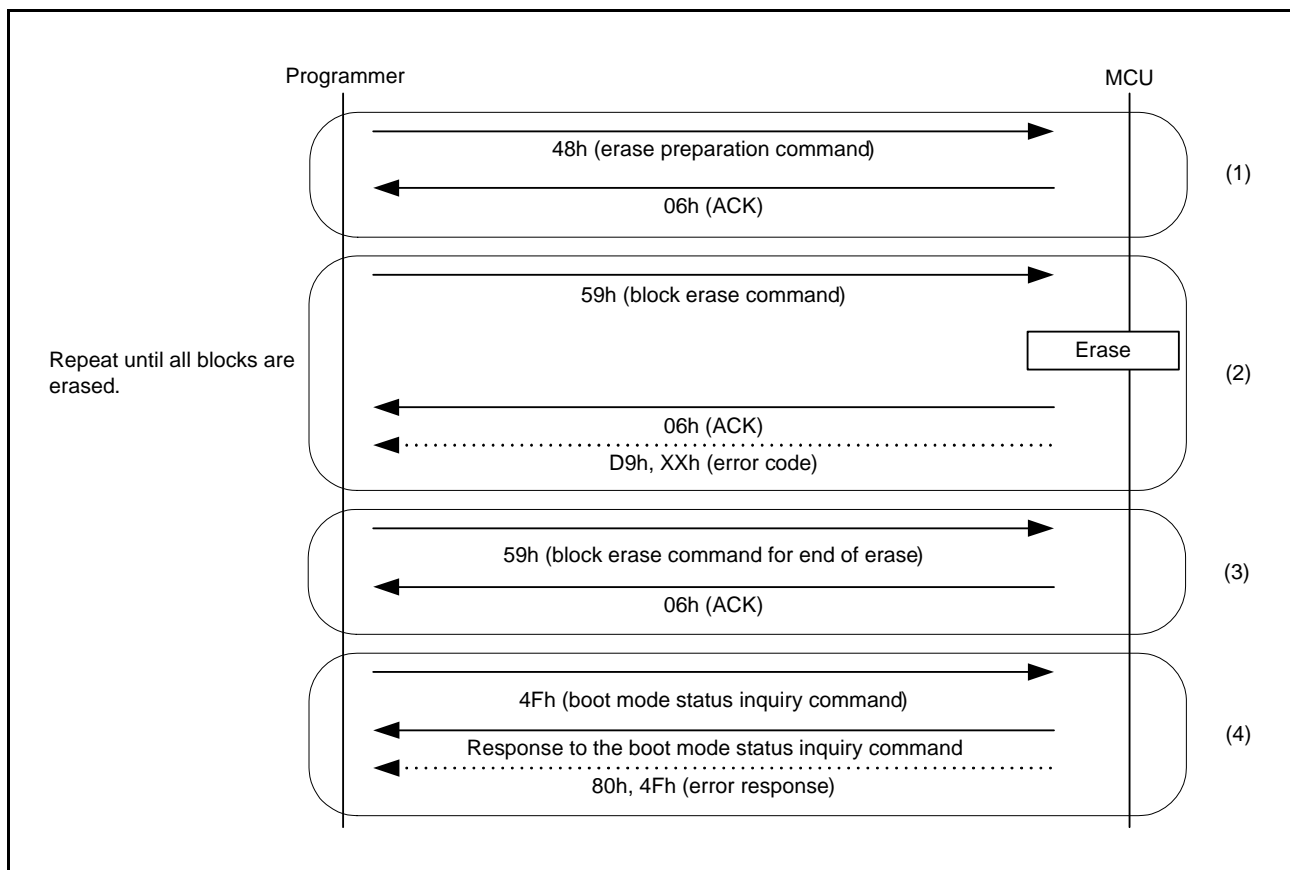


Figure 41.33 Procedure to Send Commands in Erase Ready Operation

### 41.10.9.7 Erase the User Area and Data Area

Erase blocks that are programmed in the user area and data area to program a user program.

- (1) Send an erase preparation command (48h) to place the MCU in the erase wait state. The MCU enters the erase wait state and sends a response (06h).
- (2) Send a block erase command (59h). Set the block that is erased in the block start address. When the selected block is erased successfully, the MCU sends a response (06h). When the MCU fails to receive, the MCU sends an error response (D9h).
- (3) In order to place the MCU in the program/erase state, send a block erase command for end of erase (59h 04h FFh FFh FFh FFh A7h). The MCU enters the program/erase state and sends a response (06h).

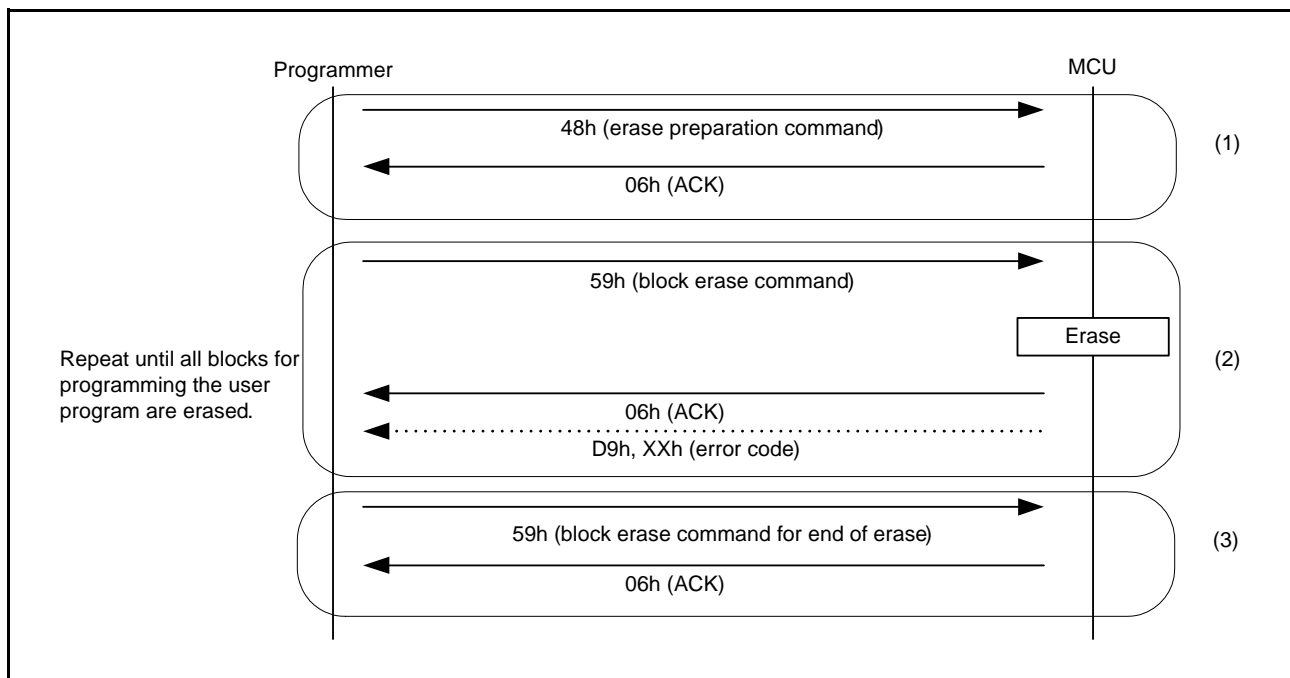


Figure 41.34 Procedure to Erase the User Area and Data Area

### 41.10.9.8 Program the User Area and Data Area

Program a user program in the user area and data area.

- (1) Send the user/data area program preparation command (43h) to place the MCU in the program wait state. The MCU enters the program wait state and sends a response (06h).
- (2) Send the program command (50h). Set the program address to an address aligned on a 256-byte boundary. Set program data in 256 bytes. When the data is programmed successfully, the MCU sends a response (06h). When the MCU fails to receive, the MCU sends an error response (D0h, XXh).
- (3) In order to place the MCU in the program/erase state, send the program command for end of program (50h FFh FFh FFh B4h). The MCU enters the program/erase state and sends a response (06h).

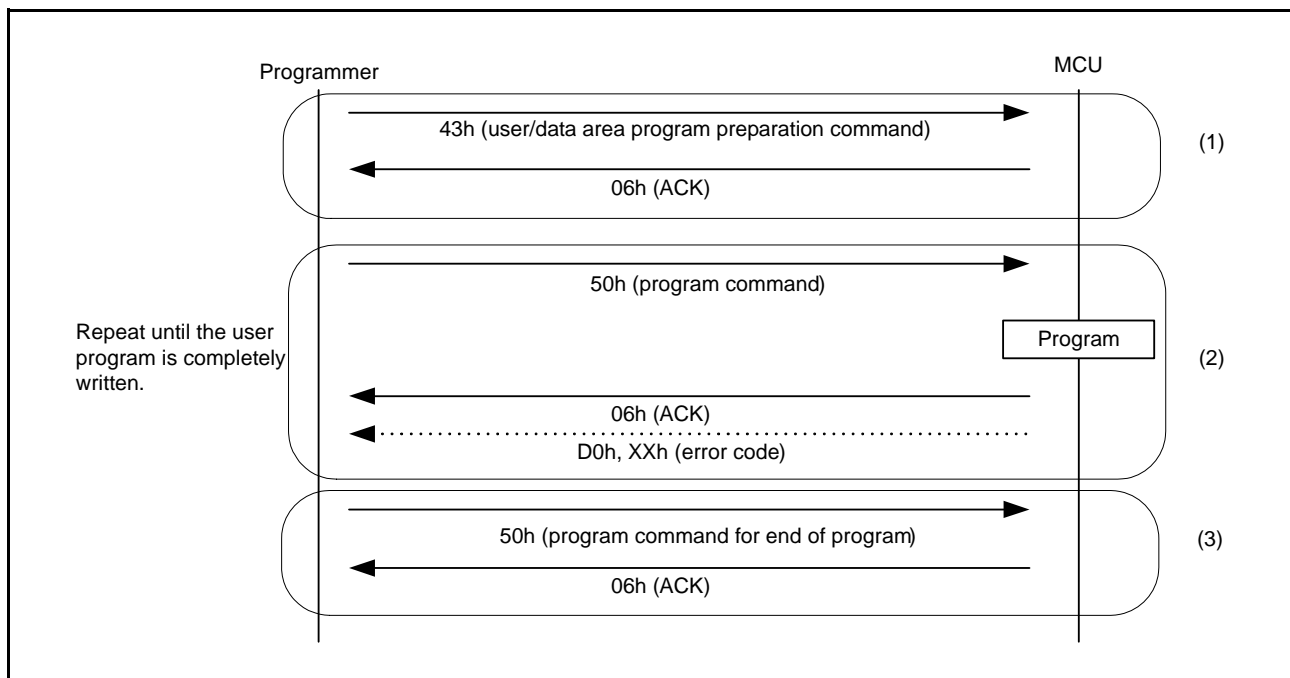


Figure 41.35 Procedure to Program the User Area and Data Area

### 41.10.9.9 Check Data in the User Area

Read and check, checksum, and blank check the user area to check the programmed data in the user area.

- (1) The read and check operation is used to read data in the user area and compare the read data with the programmed data to check if the program operation is performed successfully. Send a memory read command (52h) to read data in the user area. Set the read area in the read address and read size. The MCU sends data for the size from the address set in the read address. If the MCU fails to receive, the MCU sends an error response (D2h).
- (2) Send the user area checksum command (4Bh) to check program data using the checksum of user area. The MCU reads data from the start address to the end address of the user area in bytes and send the read result as a response.
- (3) Send a user area blank check command (4Dh) to check if the user area has data. When there is no data in the start address to the end address of the user area, the MCU sends a response (06h). When there is at least 1 byte of data, the MCU sends a response (CDh, 52h) indicating that the selected area is not blank.

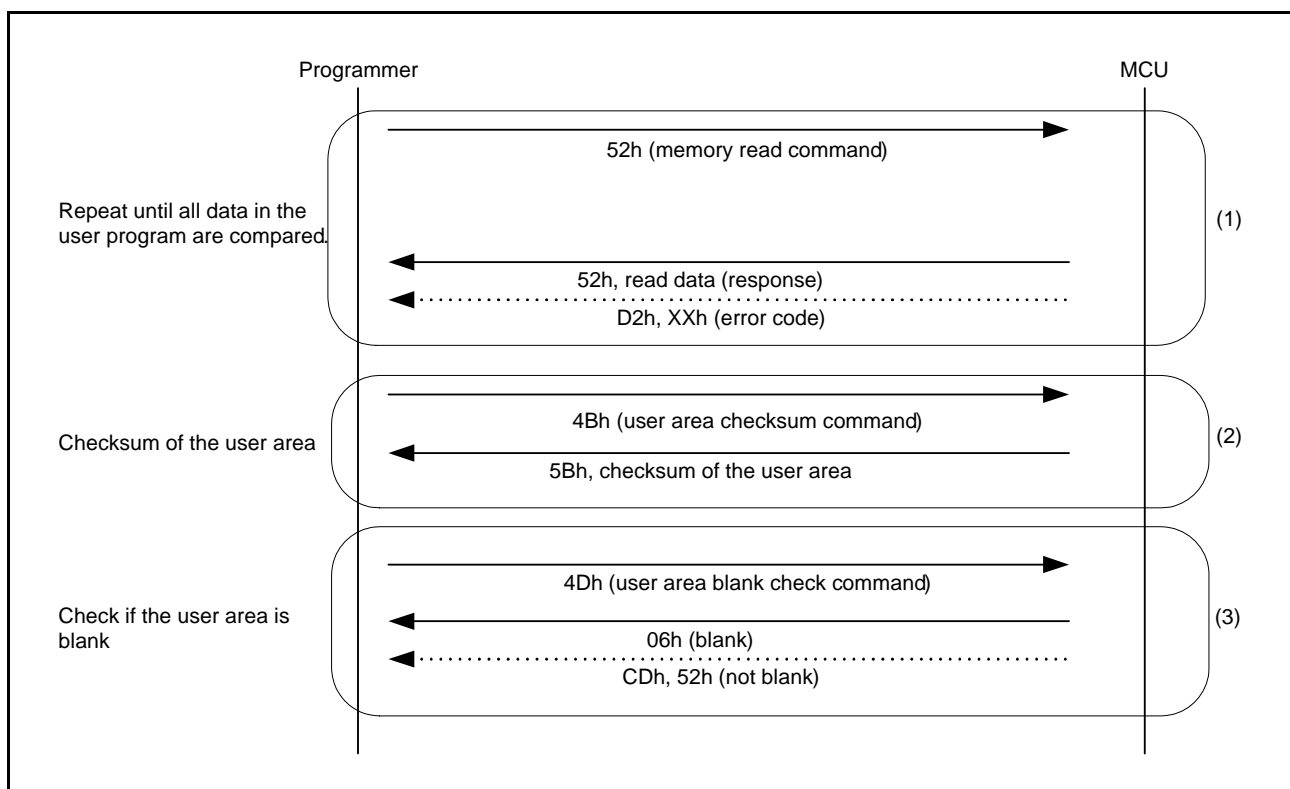


Figure 41.36 Procedure to Check Data in the User Area

### 41.10.9.10 Check Data in the Data Area

Read and check, checksum, and blank check the user area to check the programmed data in the data area.

- (1) The read and check operation is used to read data in the data area and compare the read data with the programmed data to check if the program operation is performed successfully. Send a memory read command (52h) to read data in the data area. Set the read area in the read address and read size. The MCU sends data for the size from the address set in the read address. If the MCU fails to receive, the MCU sends an error response (D2h).
- (2) Send the data area checksum command (61h) to check program data using the checksum of data area. The MCU reads data from the start address to the end address of the data area in bytes and send the read result as a response.
- (3) Send the data area blank check command (62h) to check if the data area has data. When there is no data in the start address to the end address of the data area, the MCU sends a response (06h). When there is at least 1 byte of data, the MCU sends a response (E2h, 52h) indicating that the selected area is not blank.

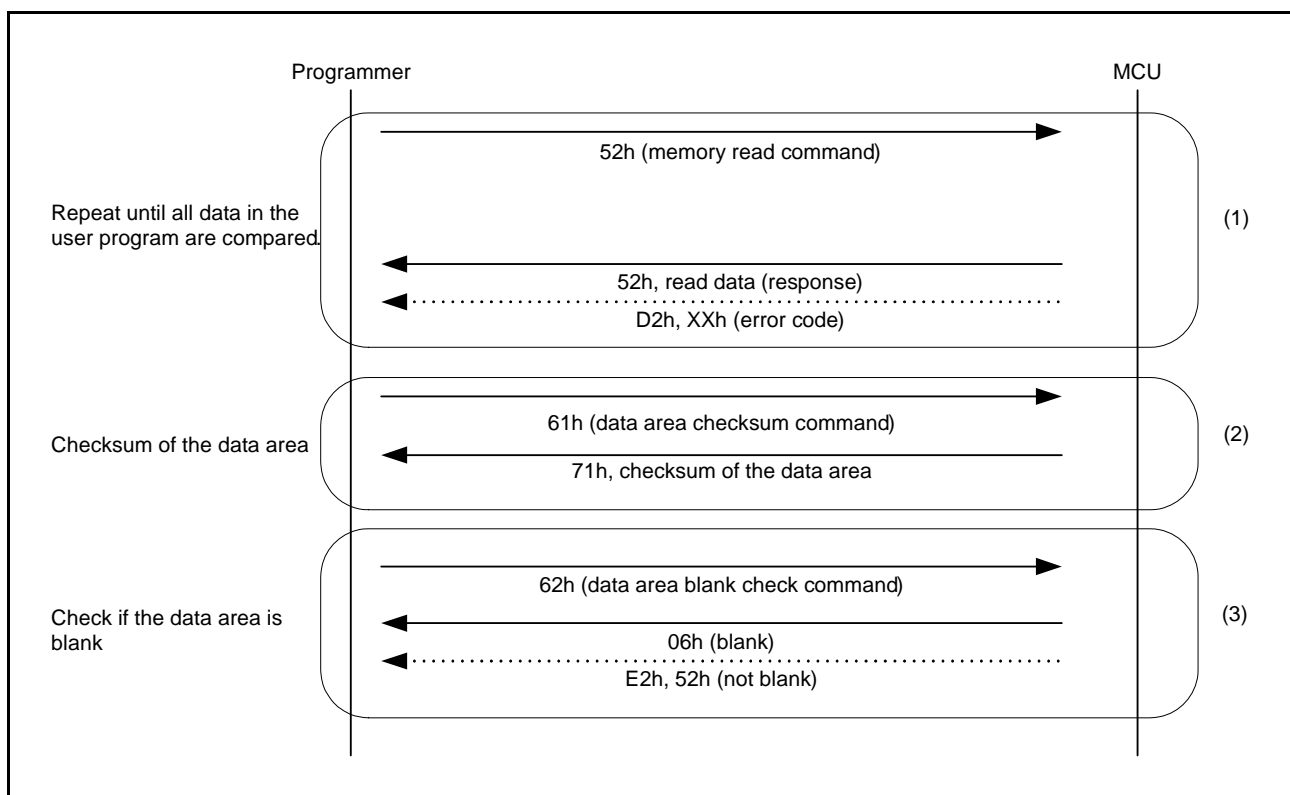


Figure 41.37 Procedure to Check Data in the Data Area

### 41.10.9.11 Set the Access Window in the User Area

Set the access window to avoid unintentionally rewriting the user area by the self-programming.

- (1) Send the access window program command (74h) to set the access window or clear the access window settings. When setting the access window, set 00h in the access window, set the start address of the area that can be programmed by self-programming in the access window start address LH and access window start address HL, and set the end address of the area that can be programmed by the self-programming in the access window end address LH and access window end address HL. When clearing the access window settings, set FFh in the access window, access window start address LH, access window start address HL, access window end address LH, and access window end address HL. When the MCU writes the addresses, the MCU sends a response (06h). If the MCU fails to receive, the MCU sends an error response (F4h).
- (2) Send the access window read command (73h) to confirm the access window settings. The MCU sends the current access window settings. When the MCU fails to receive, the MCU sends an error response (F3h).

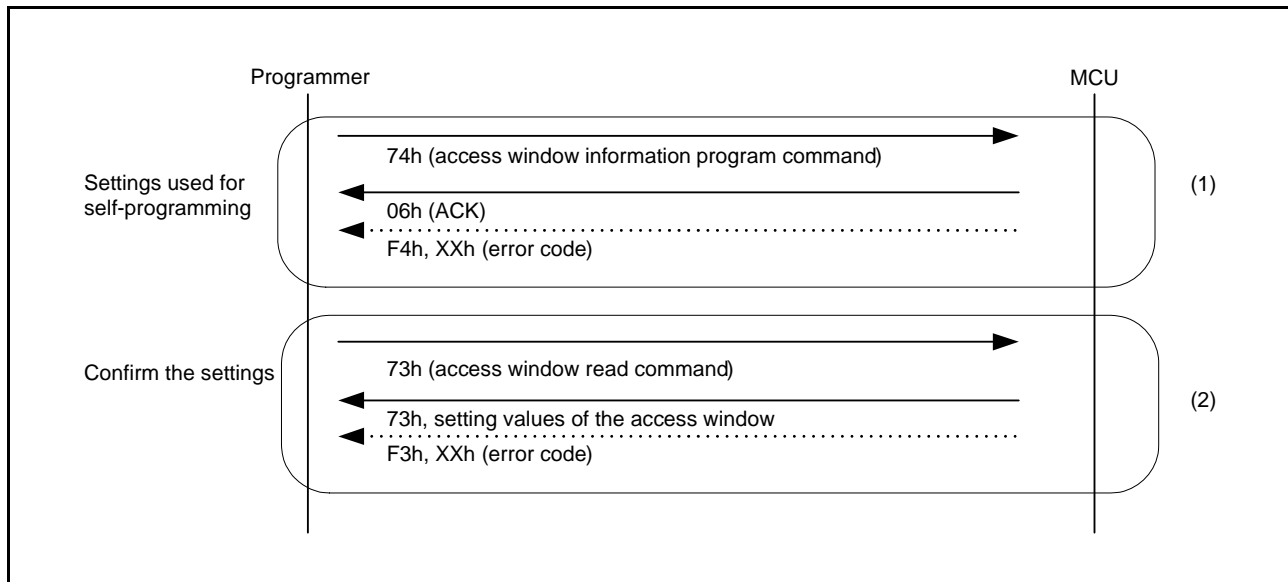


Figure 41.38 Procedure to Set the Access Window in the User Area



## 41.11 Rewriting by Self-Programming

### 41.11.1 Overview

The MCU supports rewriting of the flash memory by the user program. The ROM and E2 DataFlash can be rewritten by preparing a routine to rewrite the flash memory (flash rewrite routine) in the user program.

When rewriting the E2 DataFlash, the BGO can be used to execute the flash rewrite routine on the ROM. The E2 DataFlash can also be rewritten by executing the flash rewrite routine that is transferred on the RAM in advance.

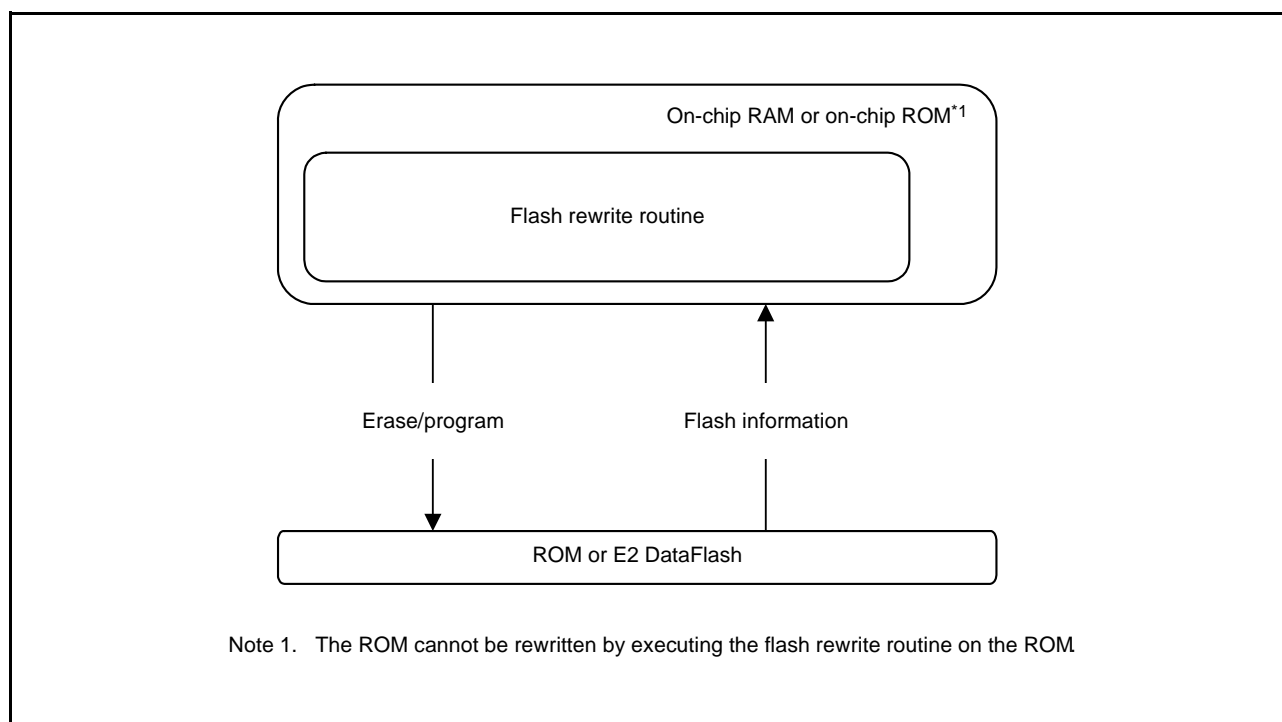


Figure 41.39 Self-Programming Overview

## 41.12 Usage Notes

### (1) Access the Block Where Erase Operation is Forcibly Stopped

When forcibly stopping an erase operation, data in the block where the erase operation is aborted is undefined. To avoid malfunctions caused by reading undefined data, do not execute instructions or read data in the block where an erase operation is forcibly stopped.

### (2) Processing After Forced Stop of Erase Operation

When an erase operation is forcibly stopped, issue a block erase command again to the same block.

### (3) Additional Programming Disabled

The same address cannot be programmed more than once. When programming an area that has been already programmed, erase the area first.

### (4) Reset during Program/Erase

If inputting a reset from the RES# pin, release the reset after reset input time of at least tRESW (refer to section 42, Electrical Characteristics) within the range of the operating voltage defined in the electrical characteristics. The IWDT reset and software reset can be used regardless of tRESW.

### (5) Non-maskable Interrupt Disabled during Program/Erase

When a non-maskable interrupt (NMI pin interrupt, oscillation stop detection interrupt, IWDT underflow/refresh error, voltage monitoring 1 interrupt, or voltage monitoring 2 interrupt) occurs during a program/erase operation, the vectors are fetched from the ROM, and undefined data is read. Therefore, do not generate a non-maskable interrupt during a program/erase operation on the ROM.

(The description in (5) applies only to the ROM.)

### (6) Location of Interrupt Vectors during a Program/Erase Operation

When an interrupt occurs during a program/erase operation, the vector may be fetched from the ROM. To avoid fetching the vector from the ROM, set the destination for fetching interrupt vectors to an area other than the ROM with the CPU interrupt table register (INTB).

### (7) Program/Erase in Low-Speed Operating Mode

Do not program or erase the flash memory when low-speed operating mode is selected by the SOPCCR register for low-power consumption functions.

### (8) Abnormal Termination during Program/Erase

When the voltage exceeds the range of the operating voltage during a program/erase operation or when a program/erase operation is not completed successfully due to a reset or prohibited actions described in (9), erase the area again.

### (9) Actions Prohibited during Program/Erase

To prevent the damage to the flash memory, comply with the following instructions.

- Do not use the MCU power supply that is outside the operating voltage range.
- Do not update the value of the OPCCR.OPCM[2:0] bits.
- Do not update the value of the SOPCCR.SOPCM bit.
- Do not change the clock source select bit in the SCKCR3 register.
- Do not enable switching clock sources by setting the RSTCKCR.RSTCKEN bit when exiting sleep mode.
- Do not change the division ratio of the flash interface clock (FCLK).
- Do not place the MCU in deep sleep mode or software standby mode.
- Do not access the E2 DataFlash during a program/erase operation to the ROM.
- Do not change the DFLCTL.DFLEN bit value during a program/erase operation to the E2 DataFlash.

### (10) FCLK during Program/Erase

For programming/erasure by self-programming, it is necessary to set the FlashIF clock (FCLK) and specify an integer FCLK frequency in the flash initial setting register (FISR). Note that when the FCLK is 4 to 32 MHz, a rounded-up value should be set for a non-integer frequency such as 12.5 MHz (i.e. 12.5 MHz should be set rounded up to 13 MHz).

### 41.13 Usage Notes in Boot Mode

- (1) Notes on Communication Errors in Boot Mode  
When communication with the MCU cannot be performed properly, reset and start up in boot mode again.
- (2) Notes on Power Supply Voltage in Boot Mode (SCI)  
When the bit rate exceeds 500 kbps in boot mode (SCI), use a voltage that is 3.0 or higher, and 3.6 V or lower.
- (3) Notes on Using Boot Mode (SCI) and USB0 Self-Powered Mode  
When boot mode (SCI) is selected, do not connect a USB cable.
- (4) Notes on Using USB0 Battery Charging Function  
Do not select boot mode (SCI).
- (5) Notes on Option-Setting Memory in Boot Mode  
The settings of registers OFS0, OFS1 and MDE are disabled.
- (6) Notes on Clocks in Boot Mode (USB Interface)  
When USB interface mode is selected, externally input a clock to the EXTAL or XTAL pin, or connect a crystal or ceramic resonator to supply a clock.  
Use a 6, 8, 12, or 16 MHz external clock in boot mode (USB interface). An clock other than a 6, 8, 12, or 16 MHz external clock cannot be used.
- (7) Notes on Power Supply Voltage in Boot Mode (USB Interface)  
Use a voltage that is 3.0 V or higher, and 3.6 V or lower in boot mode (USB interface). A voltage that is 1.8 V or higher, and lower than 3.0 V cannot be used.
- (8) Notes on Switching the Start-Up Area  
Switch the start-up area by self-programming.

## 42. Electrical Characteristics

### 42.1 Absolute Maximum Ratings

**Table 42.1 Absolute Maximum Ratings**

 Conditions:  $V_{SS} = AVSS0 = VREFL0 = VREFL = VSS\_USB = 0\text{ V}$ 

Item		Symbol	Value	Unit
Power supply voltage		VCC, VCC_USB	-0.3 to +4.6	V
Input voltage (except for ports for 5 V tolerant*1)		$V_{in}$	-0.3 to VCC + 0.3	V
Input voltage (ports for 5 V tolerant*1)		$V_{in}$	-0.3 to +6.5	V
Reference power supply voltage		VREFH0	-0.3 to AVCC0 + 0.3	V
		VREFH		
Analog power supply voltage		AVCC0	-0.3 to +4.6	V
Analog input voltage		$V_{AN}$	-0.3 to AVCC0 + 0.3 (when AN000 to AN007 and AN021 used) -0.3 to VCC + 0.3 (when AN008 to AN015 used)	V
LCD voltage	$V_{L1}$ voltage	$V_{L1}$	-0.3 to +2.8	V
	$V_{L2}$ voltage	$V_{L2}$	-0.3 to +6.5	
	$V_{L3}$ voltage	$V_{L3}$	-0.3 to +6.5	
	$V_{L4}$ voltage	$V_{L4}$	-0.3 to +6.5	
Operating temperature*2		$T_{opr}$	-40 to +85 -40 to +105	°C
Storage temperature		$T_{stg}$	-55 to +125	°C

Caution: Permanent damage to the MCU may result if absolute maximum ratings are exceeded.

To preclude any malfunctions due to noise interference, insert capacitors of high frequency characteristics between the VCC and VSS pins, between the AVCC0 and AVSS0 pins, between the VCC\_USB and VSS\_USB pins, between the VREFH0 and VREFL0 pins, and between the VREFH and VREFL pins. Place capacitors of about 0.1  $\mu\text{F}$  as close as possible to every power supply pin and use the shortest and heaviest possible traces. Also, connect capacitors as stabilization capacitance.

Connect the VCL pin to a VSS pin via a 4.7  $\mu\text{F}$  capacitor. The capacitor must be placed close to the pin.

Note 1. Ports 16, 17, A6, and B0 are 5 V tolerant.

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Note 2. The upper limit of operating temperature is 85°C or 105°C, depending on the product. For details, refer to Table 1.3, List of Products.

**Table 42.2 Recommended Operating Conditions**

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Power supply voltages	VCC	When USB not used	1.8	—	3.6	V
		When USB used	3.0	—	3.6	
	VSS		—	0	—	
USB power supply voltages	VCC_USB		—	VCC	—	V
	VSS_USB		—	0	—	
Analog power supply voltages	AVCC0*1 to *3		1.8	—	3.6	V
	AVSS0		—	0	—	

Note 1. AVCC0 and VCC can be set individually within the operating range, but there is the following relationship between the voltage applied to the PJ0 and PJ2 pins, VCC, and AVCC0.

When 12-bit D/A converter used: Voltage applied to port J0 and J2 pins (D/A output voltage)  $\leq$  VCC

When general ports selected:  $VCC \leq AVCC0$

Note 2. For details, refer to section 35.8.11, Voltage Range of Analog Power Supply Pins.

Note 3. Sequence of Powering on AVCC0 and VCC

When powering on AVCC0 and VCC, power them on at the same time or VCC first.

## 42.2 DC Characteristics

**Table 42.3 DC Characteristics (1)**Conditions:  $VCC = VCC\_USB = 2.7$  to  $3.6$  V,  $AVCC0 = 2.7$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	RIIC input pin (except for SMBus, 5 V tolerant)	$V_{IH}$	$VCC \times 0.7$	—	5.8	V	
	Ports 16, 17, port A6, port B0 (5 V tolerant)		$VCC \times 0.8$	—	5.8		
	Other pins Ports 02, 04, 07, ports 10 to 15 ports 20 to 27, ports 30 to 32, 35 ports 50 to 56, ports A0 to A7 ports B0 to B7 ports C0 to C7 ports D0 to D4 ports E0 to E7 ports F6, F7 port H7, RES# ports J0, J2, J3*1		$VCC \times 0.8$	—	$VCC + 0.3$		
	RIIC input pin (except for SMBus)	$V_{IL}$	-0.3	—	$VCC \times 0.3$		
	Other than RIIC input pin		-0.3	—	$VCC \times 0.2$		
	RIIC input pin (except for SMBus)	$\Delta V_T$	$VCC \times 0.05$	—	—		
	Other than RIIC input pin		$VCC \times 0.1$	—	—		
Input level voltage (except for Schmitt trigger input pins)	MD	$V_{IH}$	$VCC \times 0.9$	—	$VCC + 0.3$	V	
	XTAL (external clock input)		$VCC \times 0.8$	—	$VCC + 0.3$		
	Ports 40 to 44, 46, ports J6, J7, ports 90 to 92		$AVCC0 \times 0.7$	—	$AVCC0 + 0.3$		
	RIIC input pin (SMBus)		2.1	—	$VCC + 0.3$		
	MD	$V_{IL}$	-0.3	—	$VCC \times 0.1$		
	XTAL (external clock input)		-0.3	—	$VCC \times 0.2$		
	Ports 40 to 44, 46, ports J6, J7, ports 90 to 92		-0.3	—	$AVCC0 \times 0.3$		
	RIIC input pin (SMBus)		-0.3	—	0.8		

Note 1. There are restrictions on  $AVCC0$  and  $VCC$  depending on the usage conditions for the 12-bit D/A converter and I/O ports. When using ports J0 and J2 multiplexed with DA0 and DA1 as general I/O ports, make sure that  $VCC \leq AVCC0$ .

**Table 42.4 DC Characteristics (2)**Conditions:  $VCC = VCC\_USB = 1.8$  to  $2.7$  V,  $AVCC0 = 1.8$  to  $2.7$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	Ports 16, 17, port A6, port B0 (5 V tolerant)	$V_{IH}$	$VCC \times 0.8$	—	5.8	V	
	Other pins Ports 02, 04, 07, ports 10 to 15 ports 20 to 27, ports 30 to 32, 35 ports 50 to 56, ports A0 to A7 ports B0 to B7 ports C0 to C7 ports D0 to D4 ports E0 to E7 ports F6, F7 port H7, RES# ports J0, J2, J3*1		$VCC \times 0.8$	—	$VCC + 0.3$		
	All pins	$V_{IL}$	-0.3	—	$VCC \times 0.2$		
	All pins	$\Delta V_T$	$VCC \times 0.01$	—	—		
Input level voltage (except for Schmitt trigger input pins)	MD	$V_{IH}$	$VCC \times 0.9$	—	$VCC + 0.3$	V	
	XTAL (external clock input)		$VCC \times 0.8$	—	$VCC + 0.3$		
	Ports 40 to 44, 46, ports J6, J7, ports 90 to 92		$AVCC0 \times 0.7$	—	$AVCC0 + 0.3$		
	MD	$V_{IL}$	-0.3	—	$VCC \times 0.1$		
	XTAL (external clock input)		-0.3	—	$VCC \times 0.2$		
	Ports 40 to 44, 46, ports J6, J7, ports 90 to 92		-0.3	—	$AVCC0 \times 0.3$		

Note 1. There are restrictions on  $AVCC0$  and  $VCC$  depending on the usage conditions for the 12-bit D/A converter and I/O ports. When using ports J0 and J2 multiplexed with DA0 and DA1 as general I/O ports, make sure that  $VCC \leq AVCC0$ .

**Table 42.5 DC Characteristics (3)**Conditions:  $VCC = AVCC0 = VCC\_USB = 1.8$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	RES#, MD, port 35 port H7	$ I_{in} $	—	—	1.0	$\mu\text{A}$	$V_{in} = 0$ V, $VCC$
Three-state leakage current (off-state)	Ports for 5 V tolerant	$ I_{TSI} $	—	—	1.0	$\mu\text{A}$	$V_{in} = 0$ V, 5.8 V
	Pins other than above		—	—	1.0		$V_{in} = 0$ V, $VCC$
Input capacitance	All input pins (except for port 35, port 16, USB0_DM, USB0_DP)	$C_{in}$	—	—	15	pF	$V_{in} = 0$ V Frequency: 1 MHz $T_a = 25^\circ\text{C}$
	Port 35, port 16, USB0_DM, USB0_DP		—	—	30		

**Table 42.6 DC Characteristics (4)**Conditions:  $VCC = AVCC0 = VCC\_USB = 1.8$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input pull-up resistor	All ports (except for ports 35, PH7)	$R_U$	10	20	100	k $\Omega$	$V_{in} = 0$ V

**Table 42.7 DC Characteristics (5)**

Conditions: VCC = AVCC0 = VCC\_USB = 1.8 to 3.6 V, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

Item				Symbol	Typ *4	Max	Unit	Test Conditions			
Supply current*1	High-speed operating mode	Normal operating mode	No peripheral operation*2	ICLK = 32 MHz	I <sub>CC</sub>	3.6	—	mA			
				ICLK = 16 MHz		2.4	—				
				ICLK = 8 MHz		1.8	—				
			All peripheral operation: Normal*3	ICLK = 32 MHz		14.0	—				
				ICLK = 16 MHz		7.9	—				
				ICLK = 8 MHz		4.9	—				
			All peripheral operation: Max.*3	ICLK = 32 MHz		—	30.0				
				Sleep mode		No peripheral operation*2	ICLK = 32 MHz			1.9	—
							ICLK = 16 MHz			1.5	—
		ICLK = 8 MHz	1.3		—						
		All peripheral operation: Normal*3	ICLK = 32 MHz	8.2	—						
			ICLK = 16 MHz	4.8	—						
			ICLK = 8 MHz	3.1	—						
		Deep sleep mode	No peripheral operation*2	ICLK = 32 MHz	1.1	—					
				ICLK = 16 MHz	0.95	—					
	ICLK = 8 MHz			0.86	—						
	All peripheral operation: Normal*3		ICLK = 32 MHz	6.4	—						
			ICLK = 16 MHz	3.8	—						
			ICLK = 8 MHz	2.4	—						
	Increase during BGO operation*5					2.5	—				
	Middle-speed operating modes		Normal operating mode	No peripheral operation*6	ICLK = 12 MHz	I <sub>CC</sub>	2.1	—		mA	
					ICLK = 8 MHz		1.4	—			
		ICLK = 1 MHz			0.77		—				
		All peripheral operation: Normal*7		ICLK = 12 MHz	6.3		—				
ICLK = 8 MHz				4.6	—						
ICLK = 1 MHz				1.6	—						
All peripheral operation: Max.*7		ICLK = 12 MHz		—	14.2						
		Sleep mode		No peripheral operation*6	ICLK = 12 MHz		1.4	—			
					ICLK = 8 MHz		0.90	—			
ICLK = 1 MHz					0.68		—				
All peripheral operation: Normal*7		ICLK = 12 MHz		3.9	—						
		ICLK = 8 MHz		2.9	—						
		ICLK = 1 MHz		1.4	—						
Deep sleep mode		No peripheral operation*6		ICLK = 12 MHz	1.1		—				
				ICLK = 8 MHz	0.63		—				
			ICLK = 1 MHz	0.55	—						
		All peripheral operation: Normal*7	ICLK = 12 MHz	3.3	—						
			ICLK = 8 MHz	2.4	—						
			ICLK = 1 MHz	1.2	—						
Increase during BGO operation*5					2.5	—					

Item				Symbol	Typ *4	Max	Unit	Test Conditions
Supply current*1	Low-speed operating mode	Normal operating mode	No peripheral operation*8	I <sub>CC</sub>	4.3	—	μA	
			All peripheral operation: Normal*9, *10		15.0	—		
			All peripheral operation: Max.*9, *10		—	62		
		Sleep mode	No peripheral operation*8	2.3	—			
			All peripheral operation: Normal*9	8.6	—			
			Deep sleep mode	No peripheral operation*8	1.7	—		
	All peripheral operation: Normal*9	7.0		—				

- Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.
- Note 2. Clock supply to the peripheral functions is stopped. This does not include BGO operation. The clock source is PLL. FCLK and PCLK are set to divided by 64.
- Note 3. Clocks are supplied to the peripheral functions. This does not include BGO operation. The clock source is PLL. FCLK and PCLK are set to the same frequency as ICLK.
- Note 4. Values when VCC = 3.3 V.
- Note 5. This is the increase for programming or erasure of the ROM or E2 DataFlash during program execution.
- Note 6. Clock supply to the peripheral functions is stopped. The clock source is PLL when ICLK = 12 MHz, and HOCO otherwise. FCLK and PCLK are set to divided by 64.
- Note 7. Clocks are supplied to the peripheral functions. The clock source is PLL when ICLK = 12 MHz, and HOCO otherwise. FCLK and PCLK are set to the same frequency as ICLK.
- Note 8. Clock supply to the peripheral functions is stopped. The clock source is the sub-clock oscillator. FCLK and PCLK are set to divided by 64.
- Note 9. Clocks are supplied to the peripheral functions. The clock source is the sub-clock oscillator. FCLK and PCLK are set to the same frequency as ICLK.
- Note 10. Values when the MSTPCRA.MSTPA17 bit (12-bit A/D converter module stop bit) is set to "transition to the module stop state is made".

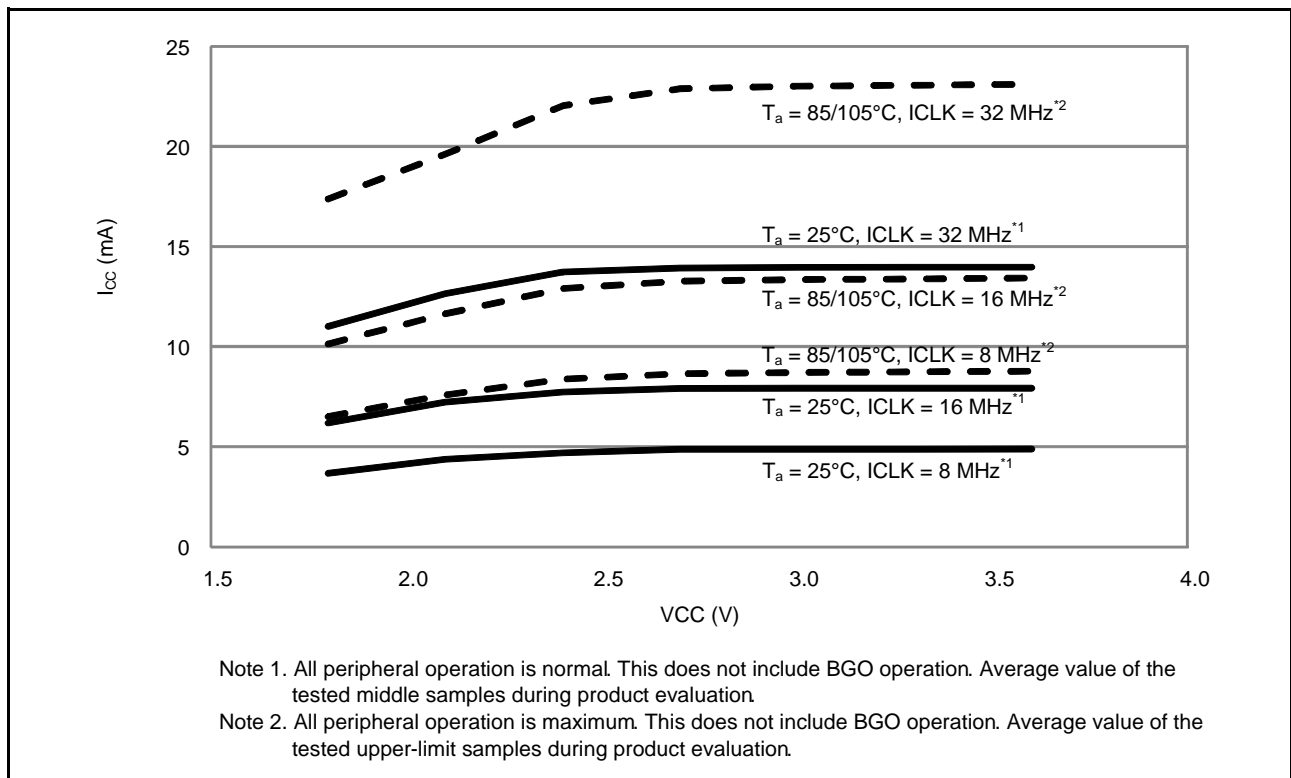
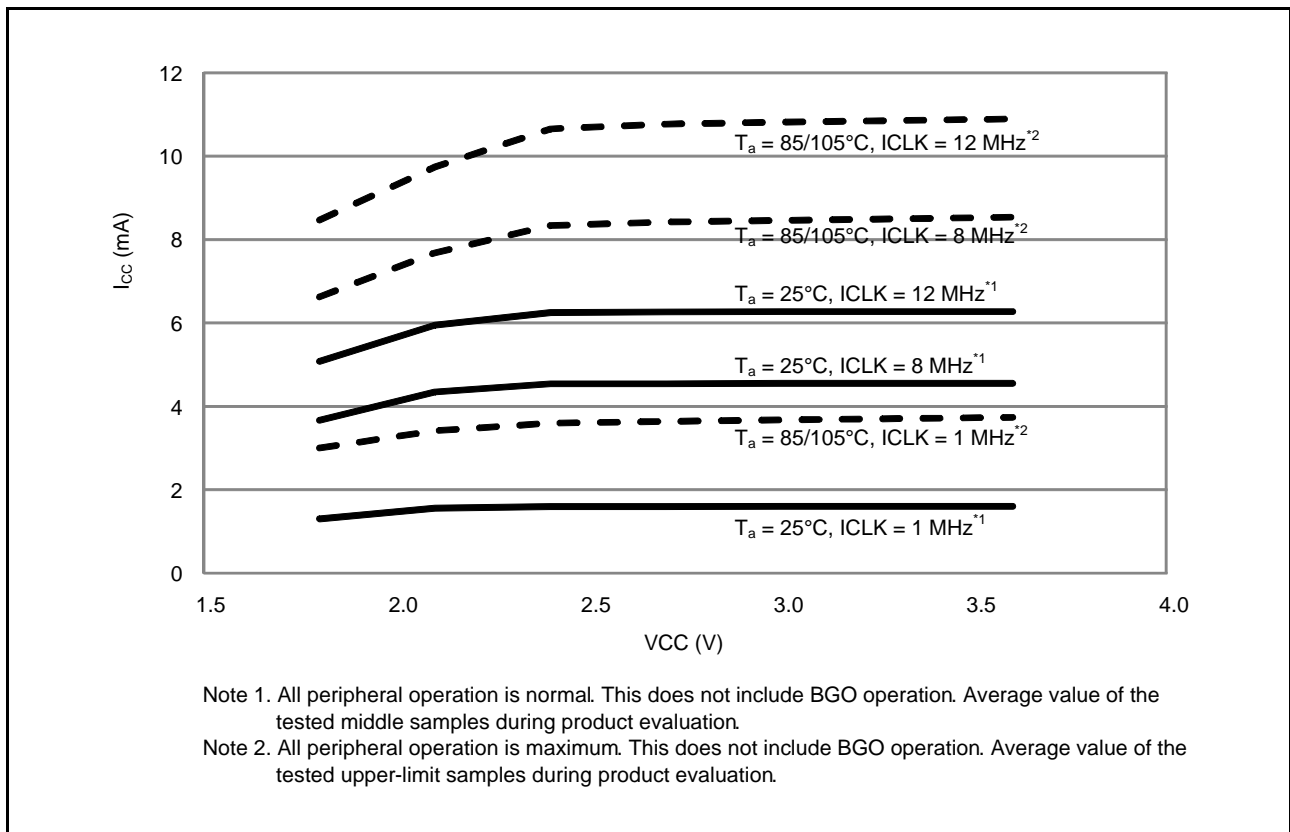
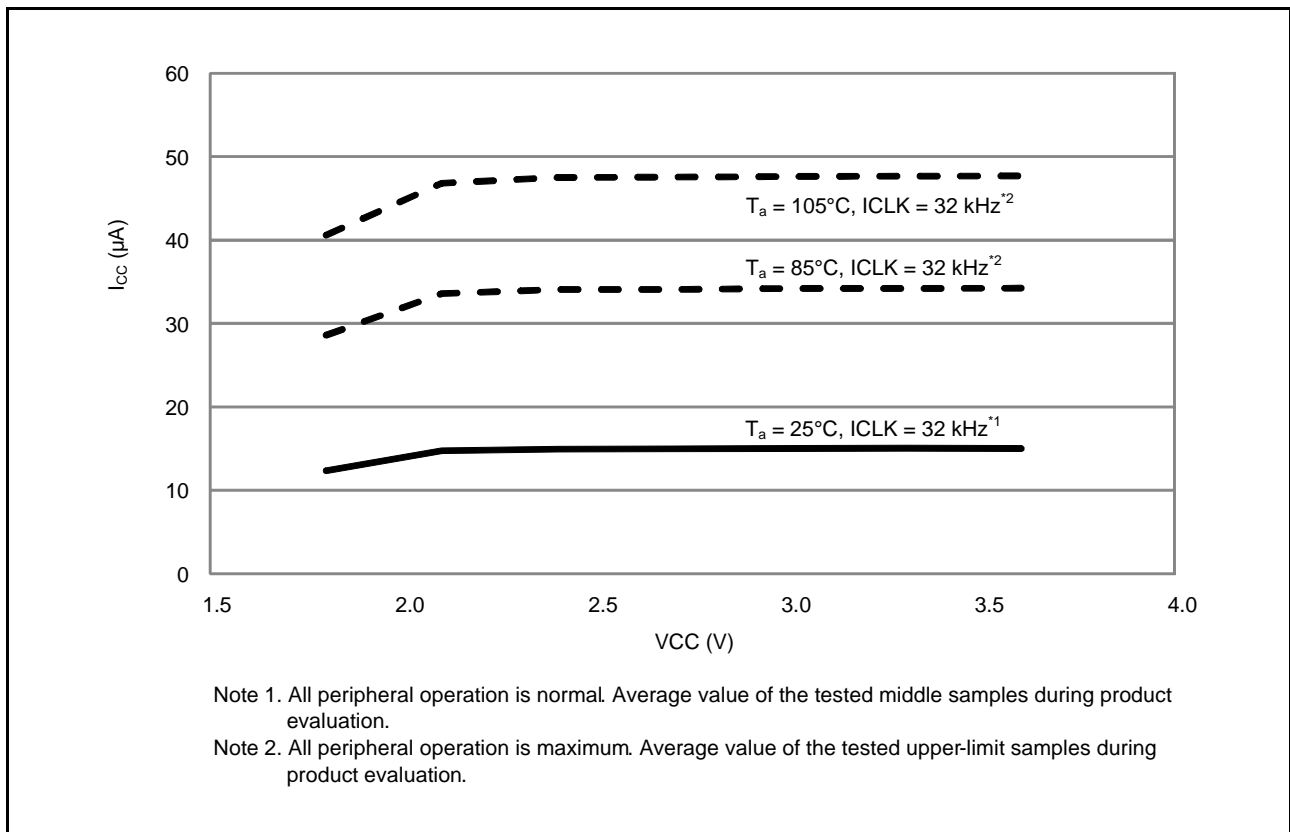


Figure 42.1 Voltage Dependency in High-Speed Operating Mode (Reference Data)





**Figure 42.2 Voltage Dependency in Middle-Speed Operating Mode (Reference Data)**



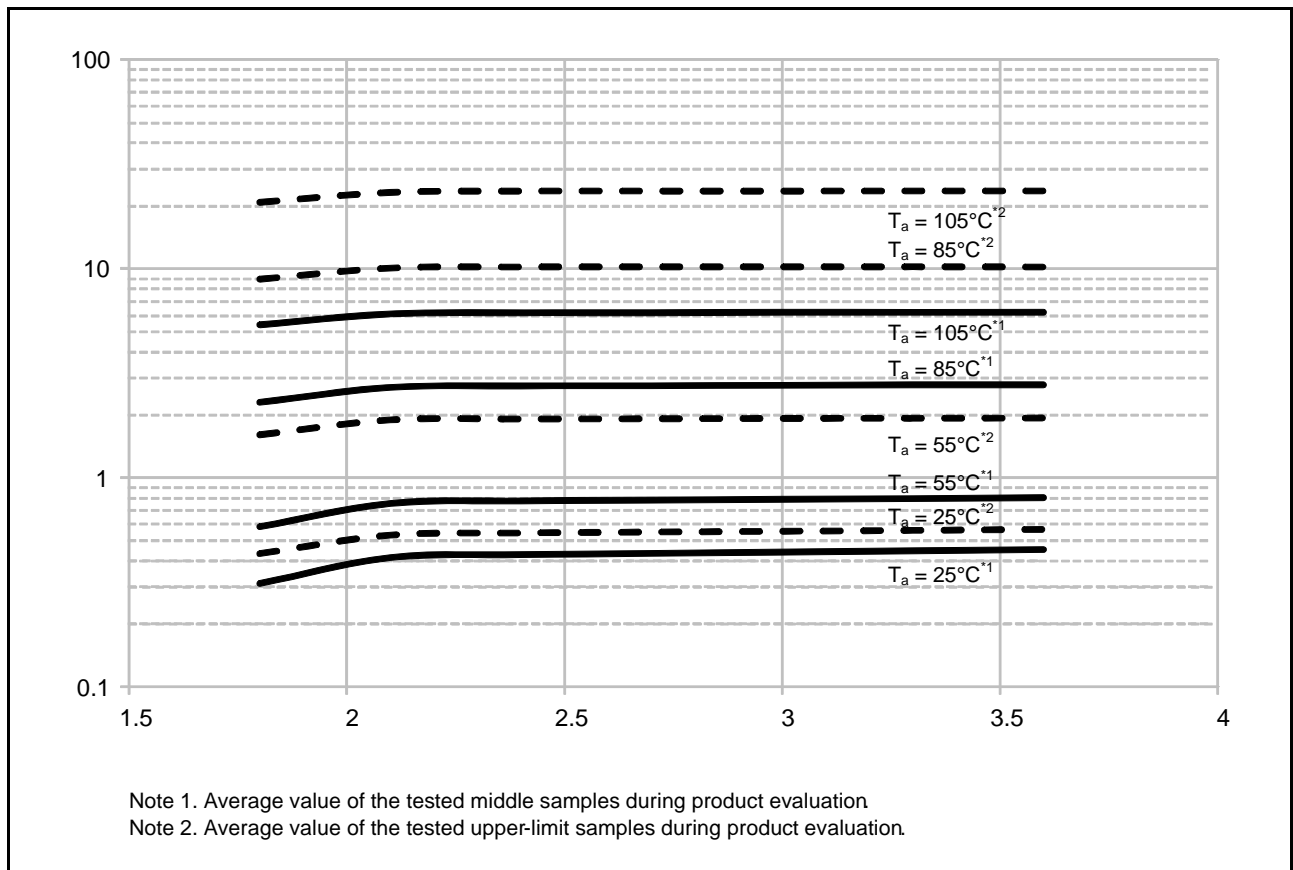
**Figure 42.3 Voltage Dependency in Low-Speed Operating Mode (Reference Data)**

**Table 42.8 DC Characteristics (6)**

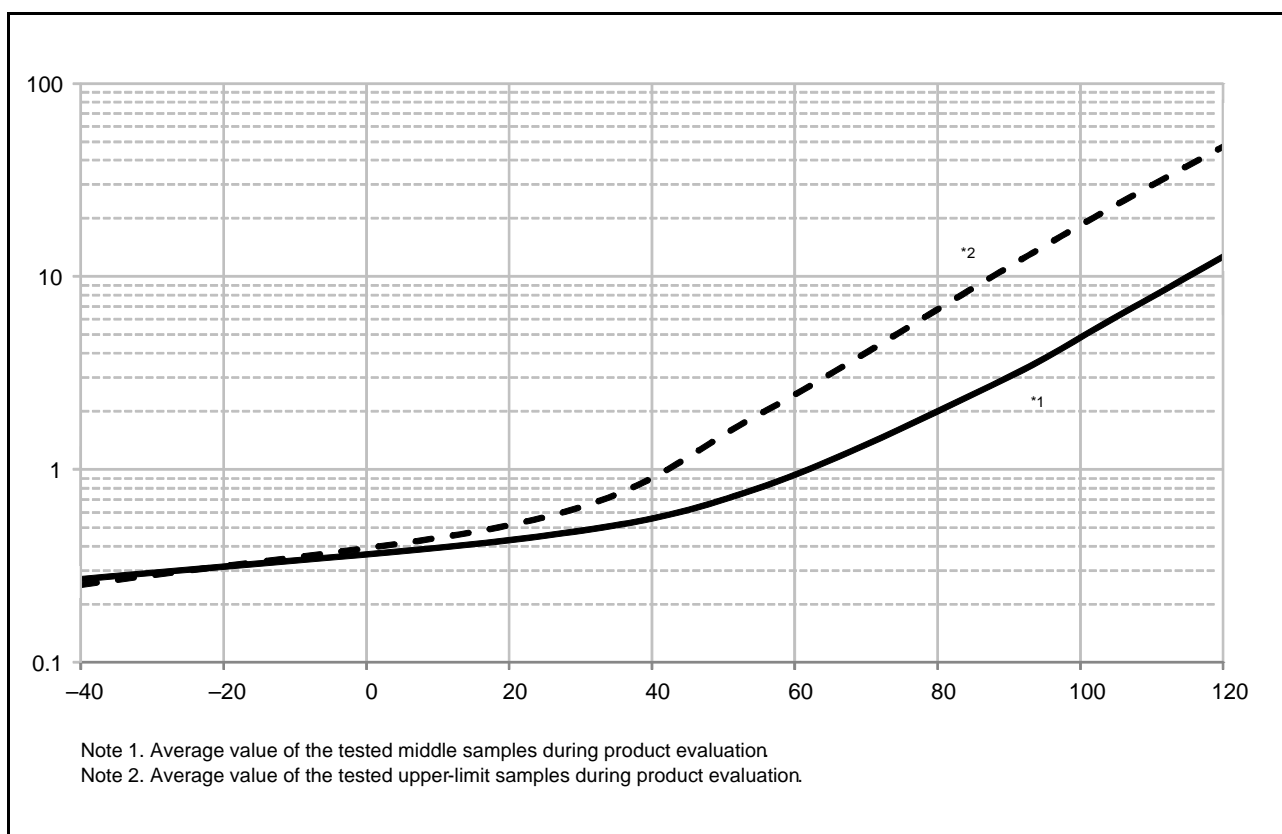
Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 1.8$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

Item		Symbol	Typ.*3	Max.	Unit	Test Conditions	
Supply current*1	Software standby mode*2	$I_{CC}$	$T_a = 25^\circ\text{C}$	0.44	0.98	$\mu\text{A}$	
			$T_a = 55^\circ\text{C}$	0.80	3.47		
			$T_a = 85^\circ\text{C}$	2.7	12.0		
			$T_a = 105^\circ\text{C}$	6.17	42.7		
	Increment for RTC operation*4		0.31	—	RCR3.RTCDV[2:0] set to low drive capacity		
			1.09	—	RCR3.RTCDV[2:0] set to normal drive capacity		

- Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.
- Note 2. The IWDT and LVD are stopped.
- Note 3.  $V_{CC} = 3.3$  V.
- Note 4. Includes the oscillation circuit.



**Figure 42.4 Voltage Dependency in Software Standby Mode (Reference Data)**



**Figure 42.5 Temperature Dependency in Software Standby Mode (Reference Data)**

**Table 42.9 DC Characteristics (7)**

Conditions: Products with operating temperature ( $T_a$ )  $-40$  to  $+105^\circ\text{C}$   
 $V_{CC} = AV_{CC0} = V_{CC\_USB} = 1.8$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V

Item	Symbol	Typ.	Max.	Unit	Test Conditions
Permissible junction temperature	T <sub>j</sub>	—	120	°C	High-speed operating mode
		—	105		Middle-speed operating mode
		—	120		Low-speed operating mode

Note: Make sure that  $T_j < T_a + 0.1 \times \text{total power consumption (mW)}$ , where total power consumption =  $(V_{CC} - V_{OH}) \times \Sigma I_{OH} + V_{OL} \times \Sigma I_{OL} + I_{CCmax} \times V_{CC}$ .

**Table 42.10 DC Characteristics (8)**

Conditions: Products with operating temperature ( $T_a$ )  $-40$  to  $+85^\circ\text{C}$   
 $V_{CC} = AV_{CC0} = V_{CC\_USB} = 1.8$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V

Item	Symbol	Typ.	Max.	Unit	Test Conditions
Permissible junction temperature	T <sub>j</sub>	—	120	°C	High-speed operating mode
		—	105		Middle-speed operating mode
		—	120		Low-speed operating mode

Note: Make sure that  $T_j < T_a + 0.1 \times \text{total power consumption (mW)}$ , where total power consumption =  $(V_{CC} - V_{OH}) \times \Sigma I_{OH} + V_{OL} \times \Sigma I_{OL} + I_{CCmax} \times V_{CC}$ .

**Table 42.11 DC Characteristics (9)**Conditions:  $VCC = AVCC0 = VCC\_USB = 1.8$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item		Symbol	Min.	Typ.*7	Max.	Unit	Test Conditions
Analog power supply current	During A/D conversion (at high-speed conversion)	$I_{AVCC}$	—	0.7	1.2	mA	
	During D/A conversion (per channel)		—	0.4	0.8		
	Waiting for A/D and D/A conversion (all units)		—	—	0.4	$\mu\text{A}$	
Reference power supply current	During A/D conversion (at high-speed conversion)	$I_{REFH0}$	—	25	52	$\mu\text{A}$	
	Waiting for A/D conversion (all units)		—	—	60	nA	
	During D/A conversion	$I_{REFH}$	—	50	100	$\mu\text{A}$	
	Waiting for D/A conversion (all units)		—	—	100	nA	
Temperature sensor*6	—	$I_{TEMP}$	—	75	—	$\mu\text{A}$	
Comparator B operating current*6	Window mode	$I_{CMP}^{*4}$	—	12.5	—	$\mu\text{A}$	
	Comparator high-speed mode		—	6.5	—	$\mu\text{A}$	
	Comparator low-speed mode		—	1.7	—	$\mu\text{A}$	
LCD operating current*6	External resistance division method*8 $f_{LCD} = f_{SUB} = 128$ Hz, 1/3 bias, and 4-time slice	$I_{LCD1}^{*5}$	—	0.04	—	$\mu\text{A}$	
	Internal voltage boosting method (VLCD.VLCD = 04) $f_{LCD} = f_{SUB} = 128$ Hz, 1/3 bias, and 4-time slice	$I_{LCD2}^{*5}$	—	0.85	—	$\mu\text{A}$	
	Internal voltage boosting method (VLCD.VLCD = 12) $f_{LCD} = f_{SUB} = 128$ Hz, 1/3 bias, and 4-time slice	$I_{LCD2}^{*5}$	—	1.55	—	$\mu\text{A}$	
	Capacitor split method $f_{LCD} = f_{SUB} = 128$ Hz, 1/3 bias, and 4-time slice	$I_{LCD3}^{*5}$	—	0.20	—	$\mu\text{A}$	
USB operating current*3	During USB communication operation under the following settings and conditions <ul style="list-style-type: none"> <li>Host controller operation is set to full-speed mode</li> <li>Bulk OUT transfer (64 bytes) <math>\times</math> 1, bulk IN transfer (64 bytes) <math>\times</math> 1</li> <li>Connect peripheral devices via a 1-meter USB cable from the USB port.</li> </ul>	$I_{USBH}^{*1}$	—	4.3 (VCC) 0.9 (VCC_USB)*3	—	mA	
	During USB communication operation under the following settings and conditions <ul style="list-style-type: none"> <li>Function controller operation is set to full-speed mode</li> <li>Bulk OUT transfer (64 bytes) <math>\times</math> 1, bulk IN transfer (64 bytes) <math>\times</math> 1</li> <li>Connect the host device via a 1-meter USB cable from the USB port.</li> </ul>	$I_{USBF}^{*1}$	—	3.6 (VCC) 1.1 (VCC_USB)*3	—	mA	
	During suspended state under the following setting and conditions <ul style="list-style-type: none"> <li>Function controller operation is set to full-speed mode (pull up the USB0_DP pin)</li> <li>Software standby mode</li> <li>Connect the host device via a 1-meter USB cable from the USB port.</li> </ul>	$I_{SUSP}^{*2}$	—	0.35 (VCC) 170 (VCC_USB)*3	—	$\mu\text{A}$	

Note 1. Current consumed only by the USB module.

Note 2. Includes the current supplied from the pull-up resistor of the USB0\_DP pin to the pull-down resistor of the host device, in addition to the current consumed by this MCU during the suspended state.

Note 3. Current consumed by the power supplies (VCC and USB\_VCC).

Note 4. Current consumed only by the comparator B module.

Note 5. Current consumed only by the LCD module. Current when the LCD panel is not connected.

Note 6. Current consumed by the power supply (VCC).

Note 7. When  $VCC = AVCC0 = VCC\_USB = 3.3$  V.

Note 8. It does not include the current that flows through external divider resistors.

**Table 42.12 DC Characteristics (10)**

Conditions:  $VCC = AVCC0 = VCC\_USB = 1.8$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RAM standby voltage	$V_{RAM}$	1.8	—	3.6	V	

**Table 42.13 DC Characteristics (11)**

Conditions:  $VCC = AVCC0 = VCC\_USB = 0$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power-on VCC rising gradient	At normal startup*1	SrVCC	0.02	—	20	ms/V	
	During fast startup time*2		0.02	—	2		
	Voltage monitoring 1 reset enabled at startup *3, *4		0.02	—	—		

Note 1. When OFS1.(STUPLVD1REN, FASTSTUP) = 11b.

Note 2. When OFS1.(STUPLVD1REN, FASTSTUP) = 10b.

Note 3. When OFS1.STUPLVD1REN = 0.

Note 4. Turn on the power supply voltage according to the normal startup rising gradient because the register settings set by OFS1 are not read in boot mode.

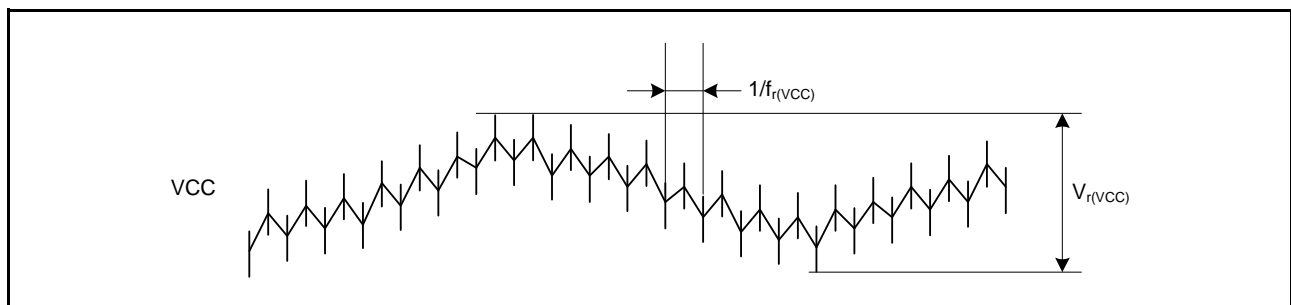
**Table 42.14 DC Characteristics (12)**

Conditions:  $VCC = AVCC0 = VCC\_USB = 1.8$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

The ripple voltage must meet the allowable ripple frequency  $f_{r(VCC)}$  within the range between the VCC upper limit (3.6 V) and lower limit (1.8 V).

When VCC change exceeds  $VCC \pm 10\%$ , the allowable voltage change rising/falling gradient  $dt/dVCC$  must be met.

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Allowable ripple frequency	$f_{r(VCC)}$	—	—	10	kHz	Figure 42.6 $V_{r(VCC)} \leq VCC \times 0.2$
		—	—	1	MHz	Figure 42.6 $V_{r(VCC)} \leq VCC \times 0.08$
		—	—	10	MHz	Figure 42.6 $V_{r(VCC)} \leq VCC \times 0.06$
Allowable voltage change rising/falling gradient	$dt/dVCC$	1.0	—	—	ms/V	When VCC change exceeds $VCC \pm 10\%$



**Figure 42.6 Ripple Waveform**

**Table 42.15 DC Characteristics (13)**Conditions:  $V_{CC} = AVCC0 = VCC\_USB = 1.8$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Permissible error of VCL pin external capacitance	$C_{VCL}$	1.4	4.7	7.0	$\mu\text{F}$	

Note: The recommended capacitance is  $4.7 \mu\text{F}$ . Variations in connected capacitors should be within the above range.**Table 42.16 Permissible Output Currents**Conditions:  $V_{CC} = AVCC0 = VCC\_USB = 1.8$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item	Symbol	Max.	Unit
Permissible output low current (average value per pin)	Ports 40 to 44, 46, ports J6, J7, ports 90 to 92	0.4	mA
	Ports other than above	8.0	
Permissible output low current (maximum value per pin)	Ports 40 to 44, 46, ports J6, J7, ports 90 to 92	0.4	mA
	Ports other than above	8.0	
Permissible output low current	Total of ports 40 to 44, 46, ports J6, J7, ports 90 to 92	$\Sigma I_{OL}$	2.4
	Total of ports 02, 04, 07, P20 to 27, P30, P31, PJ0, PJ2, PJ3		30
	Total of ports 10 to 17, ports 32, ports 50 to 56, ports C0 to C7, ports B0 to B7		30
	Total of ports E0 to E7, ports A0 to A7, ports F6, 7, ports D0 to D4		30
	Total of all output pins		60
Permissible output high current (average value per pin)	Ports 40 to 44, 46, ports J6, J7, ports 90 to 92	$I_{OH}$	-0.1
	Ports other than above		-4.0
Permissible output high current (maximum value per pin)	Ports 40 to 44, 46, ports J6, J7, ports 90 to 92		-0.1
	Ports other than above		-4.0
Permissible output high current	Total of ports 40 to 44, 46, ports J6, J7, ports 90 to 92	$\Sigma I_{OH}$	-0.6
	Total of ports 02, 04, 07, P20 to 27, P30, P31, PJ0, PJ2, PJ3		-10
	Total of ports 10 to 17, ports 32, ports 50 to 56, ports C0 to C7, ports B0 to B7		-15
	Total of ports E0 to E7, ports A0 to A7, ports F6, 7, ports D0 to D4		-15
	Total of all output pins		-40

Note: Do not exceed the permissible total supply current.

**Table 42.17 Output Values of Voltage (1)**Conditions:  $VCC = VCC\_USB = 2.7$  to  $3.6$  V,  $AVCC0 = 2.7$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item		Symbol	Min.	Max.	Unit	Test Conditions	
Output low	All output ports (except for RIIC, port 4, port 9, and port J)	$V_{OL}$	—	0.6	V	$I_{OL} = 3.0$ mA	
			—	0.4		$I_{OL} = 1.5$ mA	
	Ports 40 to 44, 46, ports J6, J7, ports 90 to 92		—	0.4		$I_{OL} = 0.4$ mA	
	RIIC pins		Standard mode	—		0.4	$I_{OL} = 3.0$ mA
			Fast mode	—		0.6	$I_{OL} = 6.0$ mA
Output high	All output ports (except for port 4, port 9, and port J)*1	$V_{OH}$	$VCC - 0.5$	—	V	$I_{OH} = -2.0$ mA	
	Ports 40 to 44, 46, ports J6, J7, ports 90 to 92		$AVCC0 - 0.5$	—		$I_{OH} = -0.1$ mA	

Note 1. There are restrictions on AVCC0 and VCC depending on the usage conditions for the 12-bit D/A converter and I/O ports. When using ports J0 and J2 multiplexed with DA0 and DA1 as general I/O ports, make sure that  $VCC \leq AVCC0$ .

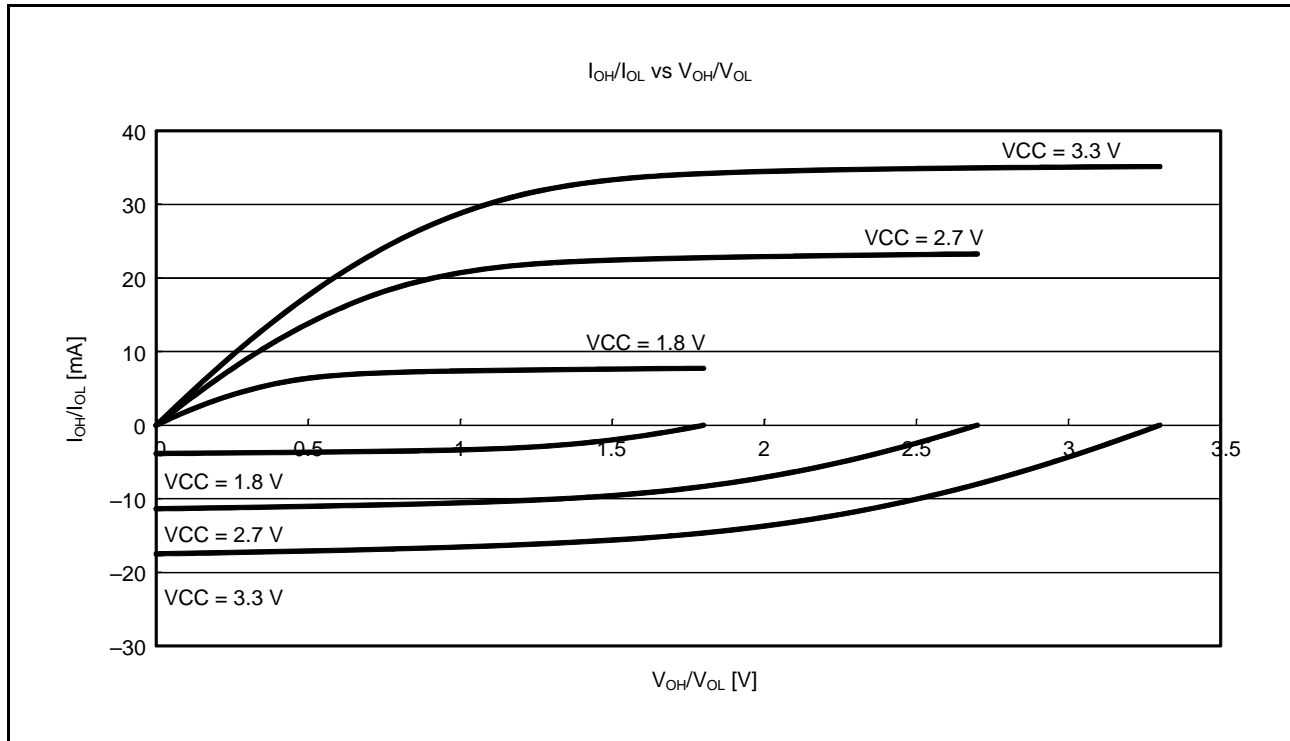
**Table 42.18 Output Values of Voltage (2)**Conditions:  $VCC = VCC\_USB = 1.8$  to  $2.7$  V,  $AVCC0 = 1.8$  to  $2.7$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item		Symbol	Min.	Max.	Unit	Test Conditions
Output low	All output ports (except for port 4, port 9, and port J)	$V_{OL}$	—	0.6	V	$I_{OL} = 1.5$ mA
	Ports 40 to 44, 46, ports J6, J7, ports 90 to 92		—	0.4		$I_{OL} = 0.4$ mA
Output high	All output ports (except for port 4, port 9, and port J)*1	$V_{OH}$	$VCC - 0.5$	—	V	$I_{OH} = -1.0$ mA
	Ports 40 to 44, 46, ports J6, J7, ports 90 to 92		$AVCC0 - 0.5$	—		$I_{OH} = -0.1$ mA

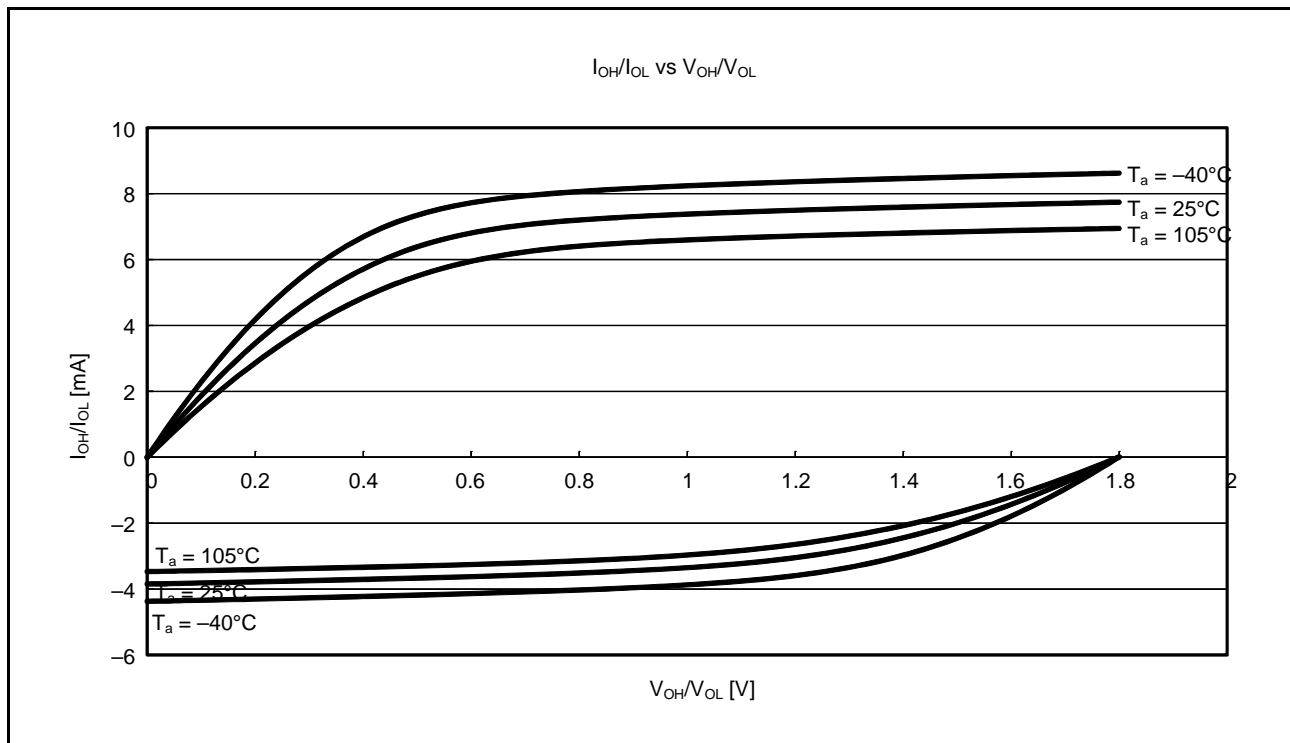
Note 1. There are restrictions on AVCC0 and VCC depending on the usage conditions for the 12-bit D/A converter and I/O ports. When using ports J0 and J2 multiplexed with DA0 and DA1 as general I/O ports, make sure that  $VCC \leq AVCC0$ .

### 42.2.1 Standard I/O Pin Output Characteristics (1)

Figure 42.7 to Figure 42.10 show the characteristics of general ports (except for the RIIC output pin, ports 4, J6, J7, and 9).

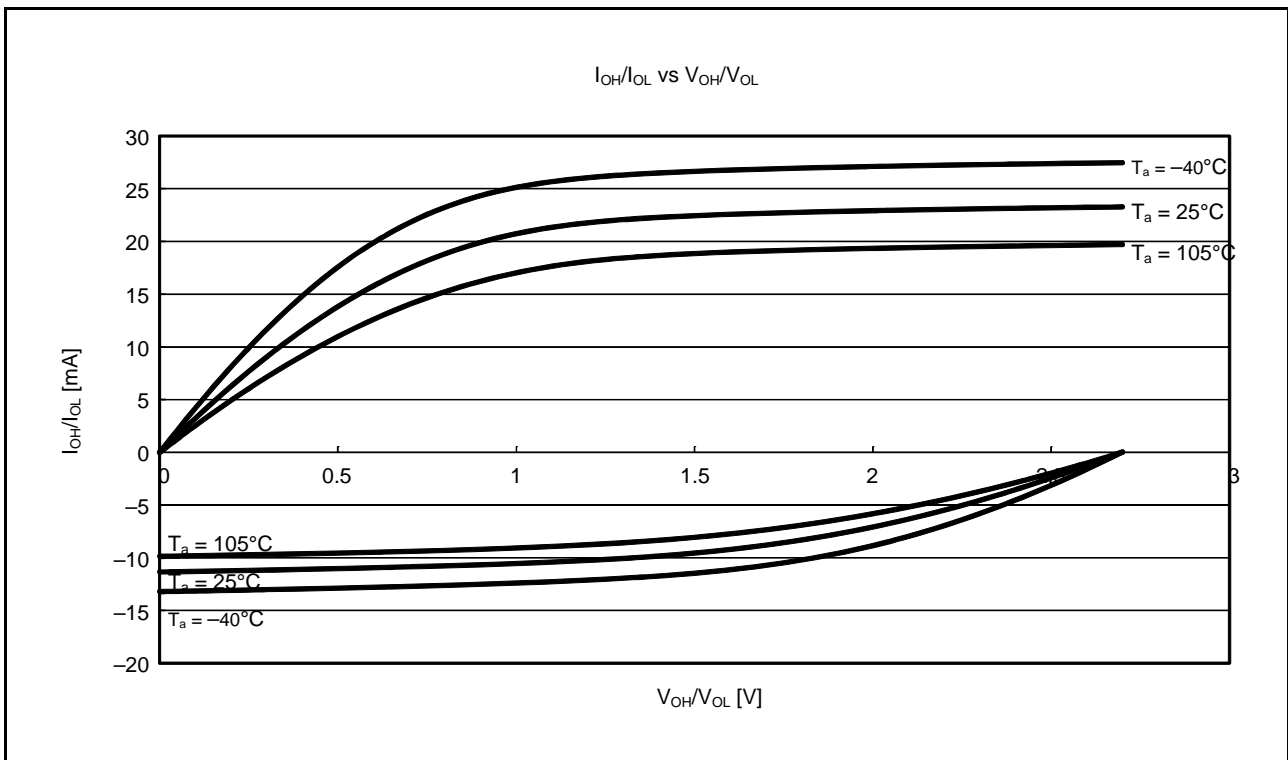


**Figure 42.7**  $V_{OH}/V_{OL}$  and  $I_{OH}/I_{OL}$  Voltage Characteristics of General Ports (Except for RIIC Output Pin, Ports 4, J6, J7, and 9) at  $T_a = 25^\circ\text{C}$  (Reference Data)

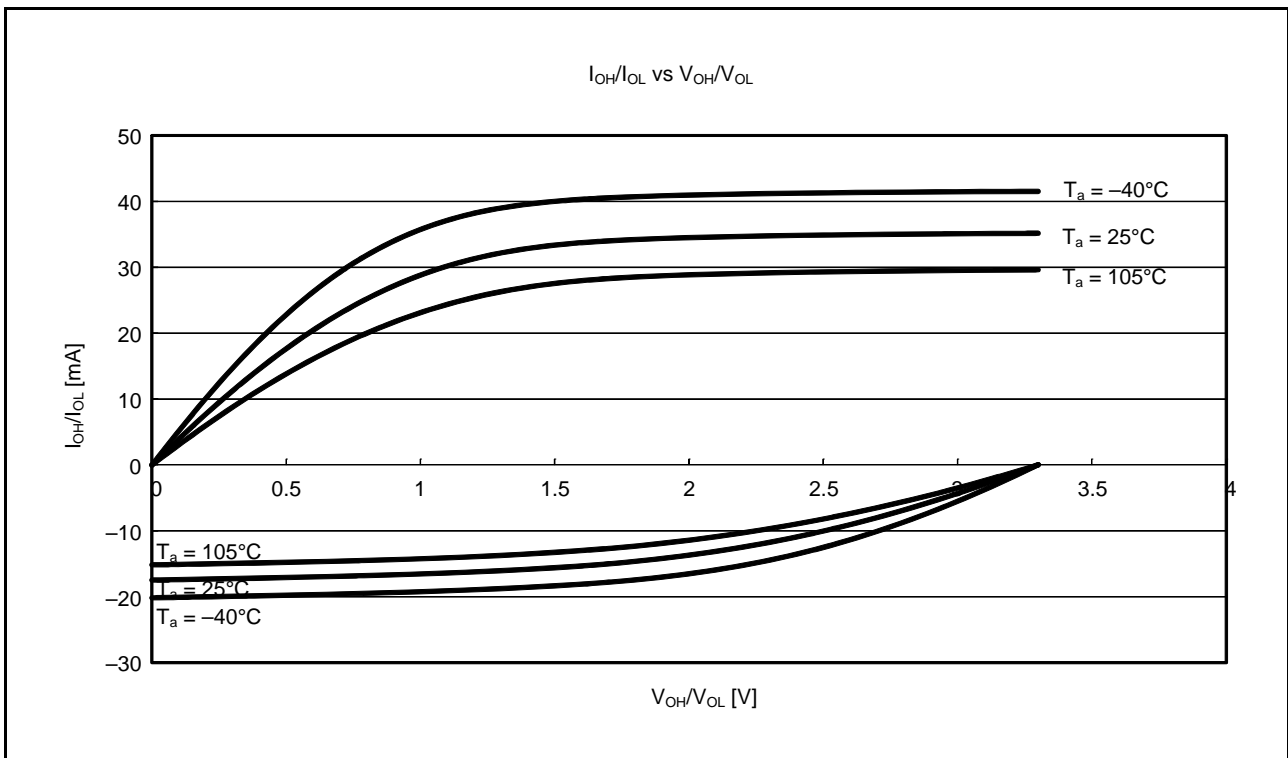


**Figure 42.8**  $V_{OH}/V_{OL}$  and  $I_{OH}/I_{OL}$  Temperature Characteristics of General Ports (Except for RIIC Output Pin, Ports 4, J6, J7, and 9) at  $V_{CC} = 1.8\text{ V}$  (Reference Data)





**Figure 42.9**  $V_{OH}/V_{OL}$  and  $I_{OH}/I_{OL}$  Temperature Characteristics of General Ports (Except for RIIC Output Pin, Ports 4, J6, J7, and 9) at  $V_{CC} = 2.7$  V (Reference Data)



**Figure 42.10**  $V_{OH}/V_{OL}$  and  $I_{OH}/I_{OL}$  Temperature Characteristics of General Ports (Except for RIIC Output Pin, Ports 4, J6, J7, and 9) at  $V_{CC} = 3.3$  V (Reference Data)

42.2.2 Standard I/O Pin Output Characteristics (2)

Figure 42.11 to Figure 42.13 show the characteristics of the RIIC output pin.

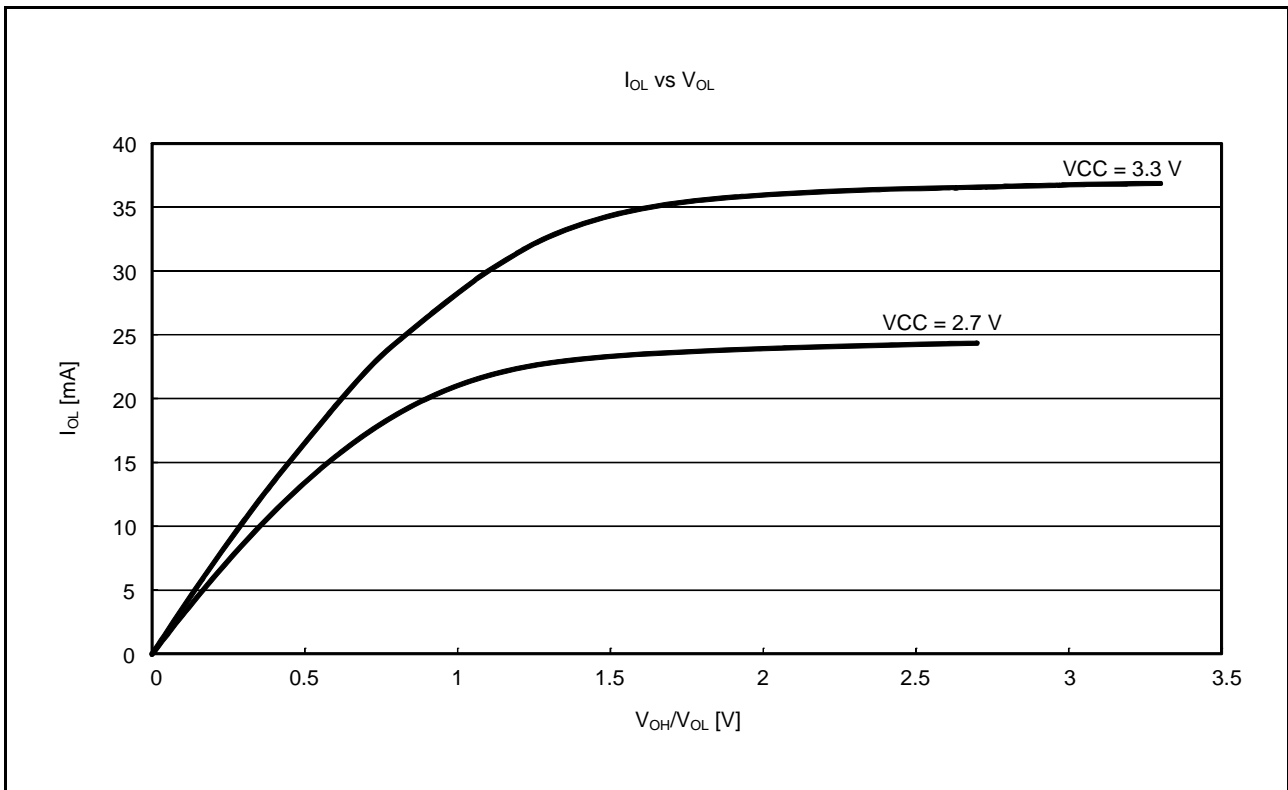


Figure 42.11 V<sub>OL</sub> and I<sub>OL</sub> Voltage Characteristics of RIIC Output Pin at T<sub>a</sub> = 25°C (Reference Data)

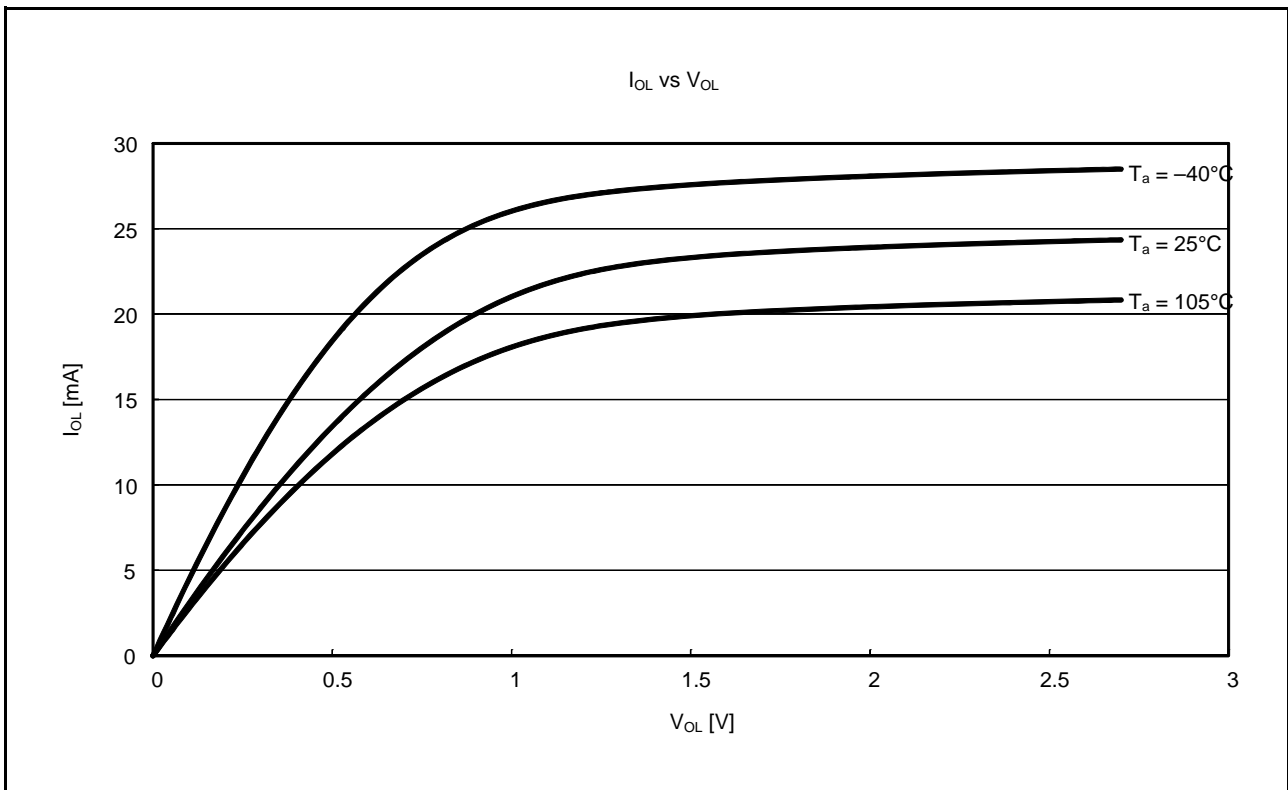


Figure 42.12 V<sub>OL</sub> and I<sub>OL</sub> Temperature Characteristics of RIIC Output Pin at VCC = 2.7 V (Reference Data)

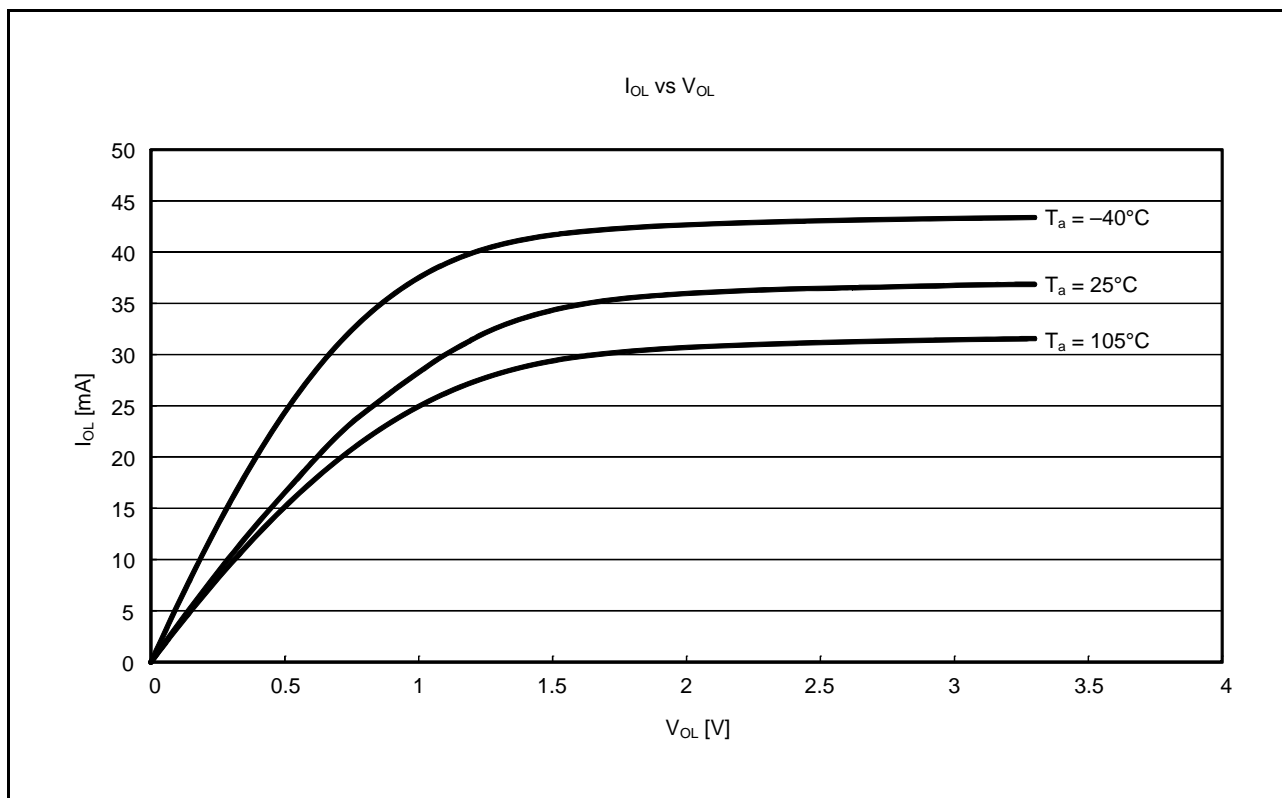


Figure 42.13 V<sub>OL</sub> and I<sub>OL</sub> Temperature Characteristics of RIIC Output Pin at VCC = 3.3 V (Reference Data)

42.2.3 Standard I/O Pin Output Characteristics (3)

Figure 42.14 to Figure 42.17 show the characteristics of ports 4, J6, J7, and 9.

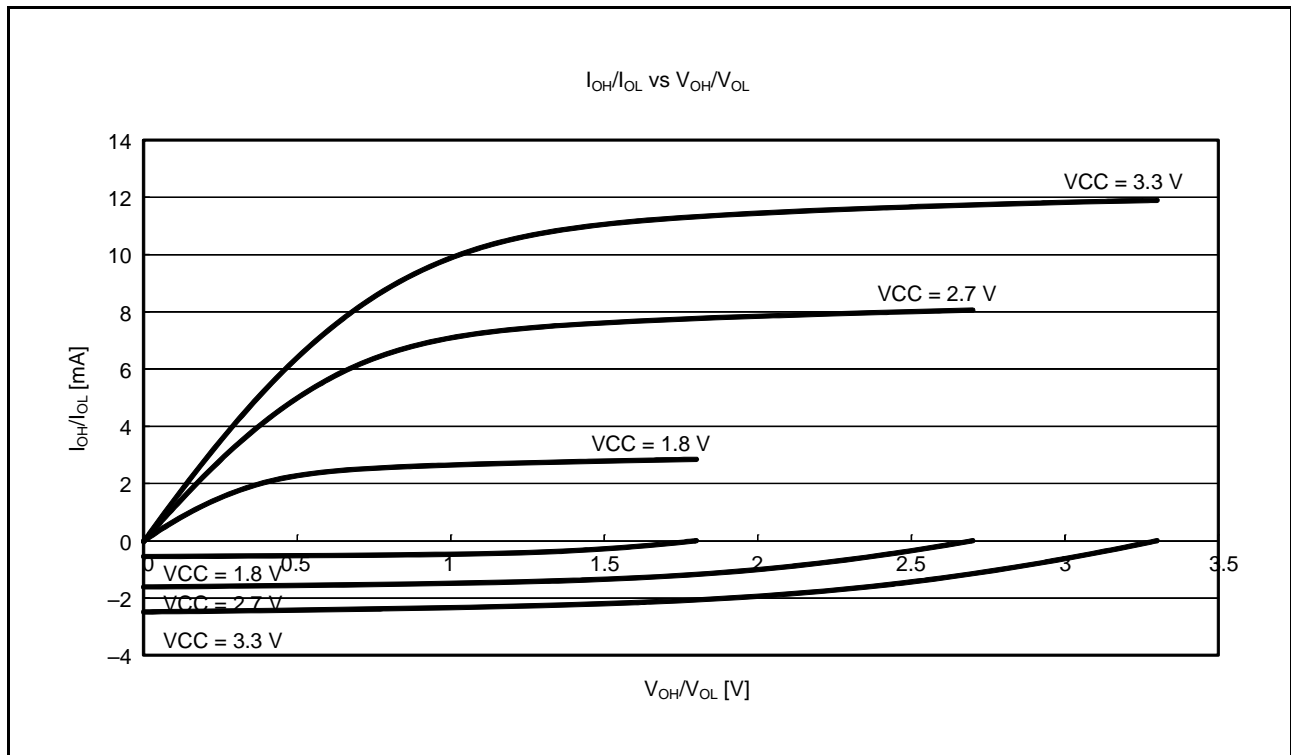


Figure 42.14  $V_{OH}/V_{OL}$  and  $I_{OH}/I_{OL}$  Voltage Characteristics of Ports 4, J6, J7, and 9 at  $T_a = 25^\circ\text{C}$  (Reference Data)

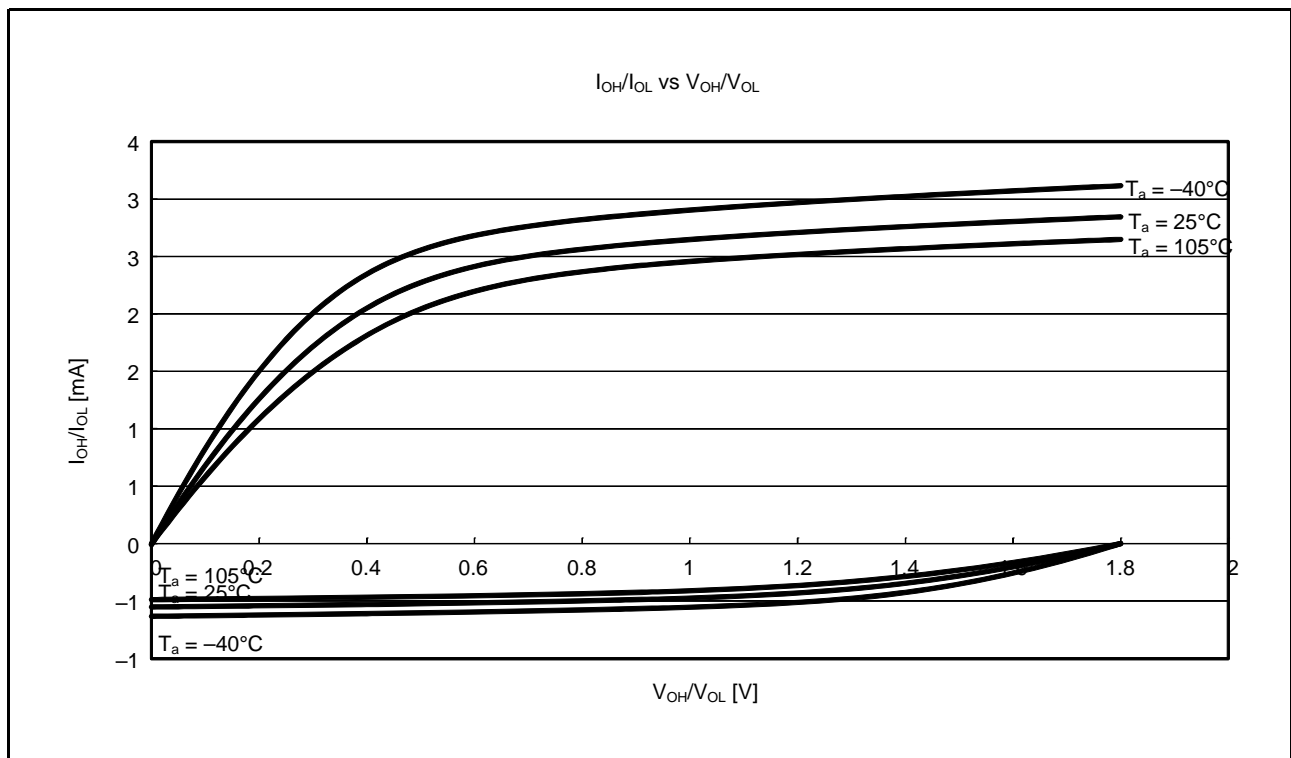


Figure 42.15  $V_{OH}/V_{OL}$  and  $I_{OH}/I_{OL}$  Temperature Characteristics of Ports 4, J6, J7, and 9 at  $V_{CC} = 1.8\text{ V}$  (Reference Data)

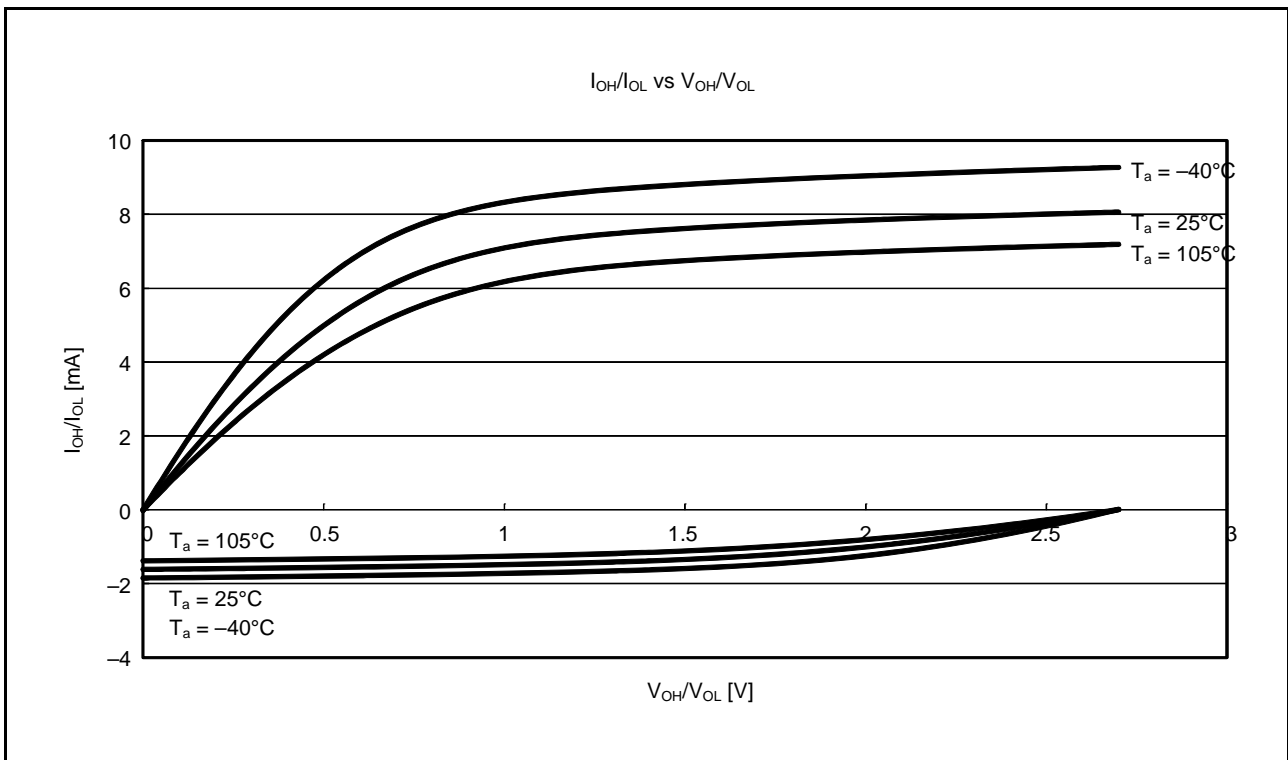


Figure 42.16  $V_{OH}/V_{OL}$  and  $I_{OH}/I_{OL}$  Temperature Characteristics of Ports 4, J6, J7, and 9 at  $V_{CC} = 2.7\text{ V}$  (Reference Data)

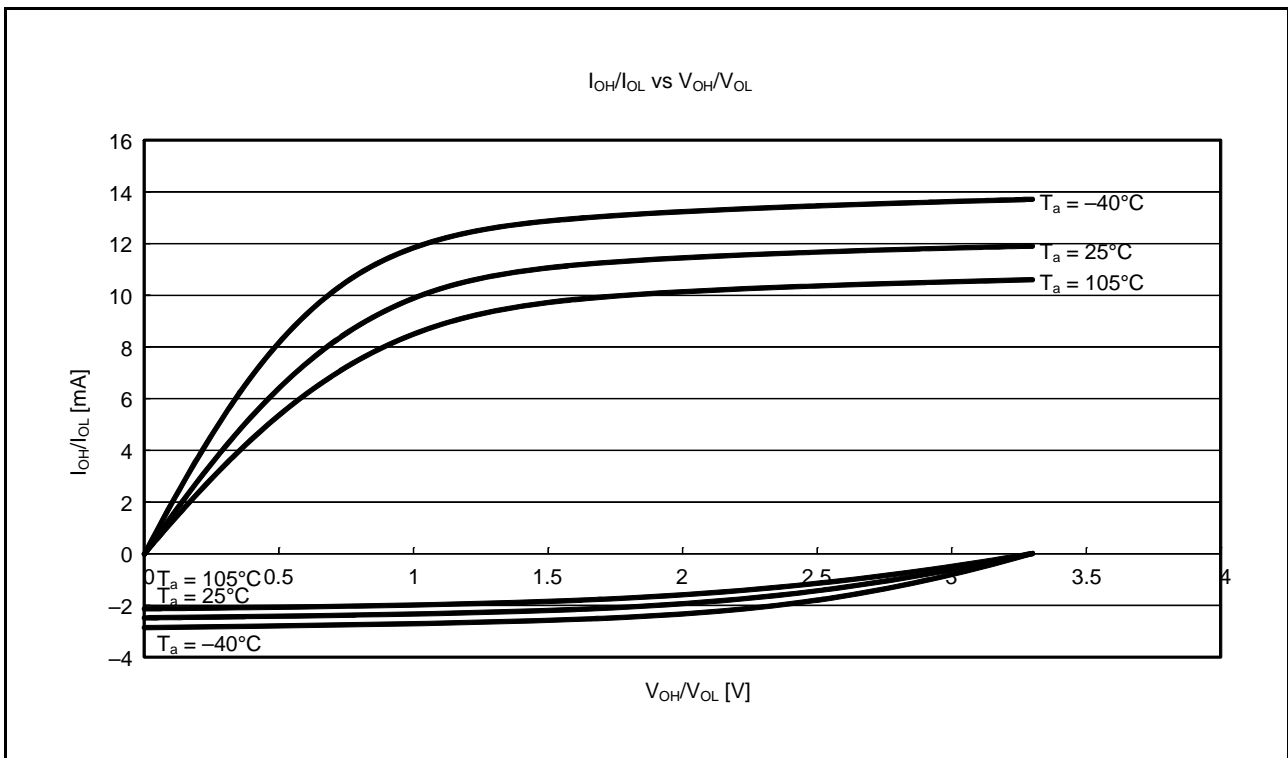


Figure 42.17  $V_{OH}/V_{OL}$  and  $I_{OH}/I_{OL}$  Temperature Characteristics of Ports 4, J6, J7, and 9 at  $V_{CC} = 3.3\text{ V}$  (Reference Data)

## 42.3 AC Characteristics

### 42.3.1 Clock Timing

**Table 42.19 Operation Frequency Value (High-Speed Operating Mode)**

Conditions: VCC = AVCC0 = VCC\_USB = 1.8 to 3.6 V, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

Item		Symbol	VCC				Unit
			1.8 to 2.4 V	2.4 to 2.7 V	2.7 to 3.6 V	When USB in Use*4	
Maximum operating frequency	System clock (ICLK)	f <sub>max</sub>	8	16	32	32	MHz
	FlashIF clock (FCLK)*1, *2		8	16	32	32	
	Peripheral module clock (PCLKB)		8	16	32	32	
	Peripheral module clock (PCLKD)*3		8	16	32	32	
	USB clock (UCLK)	f <sub>usb</sub>	—	—	—	48	

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.

Note 2. The frequency accuracy of FCLK should be ±3.5%. Confirm the frequency accuracy of the clock source.

Note 3. The lower-limit frequency of PCLKD is 4 MHz at 2.4 V or above and 1 MHz at below 2.4 V when the A/D converter is in use.

Note 4. The VCC\_USB range is 3.0 to 3.6 V when the USB clock is in use.

**Table 42.20 Operation Frequency Value (Middle-Speed Operating Mode)**

Conditions: VCC = AVCC0 = VCC\_USB = 1.8 to 3.6 V, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

Item		Symbol	VCC				Unit
			1.8 to 2.4 V	2.4 to 2.7 V	2.7 to 3.6 V	When USB in Use*4	
Maximum operating frequency	System clock (ICLK)	f <sub>max</sub>	8	12	12	12	MHz
	FlashIF clock (FCLK)*1, *2		8	12	12	12	
	Peripheral module clock (PCLKB)		8	12	12	12	
	Peripheral module clock (PCLKD)*3		8	12	12	12	
	USB clock (UCLK)	f <sub>usb</sub>	—	—	—	48	

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.

Note 2. The frequency accuracy of FCLK should be ±3.5%.

Note 3. The lower-limit frequency of PCLKD is 4 MHz at 2.4 V or above and 1 MHz at below 2.4 V when the A/D converter is in use.

Note 4. The VCC\_USB range is 3.0 to 3.6 V when the USB clock is in use.

**Table 42.21 Operation Frequency Value (Low-Speed Operating Mode)**

Conditions: VCC = AVCC0 = VCC\_USB = 1.8 to 3.6 V, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

Item		Symbol	VCC			Unit
			1.8 to 2.4 V	2.4 to 2.7 V	2.7 to 3.6 V	
Maximum operating frequency	System clock (ICLK)	f <sub>max</sub>	32.768			kHz
	FlashIF clock (FCLK)*1		32.768			
	Peripheral module clock (PCLKB)		32.768			
	Peripheral module clock (PCLKD)*2		32.768			

Note 1. Programming and erasing the flash memory is impossible.

Note 2. The A/D converter cannot be used.

**Table 42.22 Clock Timing**Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 1.8$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
XTAL external clock input cycle time	$t_{Xcyc}$	50	—	—	ns	Figure 42.18	
XTAL external clock input high pulse width	$t_{XH}$	20	—	—	ns		
XTAL external clock input low pulse width	$t_{XL}$	20	—	—	ns		
XTAL external clock rising time	$t_{Xr}$	—	—	5	ns		
XTAL external clock falling time	$t_{Xf}$	—	—	5	ns		
XTAL external clock input wait time*1	$t_{EXWT}$	0.5	—	—	$\mu\text{s}$	Figure 42.19	
Main clock oscillator oscillation frequency*2	$f_{MAIN}$	$2.4 \leq V_{CC} \leq 3.6$	1	—	20		MHz
		$1.8 \leq V_{CC} < 2.4$	1	—	8		
Main clock oscillator stabilization time (crystal)*2	$t_{MAINOSC}$	—	3	—	ms	Figure 42.19	
Main clock oscillator stabilization time (ceramic resonator)*2	$t_{MAINOSC}$	—	50	—	$\mu\text{s}$		
LOCO clock oscillation frequency	$f_{LOCO}$	3.44	4.0	4.56	MHz	Figure 42.20	
LOCO clock oscillation stabilization time	$t_{LOCO}$	—	—	0.5	$\mu\text{s}$		
IWDT-dedicated clock oscillation frequency	$f_{ILOCO}$	12.75	15	17.25	kHz	Figure 42.21	
IWDT-dedicated clock oscillation stabilization time	$t_{ILOCO}$	—	—	50	$\mu\text{s}$		
HOCO clock oscillation frequency	$f_{HOCO}$		31.52	32	32.48	MHz	$T_a = -40$ to $85^\circ\text{C}$
			31.68	32	32.32		$T_a = -20$ to $85^\circ\text{C}$
			31.36	32	32.64		$T_a = -40$ to $105^\circ\text{C}$
HOCO clock oscillation stabilization time	$t_{HOCO}$	—	—	56	$\mu\text{s}$	Figure 42.23	
PLL input frequency*3	$f_{PLLIN}$	4	—	8	MHz	Figure 42.24	
PLL circuit oscillation frequency*3	$f_{PLL}$	32	—	48	MHz		
PLL clock oscillation stabilization time	$t_{PLL}$	—	—	50	$\mu\text{s}$	Figure 42.24	
PLL free-running oscillation frequency	$f_{PLLFR}$	—	8	—	MHz		
USBPLL input frequency*5	$f_{PLLIN}$	—	6, 8*6	—	MHz	Figure 42.24	
USBPLL circuit oscillation frequency*5	$f_{PLL}$	—	48*6	—	MHz		
USBPLL clock oscillation stabilization time	$t_{PLL}$	—	—	50	$\mu\text{s}$	Figure 42.24	
Sub-clock oscillator oscillation frequency*7	$f_{SUB}$	—	32.768	—	kHz	Figure 42.25	
Sub-clock oscillator stabilization time*4	$t_{SUBOSC}$	—	0.5	—	s		

Note 1. Time until the clock can be used after the main clock oscillator stop bit (MOSCCR.MOSTP) is set to 0 (operating) when the external clock is stable.

Note 2. Reference values when an 8-MHz resonator is used.

When specifying the main clock oscillator stabilization time, set the MOSCWTCR register with a stabilization time value that is equal to or greater than the resonator-manufacturer-recommended value.

After changing the setting of the MOSCCR.MOSTP bit so that the main clock oscillator operates, read the OSCOVFSR.MOOVF flag to confirm that it has become 1, and then start using the main clock.

Note 3. The VCC range should be 2.4 to 3.6 V when the PLL is used.

Note 4. After changing the setting of the SOSCCR.SOSTP bit or RCR3.RTCEN bit so that the sub-clock oscillator operates, only start using the sub-clock after the sub-clock oscillation stabilization wait time that is equal to or greater than the oscillator-manufacturer-recommended value has elapsed.

Reference value when a 32.768-kHz resonator is used.

Note 5. The VCC range should be 3.0 to 3.6 V when the USBPLL is used.

Note 6. The input frequency can be set to 6 or 8 MHz only and the oscillation frequency can be set to 48 MHz only.

Note 7. Only 32.768 kHz can be used.

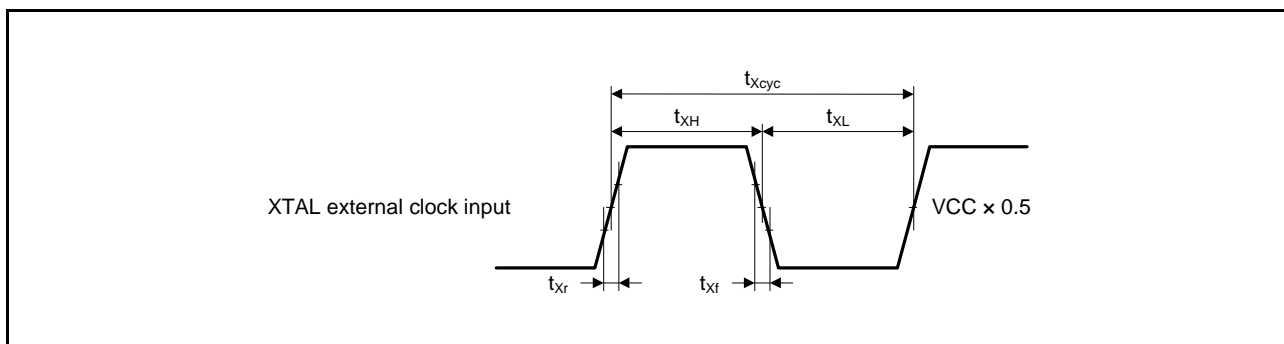


Figure 42.18 XTAL External Clock Input Timing

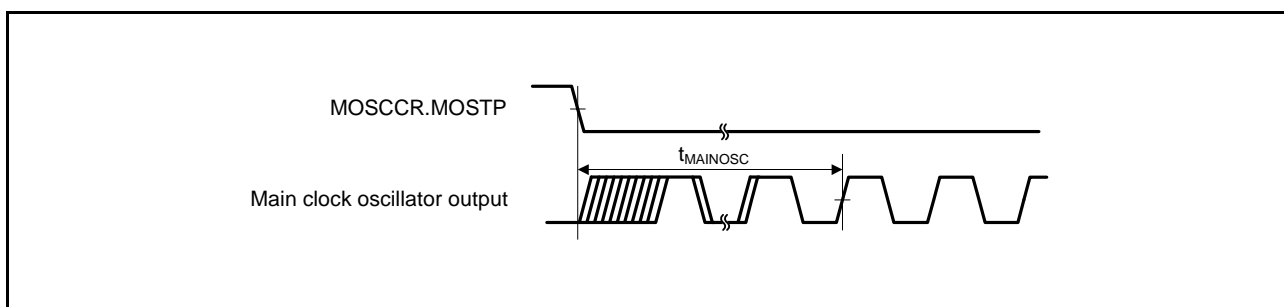


Figure 42.19 Main Clock Oscillation Start Timing

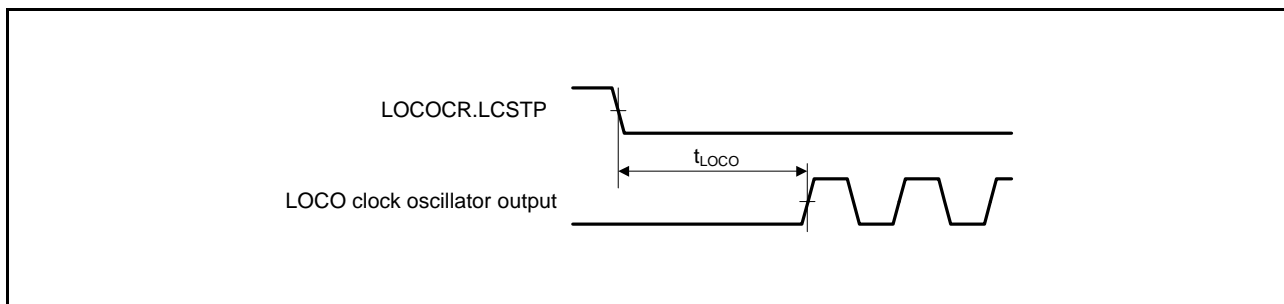


Figure 42.20 LOCO Clock Oscillation Start Timing

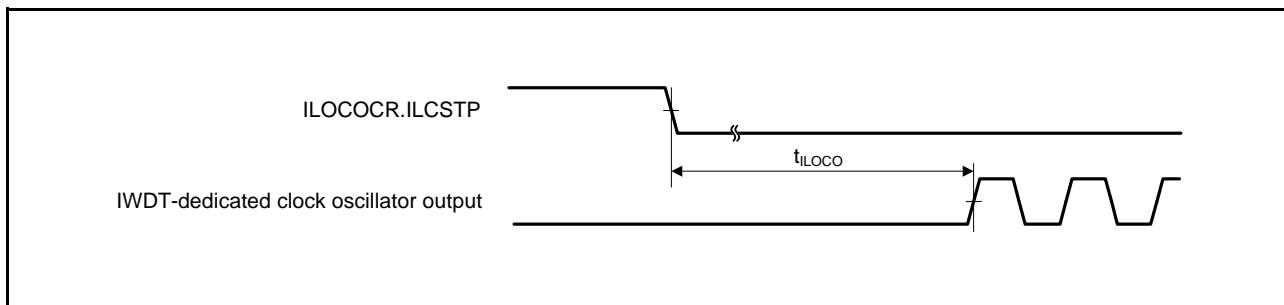
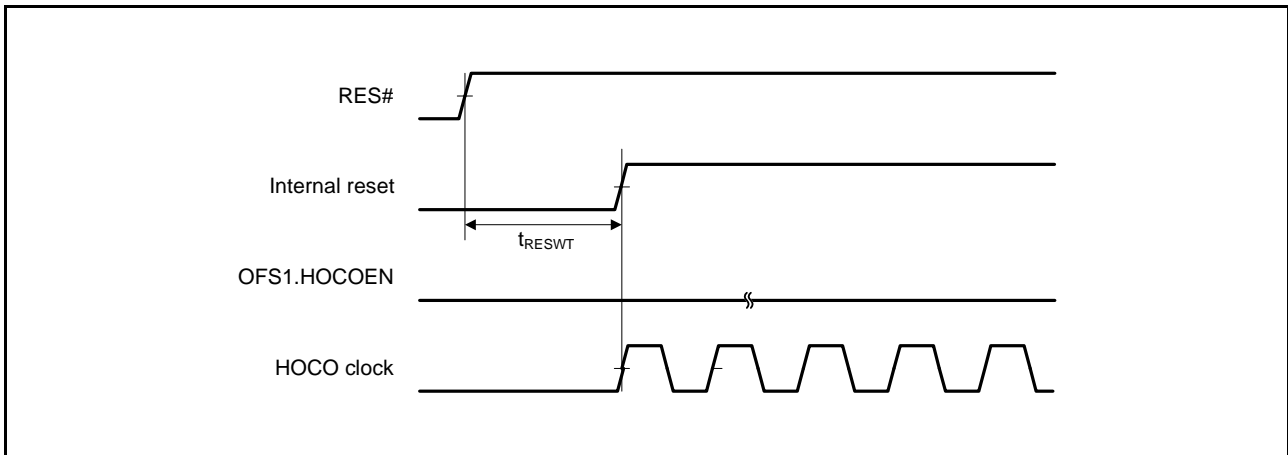
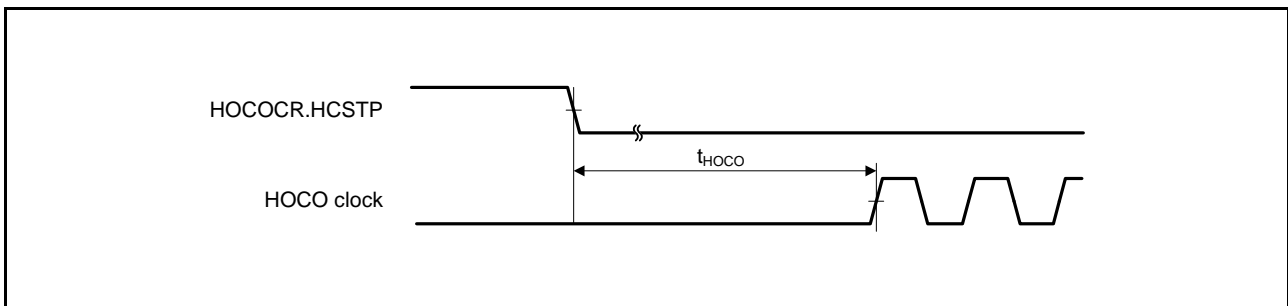


Figure 42.21 IWDT-Dedicated Clock Oscillation Start Timing

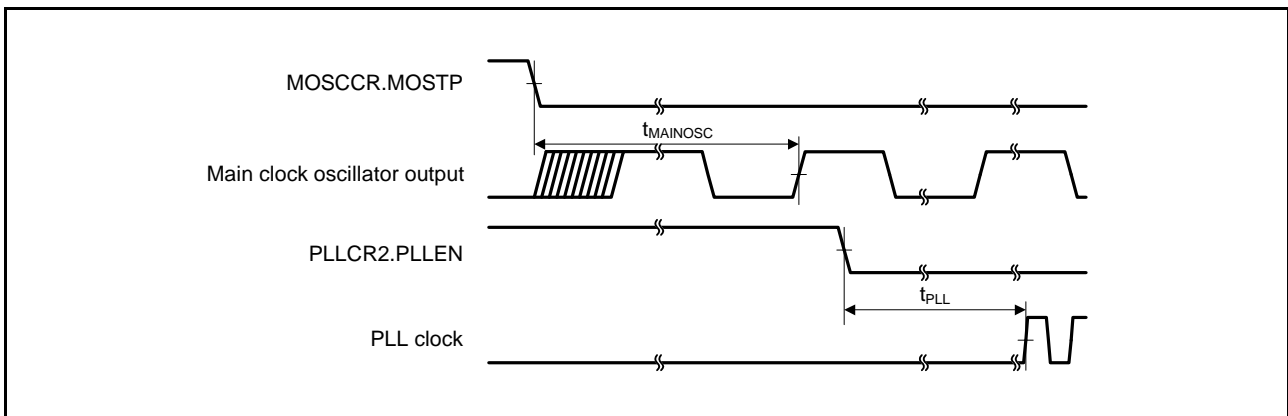




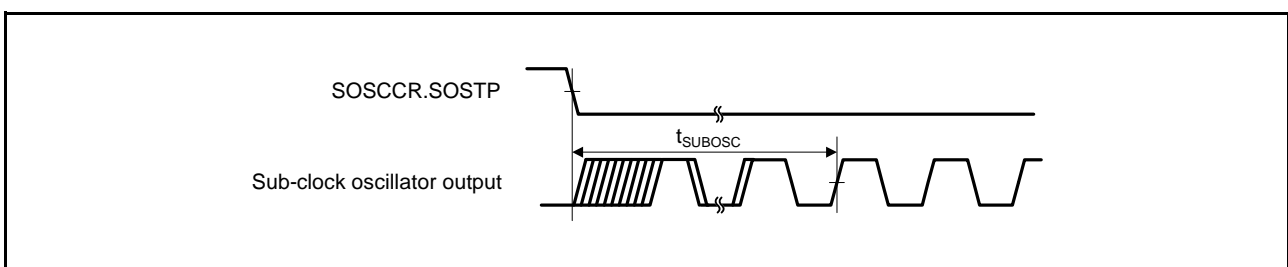
**Figure 42.22 HOCO Clock Oscillation Start Timing (After Reset is Canceled by Setting OFS1.HOCOEN Bit to 0)**



**Figure 42.23 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting HOCOCR.HCSTP Bit)**



**Figure 42.24 PLL Clock Oscillation Start Timing (PLL is Operated after Main Clock Oscillation Has Settled)**



**Figure 42.25 Sub-Clock Oscillation Start Timing**

### 42.3.2 Reset Timing

**Table 42.23 Reset Timing**

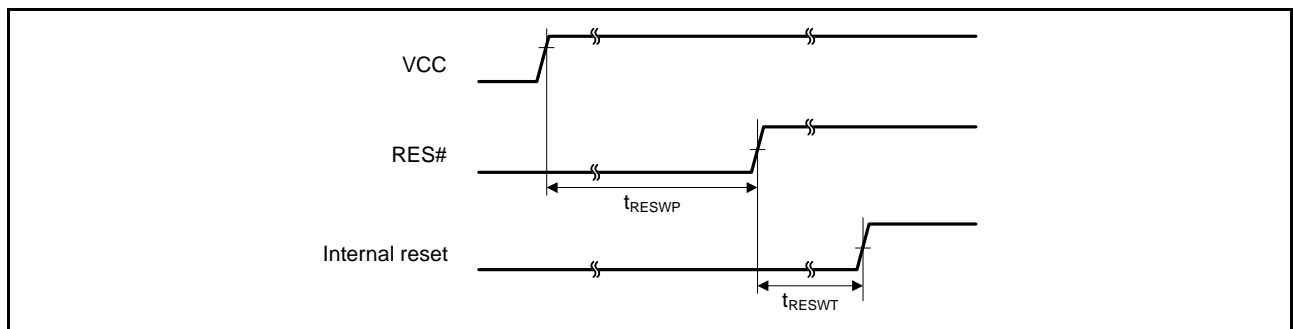
Conditions: VCC = AVCC0 = VCC\_USB = 1.8 to 3.6 V, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
RES# pulse width	At power-on	t <sub>RESWP</sub>	3	—	—	ms	Figure 42.26
	Other than above	t <sub>RESW</sub>	30	—	—	μs	Figure 42.27
Wait time after RES# cancellation (at power-on)	At normal startup*1	t <sub>RESWT</sub>	—	8.5	—	ms	Figure 42.26
	During fast startup time*2	t <sub>RESWT</sub>	—	560	—	μs	
Wait time after RES# cancellation (during powered-on state)	t <sub>RESWT</sub>	—	114	—	μs	Figure 42.27	
Independent watchdog timer reset period	t <sub>RESWIW</sub>	—	1	—	IWDT clock cycle	Figure 42.28	
Software reset period	t <sub>RESWSW</sub>	—	1	—	ICLK cycle		
Wait time after independent watchdog timer reset cancellation*3	t <sub>RESW2</sub>	—	300	—	μs		
Wait time after software reset cancellation	t <sub>RESW2</sub>	—	168	—	μs		

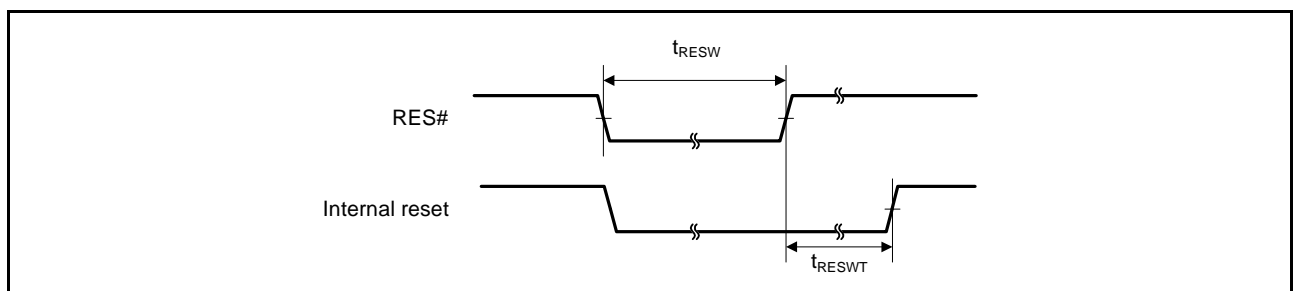
Note 1. When OFS1.(STUPLVD1REN, FASTSTUP) = 11b.

Note 2. When OFS1.(STUPLVD1REN, FASTSTUP) ≠ 11b.

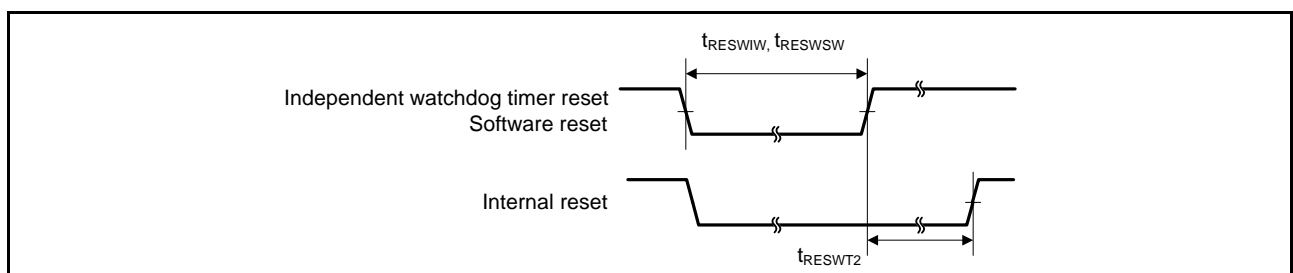
Note 3. When IWDTCR.CKS[3:0] = 0000b.



**Figure 42.26 Reset Input Timing at Power-On**



**Figure 42.27 Reset Input Timing (1)**



**Figure 42.28 Reset Input Timing (2)**

## 42.3.3 Timing of Recovery from Low Power Consumption Modes

**Table 42.24 Timing of Recovery from Low Power Consumption Modes (1)**Conditions: VCC = AVCC0 = VCC\_USB = 1.8 to 3.6 V, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

		Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time from software standby mode*1	High-speed mode	Crystal connected to main clock oscillator	Main clock oscillator operating*2	t <sub>SBYMC</sub>	—	2	3	ms	Figure 42.29
			Main clock oscillator and PLL circuit operating*3	t <sub>SBYPC</sub>	—	2	3	ms	
		External clock input to main clock oscillator	Main clock oscillator operating*4	t <sub>SBYEX</sub>	—	35	50	μs	
			Main clock oscillator and PLL circuit operating*5	t <sub>SBYPE</sub>	—	70	95	μs	
		Sub-clock oscillator operating		t <sub>SBYSC</sub>	—	650	800	μs	
		HOCO clock oscillator operating*6		t <sub>SBYHO</sub>	—	40	55	μs	
		LOCO clock oscillator operating		t <sub>SBYLO</sub>	—	40	55	μs	

Note 1. The recovery time varies depending on the state of each oscillator when the WAIT instruction is executed. The recovery time when multiple oscillators are operating varies depending on the operating state of the oscillators that are not selected as the system clock source. This applies when only the oscillator listed in each item is operating and the other oscillators are stopped.

Note 2. When the frequency of the crystal is 20 MHz.

When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.

Note 3. When the frequency of PLL is 32 MHz.

When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.

Note 4. When the frequency of the external clock is 20 MHz.

When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.

Note 5. When the frequency of PLL is 32 MHz.

When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.

Note 6. When the frequency of HOCO is 32 MHz.

When the high-speed clock oscillator wait control register (HOCOWTCR) is set to 05h.

**Table 42.25 Timing of Recovery from Low Power Consumption Modes (2)**Conditions: VCC = AVCC0 = VCC\_USB = 1.8 to 3.6 V, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

		Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time from software standby mode*1	Middle-speed mode	Crystal connected to main clock oscillator	Main clock oscillator operating*2	t <sub>SBYMC</sub>	—	2	3	ms	Figure 42.29
			Main clock oscillator and PLL circuit operating*3	t <sub>SBYPC</sub>	—	2	3	ms	
		External clock input to main clock oscillator	Main clock oscillator operating*4	t <sub>SBYEX</sub>	—	3	4	μs	
			Main clock oscillator and PLL circuit operating*5	t <sub>SBYPE</sub>	—	65	85	μs	
		Sub-clock oscillator operating		t <sub>SBYSC</sub>	—	600	750	μs	
		HOCO clock oscillator operating*6		t <sub>SBYHO</sub>	—	40	50	μs	
		LOCO clock oscillator operating		t <sub>SBYLO</sub>	—	4.8	7	μs	

Note 1. The recovery time varies depending on the state of each oscillator when the WAIT instruction is executed. The recovery time when multiple oscillators are operating varies depending on the operating state of the oscillators that are not selected as the system clock source. This applies when only the oscillator listed in each item is operating and the other oscillators are stopped.

Note 2. When the frequency of the crystal is 12 MHz.

When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.

Note 3. When the frequency of PLL is 12 MHz.

When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.

Note 4. When the frequency of the external clock is 12 MHz.

When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.

Note 5. When the frequency of PLL is 12 MHz.

When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.

Note 6. When the frequency of HOCO is 8 MHz.

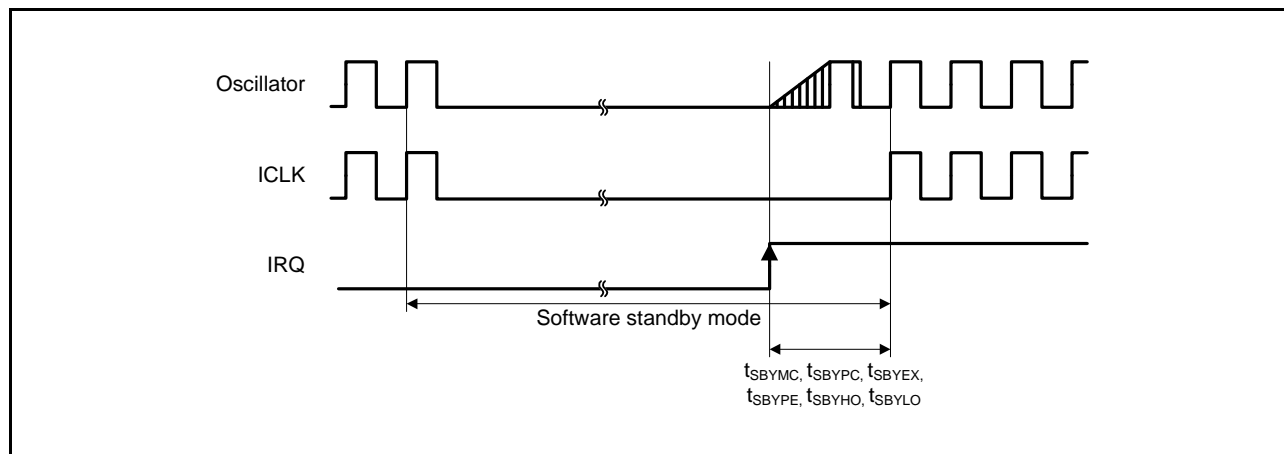
When the high-speed clock oscillator wait control register (HOCOWTCR) is set to 05h.

**Table 42.26 Timing of Recovery from Low Power Consumption Modes (3)**

Conditions:  $VCC = AVCC0 = VCC\_USB = 1.8$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time from software standby mode*1	Low-speed mode	Sub-clock oscillator operating	$t_{SBYSC}$	—	600	750	$\mu\text{s}$	Figure 42.29

Note 1. The sub-clock continues oscillating in software standby mode during low-speed mode.



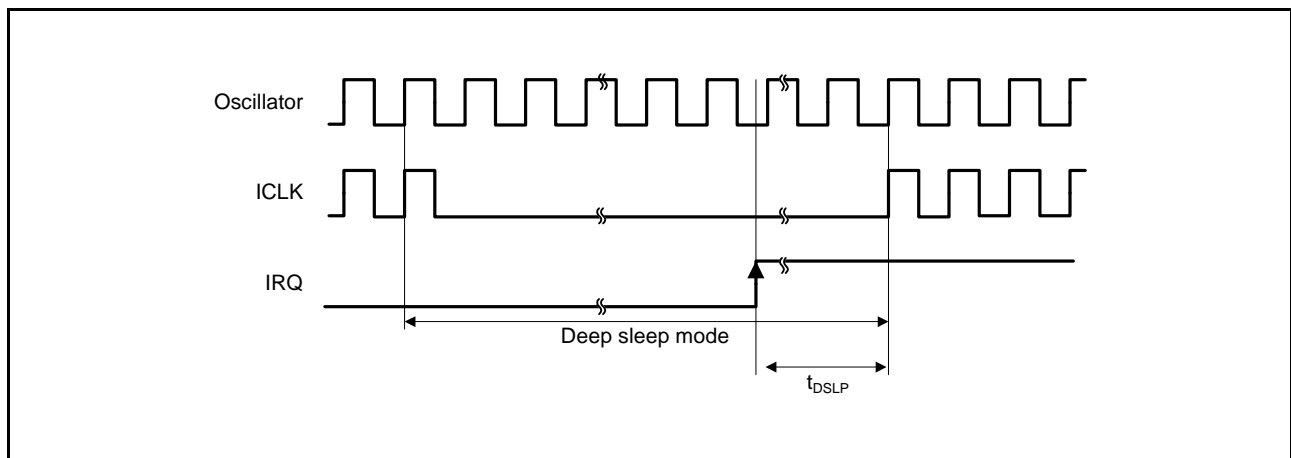
**Figure 42.29 Software Standby Mode Recovery Timing**

**Table 42.27 Timing of Recovery from Low Power Consumption Modes (4)**

Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 1.8$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time from deep sleep mode*1	High-speed mode*2	$t_{DSL P}$	—	2	3.5	$\mu\text{s}$
	Middle-speed mode*3	$t_{DSL P}$	—	3	4	$\mu\text{s}$
	Low-speed mode*4	$t_{DSL P}$	—	400	500	$\mu\text{s}$

- Note 1. Oscillators continue oscillating in deep sleep mode.
- Note 2. When the frequency of the system clock is 32 MHz.
- Note 3. When the frequency of the system clock is 12 MHz.
- Note 4. When the frequency of the system clock is 32.768 kHz.



**Figure 42.30 Deep Sleep Mode Recovery Timing**

**Table 42.28 Timing of Recovery from Low Power Consumption Modes (5) Operating Mode Transition Time**

Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 1.8$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

Mode before Transition	Mode after Transition	ICLK Frequency	Transition Time			Unit
			Min.	Typ.	Max.	
High-speed operating mode	Middle-speed operating mode	8 MHz	—	10	—	$\mu\text{s}$
Middle-speed operating mode	High-speed operating mode	8 MHz	—	37.5	—	$\mu\text{s}$
Low-speed operating mode	Middle-speed operating mode, high-speed operating mode	32.768 kHz	—	213.62	—	$\mu\text{s}$
Middle-speed operating mode, high-speed operating mode	Low-speed operating mode	32.768 kHz	—	183.11	—	$\mu\text{s}$

Note: When PCLKB, PCLKD, and FCLK are set to the same frequency division ratio as ICLK.

### 42.3.4 Control Signal Timing

**Table 42.29 Control Signal Timing**

Conditions: VCC = AVCC0 = VCC\_USB = 1.8 to 3.6 V, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

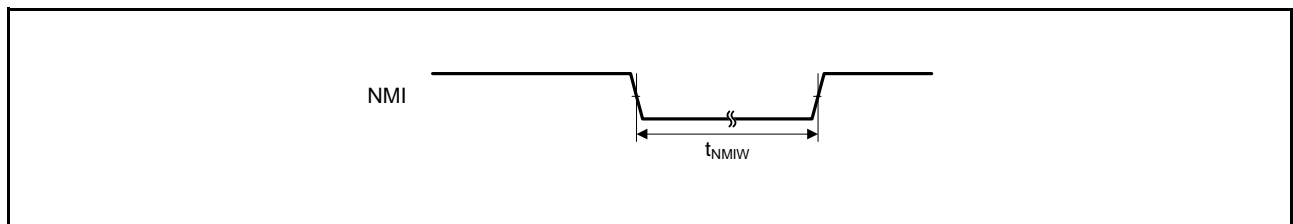
Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
NMI pulse width	t <sub>NMIW</sub>	200	—	—	ns	NMI digital filter disabled (NMIFLTE.NFLTEN = 0)	t <sub>Pcyc</sub> × 2 ≤ 200 ns
		t <sub>Pcyc</sub> × 2 <sup>*1</sup>	—	—			t <sub>Pcyc</sub> × 2 > 200 ns
		200	—	—		NMI digital filter enabled (NMIFLTE.NFLTEN = 1)	t <sub>NMICK</sub> × 3 ≤ 200 ns
		t <sub>NMICK</sub> × 3.5 <sup>*2</sup>	—	—			t <sub>NMICK</sub> × 3 > 200 ns
IRQ pulse width	t <sub>IRQW</sub>	200	—	—	ns	IRQ digital filter disabled (IRQFLTE0.FLTENi = 0)	t <sub>Pcyc</sub> × 2 ≤ 200 ns
		t <sub>Pcyc</sub> × 2 <sup>*1</sup>	—	—			t <sub>Pcyc</sub> × 2 > 200 ns
		200	—	—		IRQ digital filter enabled (IRQFLTE0.FLTENi = 1)	t <sub>IRQCK</sub> × 3 ≤ 200 ns
		t <sub>IRQCK</sub> × 3.5 <sup>*3</sup>	—	—			t <sub>IRQCK</sub> × 3 > 200 ns

Note: 200 ns minimum in software standby mode.

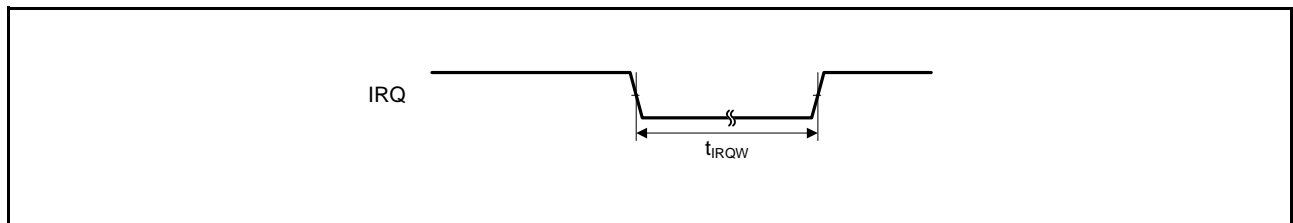
Note 1. t<sub>Pcyc</sub> indicates the cycle of PCLKB.

Note 2. t<sub>NMICK</sub> indicates the cycle of the NMI digital filter sampling clock.

Note 3. t<sub>IRQCK</sub> indicates the cycle of the IRQi digital filter sampling clock (i = 0 to 7).



**Figure 42.31 NMI Interrupt Input Timing**



**Figure 42.32 IRQ Interrupt Input Timing**

## 42.3.5 Timing of On-Chip Peripheral Modules

**Table 42.30 Timing of On-Chip Peripheral Modules (1)**Conditions: VCC = AVCC0 = VCC\_USB = 1.8 to 3.6 V, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

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 42.33		
MTU2	Input capture input pulse width	Single-edge setting	t <sub>TICW</sub>	1.5	—	t <sub>Pcyc</sub>	Figure 42.34	
		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 42.35	
	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 42.36		
TMR	Timer clock pulse width	Asynchronous	t <sub>TMCWH</sub> , t <sub>TMCWL</sub>	1.5	—	t <sub>Pcyc</sub>	Figure 42.37	
		Clock synchronous		2.5	—			
SCI	Input clock cycle	Asynchronous	t <sub>Scyc</sub>	4	—	t <sub>Pcyc</sub>	Figure 42.38 C = 30 pF	
		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 (master)	Clock synchronous		t <sub>TXD</sub>	—	40		ns
					—	65		ns
	Transmit data delay time (slave)	Clock synchronous	2.7 V or above	t <sub>TXD</sub>	—	65		ns
			1.8 V or above		—	100		ns
	Receive data setup time (master)	Clock synchronous	2.7 V or above	t <sub>RXS</sub>	65	—		ns
1.8 V or above			90		—	ns		
Receive data setup time (slave)	Clock synchronous		t <sub>RXS</sub>	40	—	ns		
Receive data hold time	Clock synchronous		t <sub>RXH</sub>	40	—	ns		
A/D converter	Trigger input pulse width	t <sub>TRGW</sub>	1.5	—	t <sub>Pcyc</sub>	Figure 42.40		
CAC	CACREF input pulse width	t <sub>Pcyc</sub> ≤ t <sub>cac</sub> <sup>*2</sup>	t <sub>CACREF</sub>	4.5 t <sub>cac</sub> + 3 t <sub>Pcyc</sub>	—	ns		
		t <sub>Pcyc</sub> > t <sub>cac</sub> <sup>*2</sup>		5 t <sub>cac</sub> + 6.5 t <sub>Pcyc</sub>				
CLKOUT	CLKOUT pin output cycle*4	VCC = 2.7 V or above	t <sub>Ccyc</sub>	125	—	ns		
		VCC = 1.8 V or above		250				
	CLKOUT pin high pulse width*3	VCC = 2.7 V or above	t <sub>CH</sub>	35	—	ns		
		VCC = 1.8 V or above		70				
	CLKOUT pin low pulse width*3	VCC = 2.7 V or above	t <sub>CL</sub>	35	—	ns		
		VCC = 1.8 V or above		70				
	CLKOUT pin output rise time	VCC = 2.7 V or above	t <sub>Cr</sub>	—	15	ns		
		VCC = 1.8 V or above		30				
CLKOUT pin output fall time	VCC = 2.7 V or above	t <sub>Cf</sub>	—	15	ns			
	VCC = 1.8 V or above		30					

Note 1. t<sub>Pcyc</sub>: PCLK cycleNote 2. t<sub>cac</sub>: CAC count clock source cycle

Note 3. When the LOCO is selected as the clock output source (CKOCR.CKOSSEL[2:0] bits = 000b), set the clock output division ratio selection to divided by 2 (CKOCR.CKODIV[2:0] bits = 001b).

Note 4. When the XTAL external clock input or an oscillator is used with divided by 1 (CKOCR.CKOSSEL[2:0] bits = 010b and CKOCR.CKODIV[2:0] bits = 000b) to output from CLKOUT, the above should be satisfied with an input duty cycle of 45 to 55%.

**Table 42.31 Timing of On-Chip Peripheral Modules (2)**Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 1.8$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ ,  $C = 30$  pF

Item			Symbol	Min.	Max.	Unit	Test Conditions		
RSPi	RSPCK clock cycle	Master	$t_{SPCyc}$	2	4096	$t_{Pcyc}^{*1}$	Figure 42.42		
		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	2.7 V or above	$t_{SPCKr}$	—	10		ns	
			1.8 V or above		$t_{SPCKf}$	—			15
		Input		—	1	$\mu\text{s}$			
	Data input setup time	Master	2.7 V or above	$t_{SU}$	10	—		ns	Figure 42.43 to Figure 42.48
			1.8 V or above		30	—			
		Slave			$25 - t_{Pcyc}$	—			
Data input hold time	Master	RSPCK set to a division ratio other than PCLKB divided by 2	$t_H$	$t_{Pcyc}$	—	ns			
		RSPCK set to PCLKB divided by 2	$t_{HF}$	0	—				
	Slave		$t_H$	$20 + 2 \times t_{Pcyc}$	—				
SSL setup time	Master		$t_{LEAD}$	$-30 + N^2 \times t_{SPCyc}$	—	ns			
	Slave			2	—		$t_{Pcyc}$		
SSL hold time	Master		$t_{LAG}$	$-30 + N^3 \times t_{SPCyc}$	—	ns			
	Slave			2	—		$t_{Pcyc}$		
Data output delay time	Master	2.7 V or above	$t_{OD}$	—	14	ns			
		1.8 V or above		—	30				
	Slave	2.7 V or above		—	$3 \times t_{Pcyc} + 65$				
		1.8 V or above		—	$3 \times t_{Pcyc} + 105$				
Data output hold time	Master	2.7 V or above	$t_{OH}$	0	—	ns			
		1.8 V or above		-20	—				
	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	2.7 V or above	$t_{Dr}, t_{Df}$	—	10	ns			
		1.8 V or above		—	20				
	Input			—	1		$\mu\text{s}$		
SSL rise/fall time	Output		$t_{SSLr}, t_{SSLf}$	—	20	ns			
	Input			—	1		$\mu\text{s}$		
Slave access time	2.7 V or above		$t_{SA}$	—	6	$t_{Pcyc}$	Figure 42.47, Figure 42.48		
	1.8 V or above			—	7				
Slave output release time	2.7 V or above		$t_{REL}$	—	5	$t_{Pcyc}$			
	1.8 V or above			—	6				

Note 1.  $t_{Pcyc}$ : PCLK cycle

Note 2. N: An integer from 1 to 8 that can be set by the RSPi clock delay register (SPCKD)

Note 3. N: An integer from 1 to 8 that can be set by the RSPi slave select negation delay register (SSLND)



**Table 42.32 Timing of On-Chip Peripheral Modules (3)**Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 1.8$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ ,  $C = 30$  pF

Item		Symbol	Min.	Max.	Unit*1	Test Conditions	
Simple SPI	SCK clock cycle output (master)	$t_{SPCyc}$	4	65536	$t_{PCyc}$	Figure 42.42	
	SCK clock cycle input (slave)		6	65536	$t_{PCyc}$		
	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 (master)	2.7 V or above	$t_{SU}$	65	—	ns	Figure 42.43, Figure 42.45
		1.8 V or above		95	—		
	Data input setup time (slave)	40		—			
	Data input hold time	$t_H$	40	—	ns		
	SS input setup time	$t_{LEAD}$	3	—	$t_{SPCyc}$		
	SS input hold time	$t_{LAG}$	3	—	$t_{SPCyc}$		
	Data output delay time (master)	$t_{OD}$	—	40	ns		
	Data output delay time (slave)		2.7 V or above	—		65	
			1.8 V or above	—		100	
	Data output hold time (master)	2.7 V or above	$t_{OH}$	-10	—	ns	
		1.8 V or above		-20	—		
	Data output hold time (slave)	—		-10	—		
	Data rise/fall time	$t_{Dr}, t_{Df}$	—	20	ns		
	SS input rise/fall time	$t_{SSLr}, t_{SSLf}$	—	20	ns		
Slave access time	$t_{SA}$	—	6	$t_{PCyc}$	Figure 42.47, Figure 42.48		
Slave output release time	$t_{REL}$	—	6	$t_{PCyc}$			

Note 1.  $t_{PCyc}$ : PCLK cycle

**Table 42.33 Timing of On-Chip Peripheral Modules (4)**

Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 2.7$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V,  $f_{PCLKB} \leq 32$  MHz,  
 $T_a = -40$  to  $+105^\circ\text{C}$

Item	Symbol	Min.*1	Max.	Unit	Test Conditions	
RIIC (Standard mode, SMBus)	SCL0 input cycle time	$t_{SCL}$	$6 (12) \times t_{IICcyc} + 1300$	—	ns	Figure 42.49
	SCL0 input high pulse width	$t_{SCLH}$	$3 (6) \times t_{IICcyc} + 300$	—	ns	
	SCL0 input low pulse width	$t_{SCLL}$	$3 (6) \times t_{IICcyc} + 300$	—	ns	
	SCL0, SDA0 input rise time	$t_{Sr}$	—	1000	ns	
	SCL0, SDA0 input fall time	$t_{Sf}$	—	300	ns	
	SCL0, SDA0 input spike pulse removal time	$t_{SP}$	0	$1 (4) \times t_{IICcyc}$	ns	
	SDA0 input bus free time	$t_{BUF}$	$3 (6) \times t_{IICcyc} + 300$	—	ns	
	START condition input hold time	$t_{STAH}$	$t_{IICcyc} + 300$	—	ns	
	Repeated START 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	
	SCL0, SDA0 capacitive load	$C_b$	—	400	pF	
RIIC (Fast mode)	SCL0 input cycle time	$t_{SCL}$	$6 (12) \times t_{IICcyc} + 600$	—	ns	Figure 42.49
	SCL0 input high pulse width	$t_{SCLH}$	$3 (6) \times t_{IICcyc} + 300$	—	ns	
	SCL0 input low pulse width	$t_{SCLL}$	$3 (6) \times t_{IICcyc} + 300$	—	ns	
	SCL0, SDA0 input rise time	$t_{Sr}$	—*2	300	ns	
	SCL0, SDA0 input fall time	$t_{Sf}$	—*2	300	ns	
	SCL0, SDA0 input spike pulse removal time	$t_{SP}$	0	$1 (4) \times t_{IICcyc}$	ns	
	SDA0 input bus free time	$t_{BUF}$	$3 (6) \times t_{IICcyc} + 300$	—	ns	
	START condition input hold time	$t_{STAH}$	$t_{IICcyc} + 300$	—	ns	
	Repeated START 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	
	SCL0, SDA0 capacitive load	$C_b$	—	400	pF	

Note:  $t_{IICcyc}$ : RIIC internal reference count clock (IIC $\phi$ ) cycle

Note 1. The value in parentheses is used when the ICMR3.NF[1:0] bits are set to 11b while a digital filter is enabled with the ICFER.NFE bit = 1.

Note 2. The minimum  $t_{sr}$  and  $t_{sf}$  specifications for fast mode are not set.

**Table 42.34 Timing of On-Chip Peripheral Modules (5)**

Conditions:  $VCC = AVCC0 = VCC\_USB = 2.7$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $fPCLKB \leq 32$  MHz,  
 $T_a = -40$  to  $+105^\circ\text{C}$

Item		Symbol	Min.	Max.	Unit	Test Conditions
Simple I <sup>2</sup> C (Standard mode)	SDA0 input rise time	$t_{Sr}$	—	1000	ns	Figure 42.49
	SDA0 input fall time	$t_{Sf}$	—	300	ns	
	SDA0 input spike pulse removal time	$t_{SP}$	0	$4 \times t_{Pcyc}^{*1}$	ns	
	Data input setup time	$t_{SDAS}$	250	—	ns	
	Data input hold time	$t_{SDAH}$	0	—	ns	
	SCL0, SDA0 capacitive load	$C_b$	—	400	pF	
Simple I <sup>2</sup> C (Fast mode)	SCL0, SDA0 input rise time	$t_{Sr}$	—	300	ns	Figure 42.49
	SCL0, SDA0 input fall time	$t_{Sf}$	—	300	ns	
	SCL0, SDA0 input spike pulse removal time	$t_{SP}$	0	$4 \times t_{Pcyc}^{*1}$	ns	
	Data input setup time	$t_{SDAS}$	100	—	ns	
	Data input hold time	$t_{SDAH}$	0	—	ns	
	SCL0, SDA0 capacitive load	$C_b$	—	400	pF	

Note:  $t_{Pcyc}$ : PCLK cycle

Note 1. This applies when the SMR.CKS[1:0] bits = 00b and the SNFR.NFCS[2:0] bits = 010b while the SNFR.NFE bit = 1 and the digital filter is enabled.

**Table 42.35 Timing of On-Chip Peripheral Modules (6)**

Conditions:  $VCC = AVCC0 = VCC\_USB = 1.8$  to  $3.6$  V,  $VSS = AVSS0 = VSS\_USB = 0$  V,  $fPCLKB \leq 32$  MHz,  
 $T_a = -40$  to  $+105^\circ\text{C}$

Item		Symbol	Min.	Max.	Unit	Test Conditions	
SSI	AUDIO_MCLK input frequency	$t_{AUDIO}$	2.7 V or above	1	25	MHz	
			1.8 V or above	1	4		
Output clock cycle		$t_o$	250	—	ns	Figure 42.50	
Input clock cycle		$t_i$	250	—	ns		
Clock high pulse width		$t_{HC}$	0.4	0.6	$t_o, t_i$		
Clock low pulse width		$t_{LC}$	0.4	0.6	$t_o, t_i$		
Clock rise time		$t_{RC}$	—	20	ns		
Data delay time		$t_{DTR}$	2.7 V or above	—	65		ns
			1.8 V or above	—	105		
Setup time	2.7 V or above	$t_{SR}$	65	—	ns		
	1.8 V or above		90	—			
Hold time		$t_{HTR}$	40	—	ns		
WS changing edge SSIDATA output delay		$t_{DTRW}$	—	105	ns	Figure 42.53	

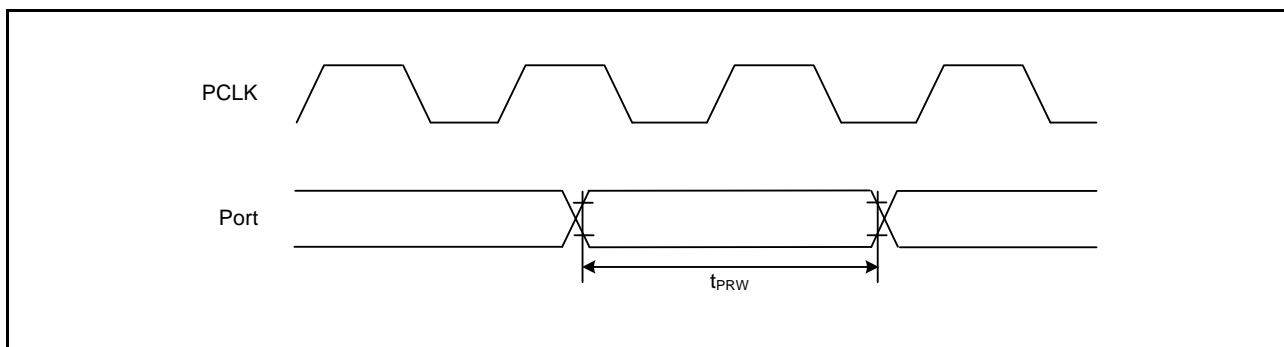


Figure 42.33 I/O Port Input Timing

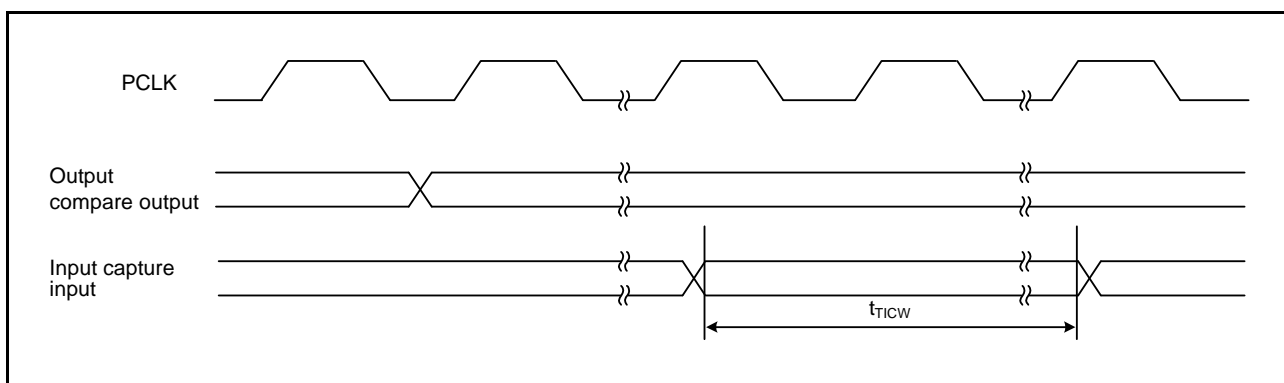


Figure 42.34 MTU2 Input/Output Timing

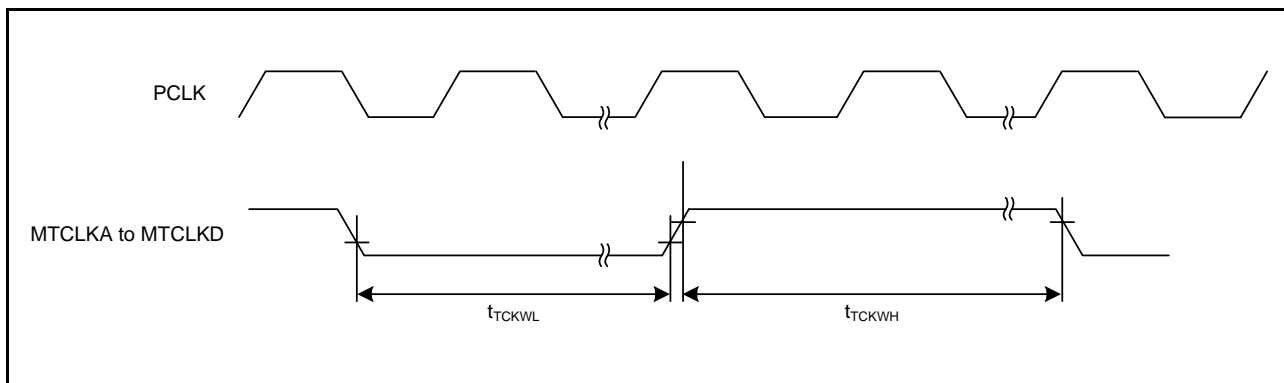


Figure 42.35 MTU2 Clock Input Timing

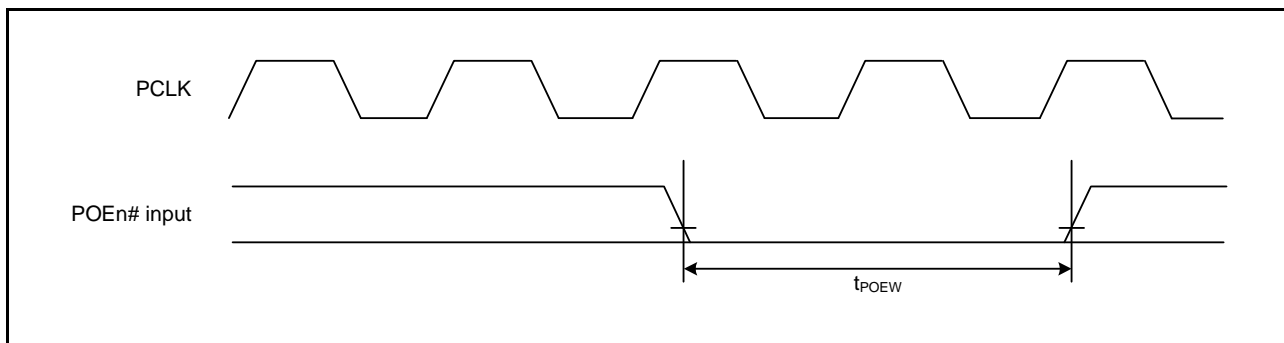


Figure 42.36 POE# Input Timing

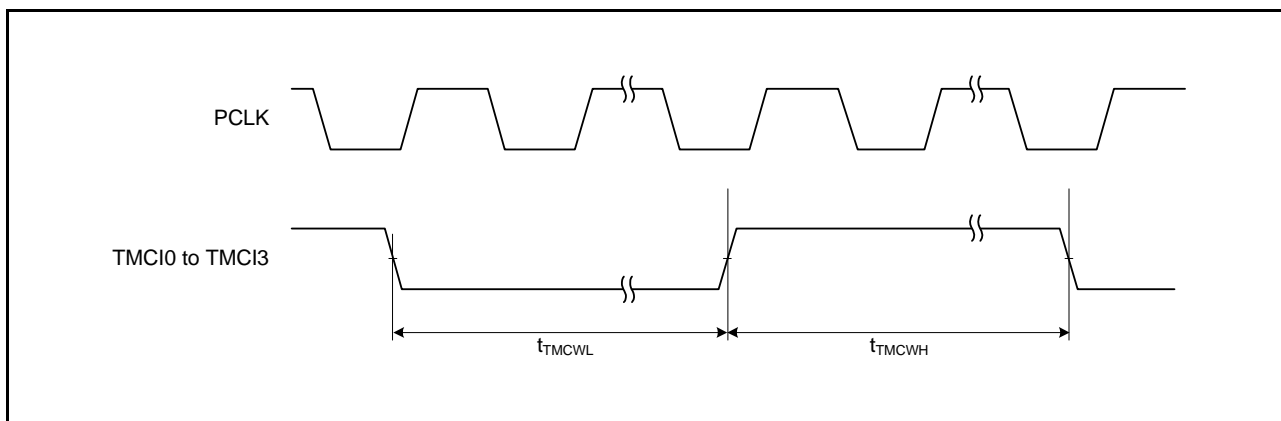


Figure 42.37 TMR Clock Input Timing

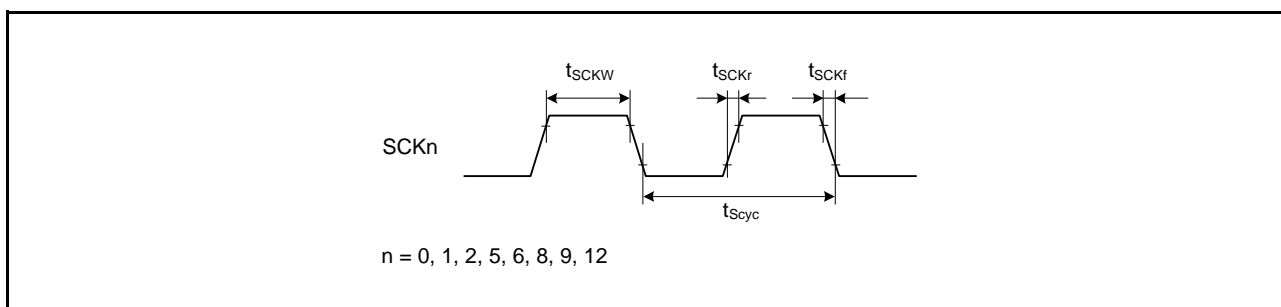


Figure 42.38 SCK Clock Input Timing

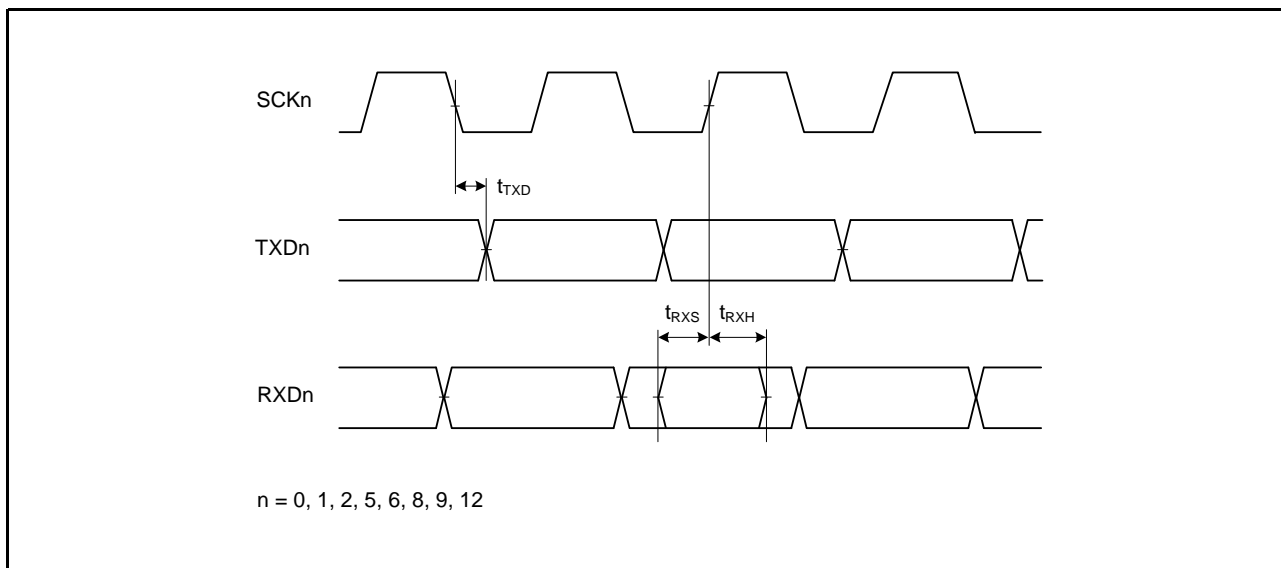


Figure 42.39 SCI Input/Output Timing: Clock Synchronous Mode

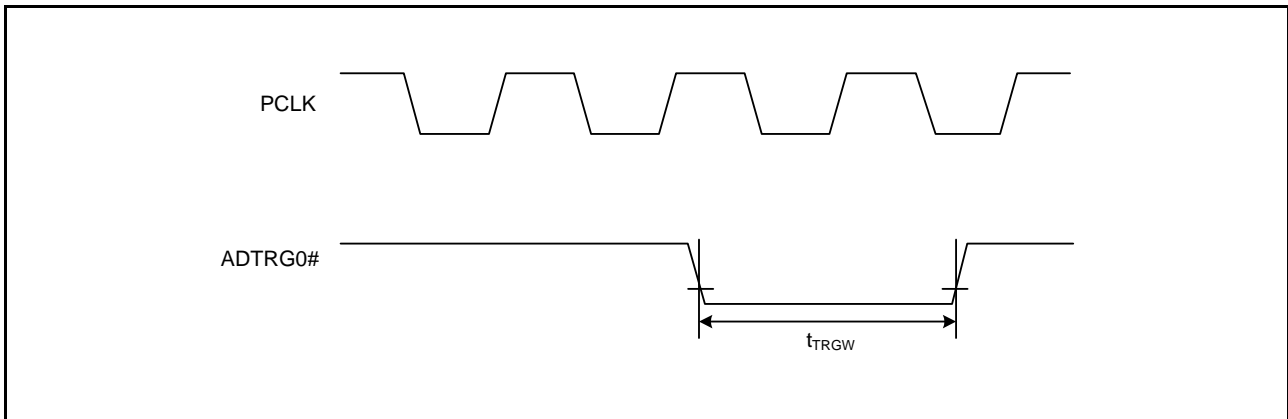


Figure 42.40 A/D Converter External Trigger Input Timing

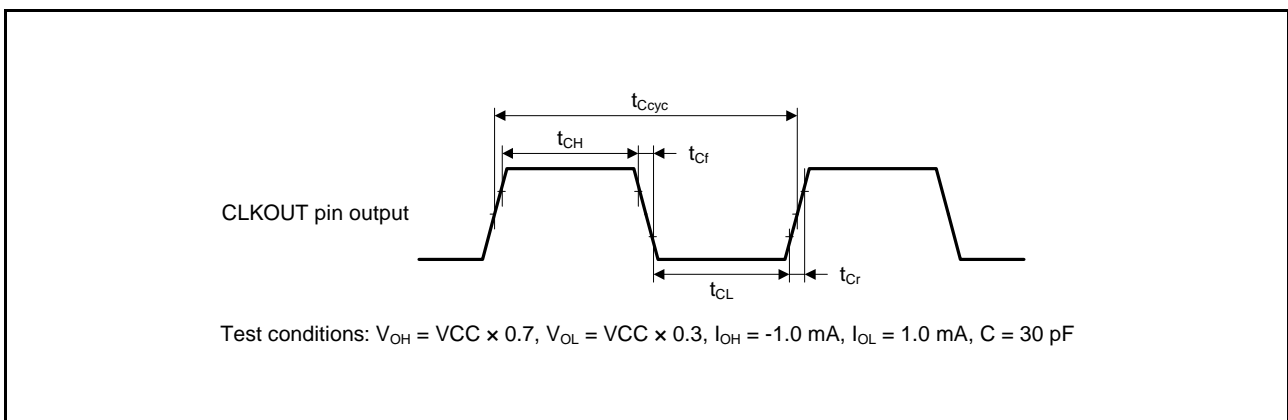


Figure 42.41 CLKOUT Output Timing

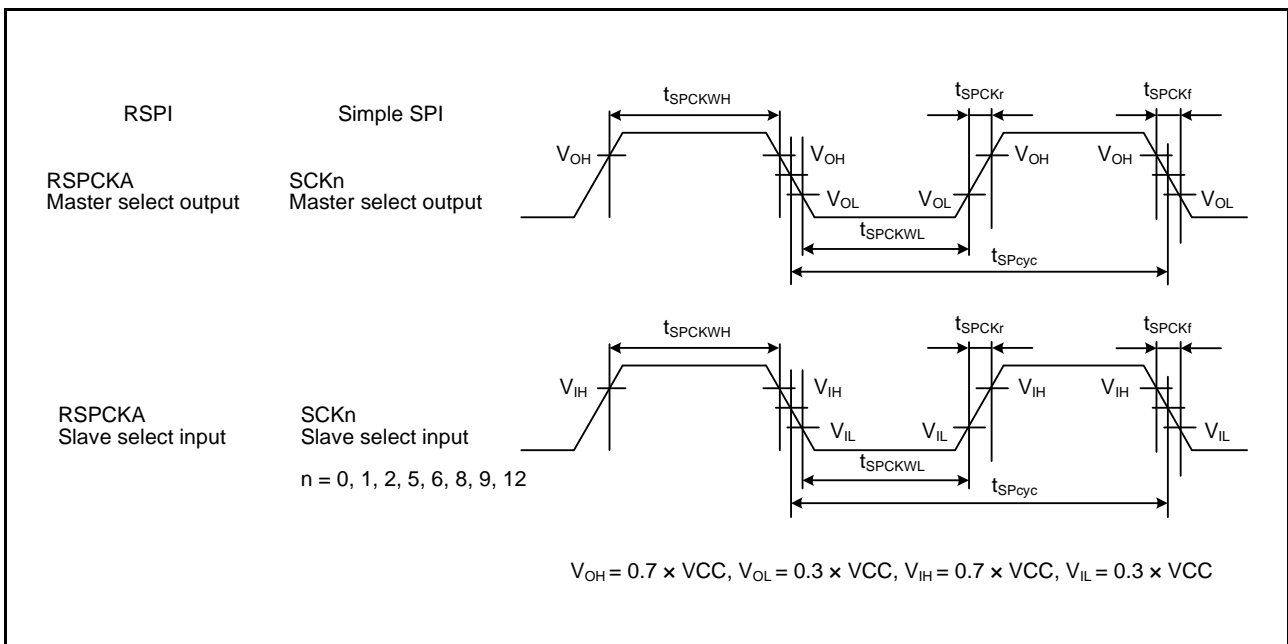


Figure 42.42 RSPi Clock Timing and Simple SPI Clock Timing

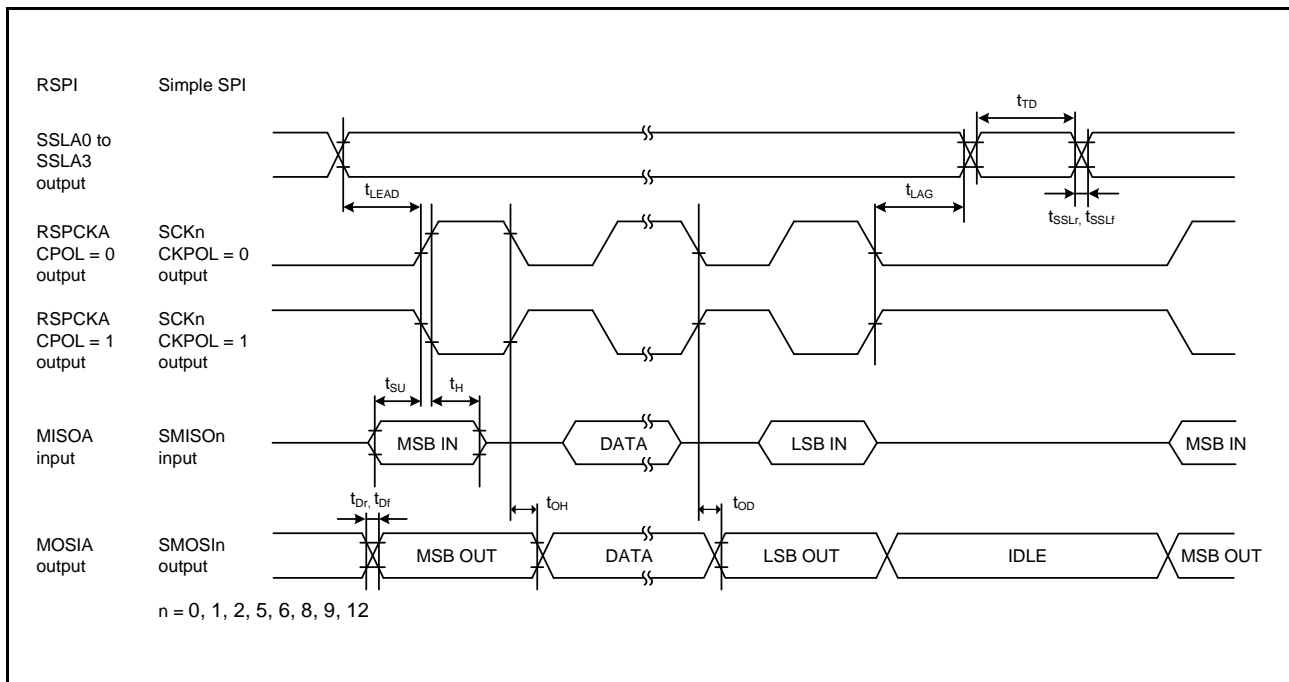


Figure 42.43 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKB Set to Division Ratio Other Than Divided by 2) and Simple SPI Timing (Master, CKPH = 1)

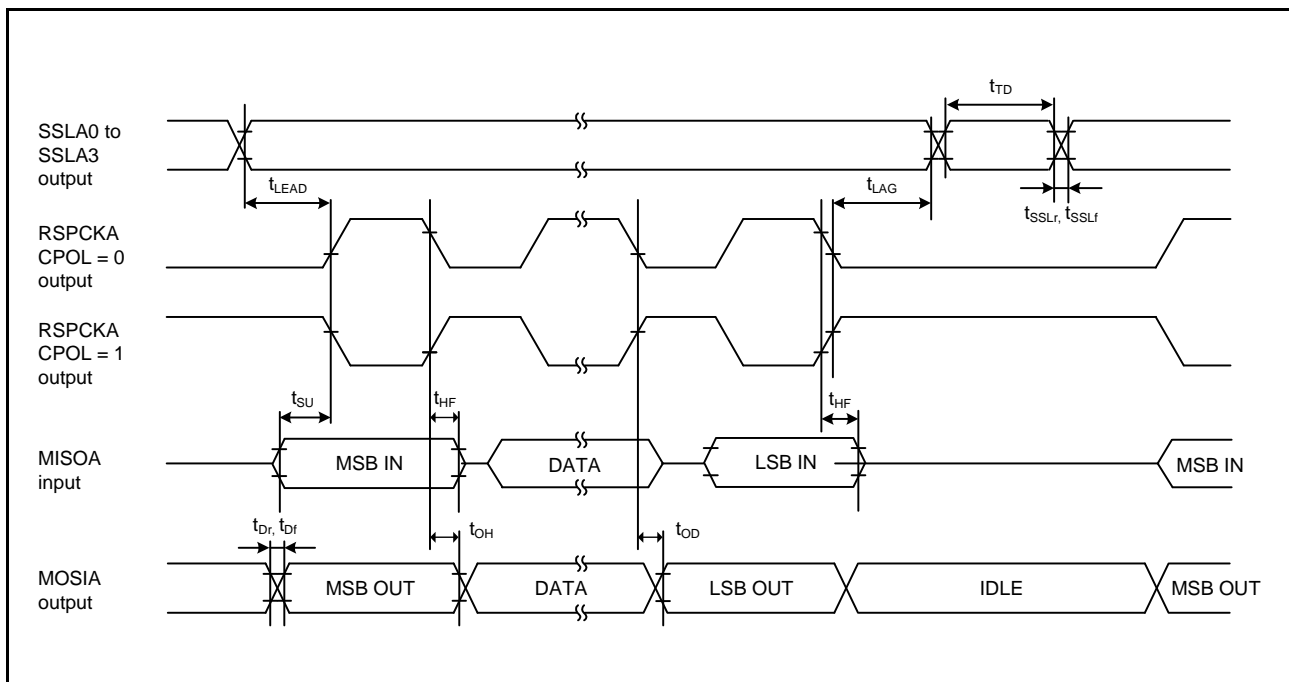


Figure 42.44 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKB Set to Divided by 2)

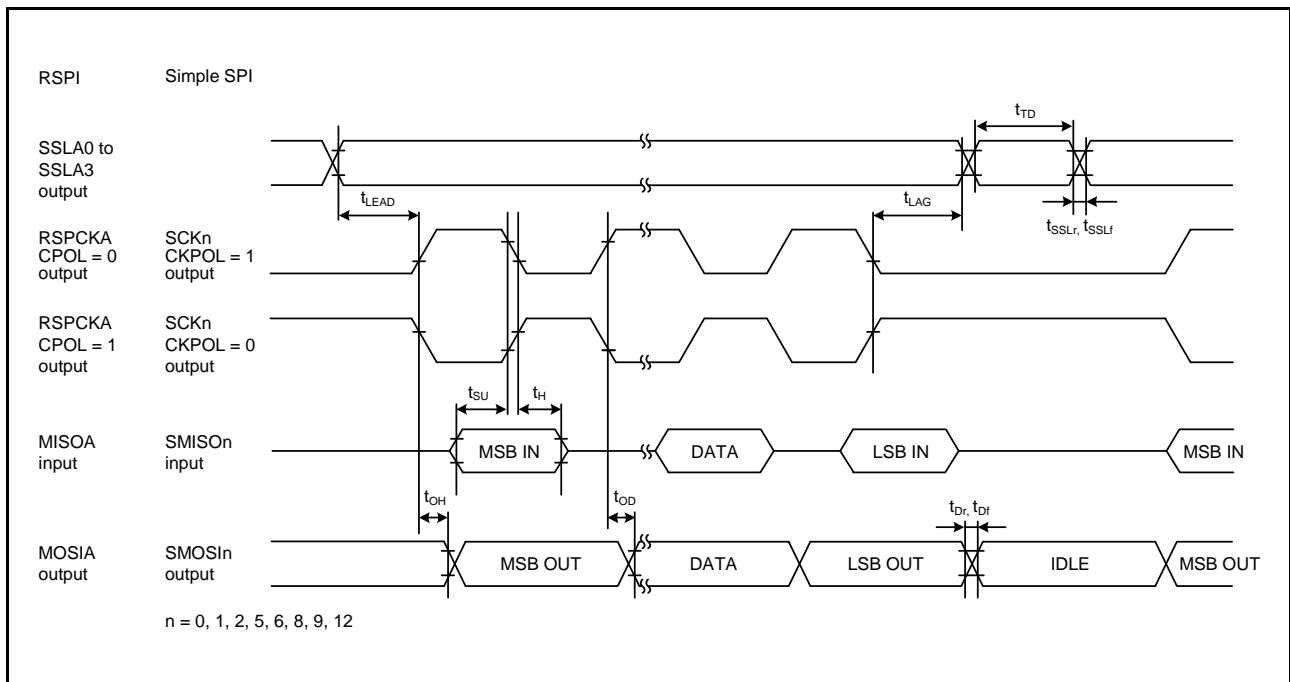


Figure 42.45 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKB Set to Division Ratio Other Than Divided by 2) and Simple SPI Timing (Master, CKPH = 0)

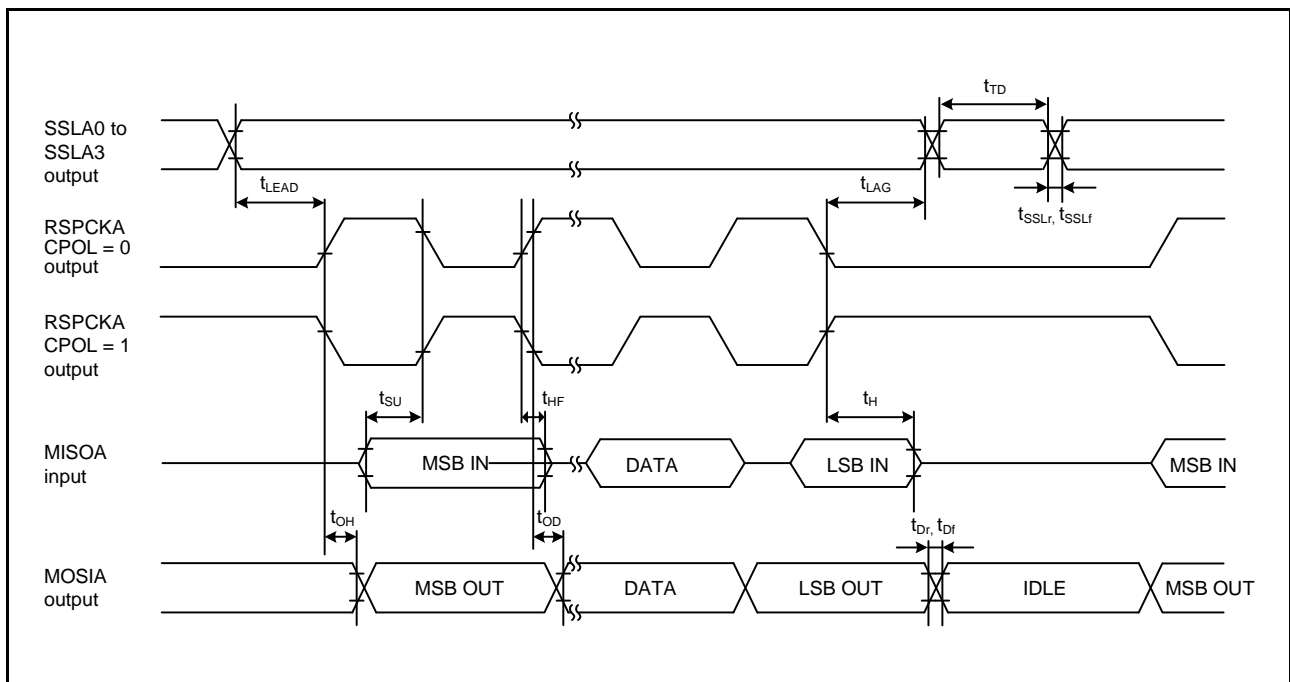


Figure 42.46 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKB Set to Divided by 2)



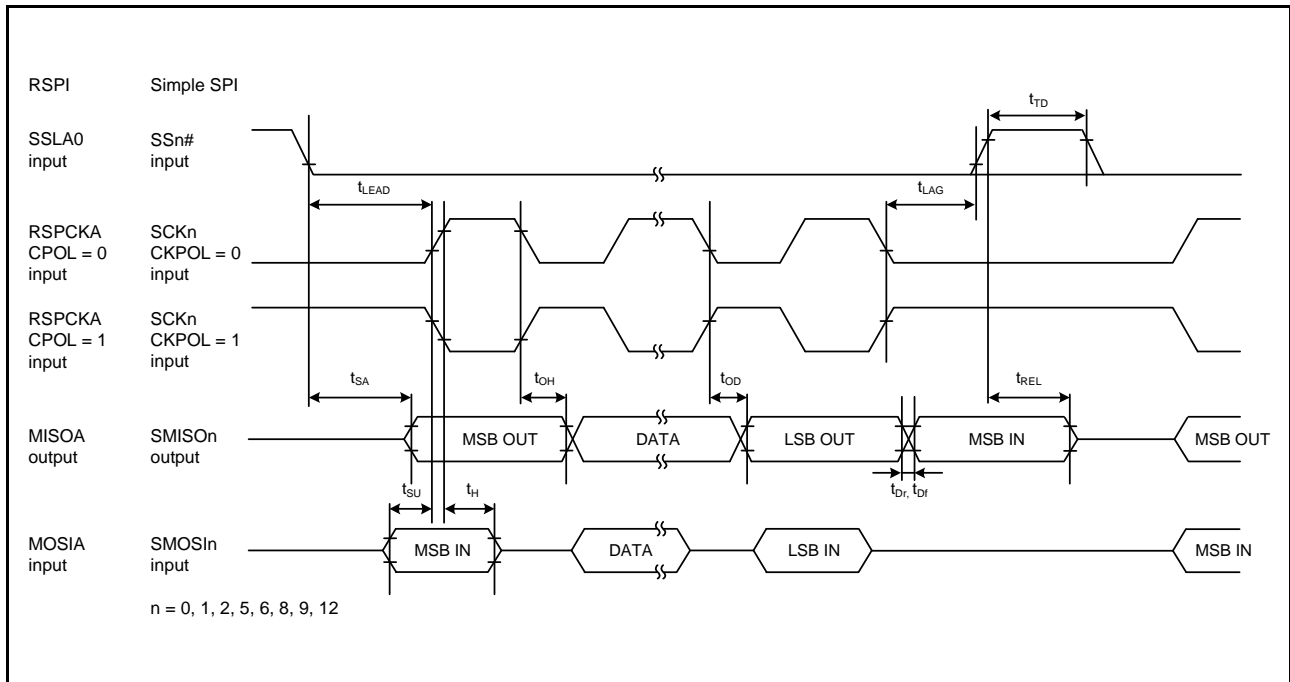


Figure 42.47 RSPI Timing (Slave, CPHA = 0) and Simple SPI Timing (Slave, CKPH = 1)

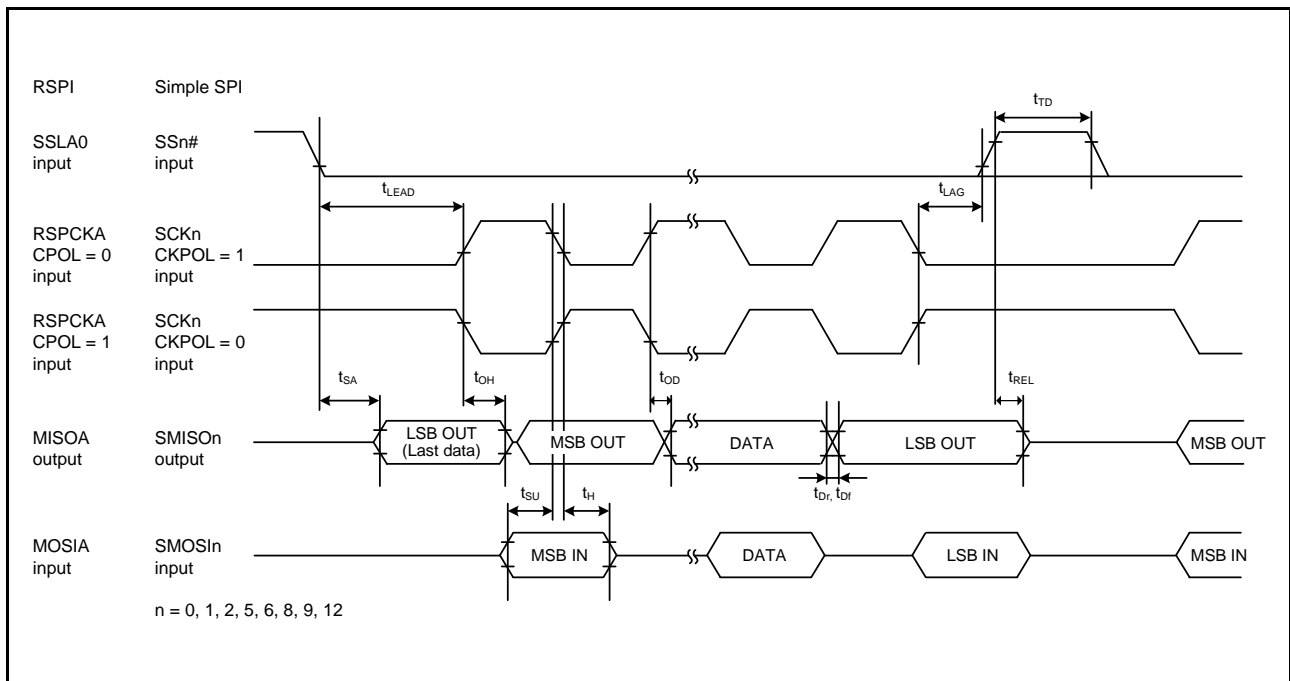


Figure 42.48 RSPI Timing (Slave, CPHA = 1) and Simple SPI Timing (Slave, CKPH = 0)

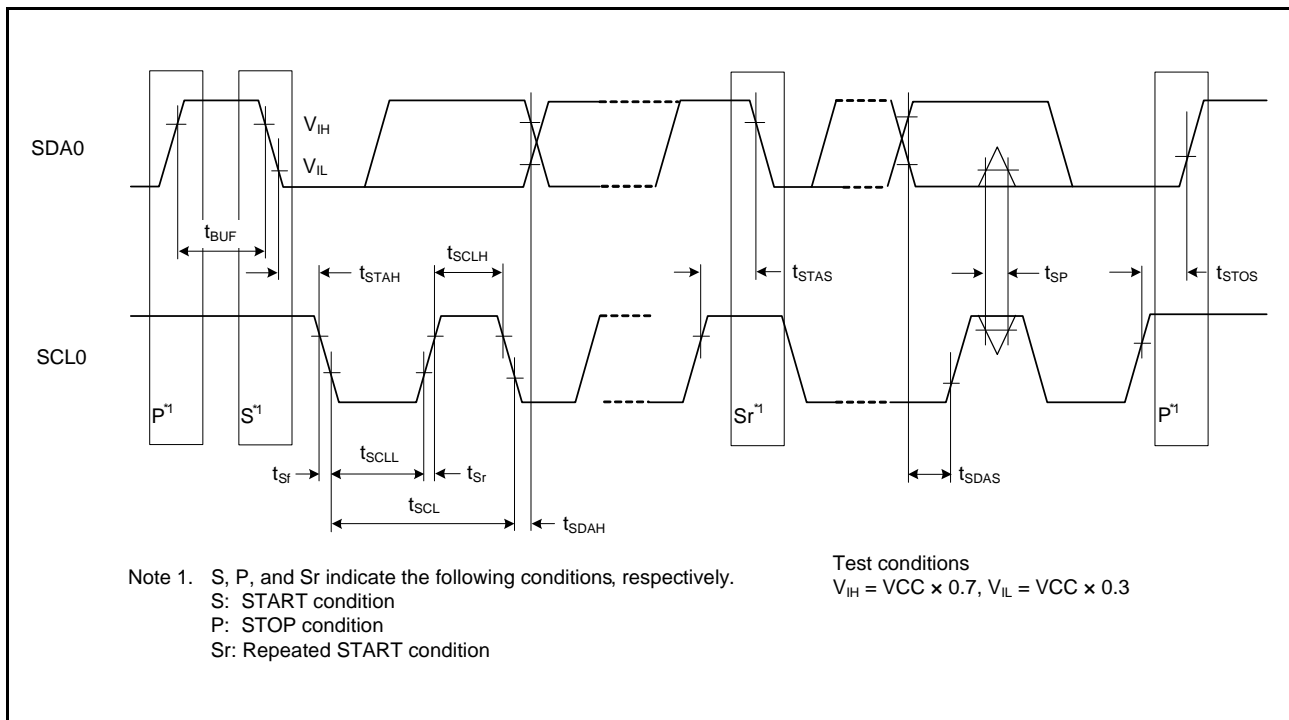


Figure 42.49 I2C Bus Interface Input/Output Timing and Simple I2C Bus Interface Input/Output Timing

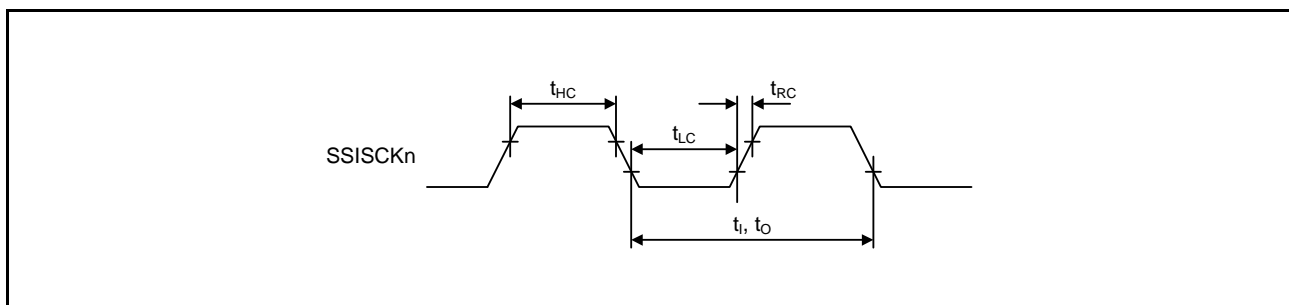


Figure 42.50 Clock Input/Output Timing

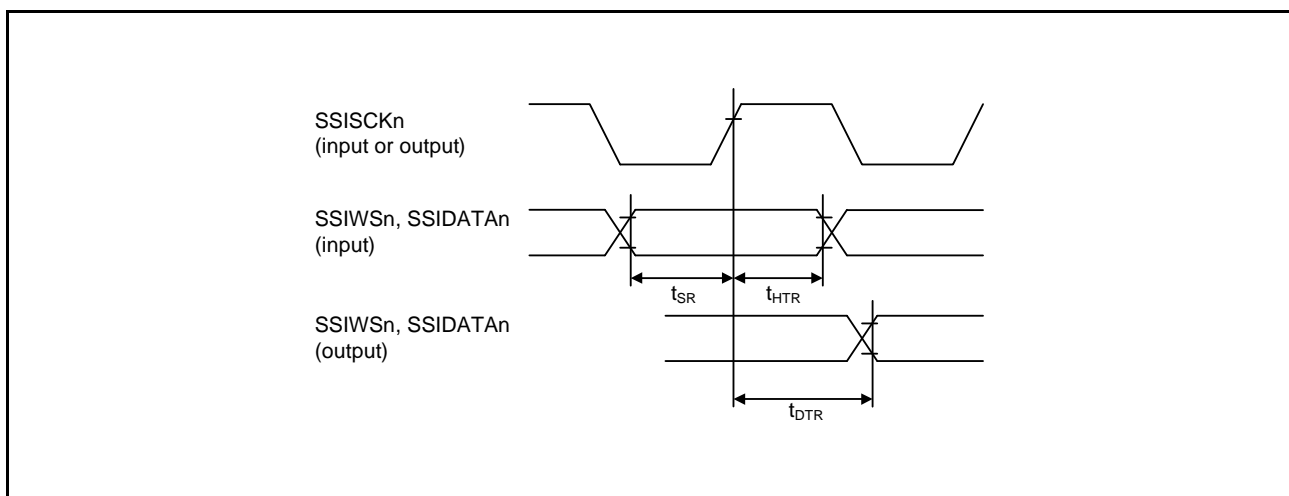


Figure 42.51 Transmission/Reception Timing (Synchronized with SSISCKn Rising Edge)

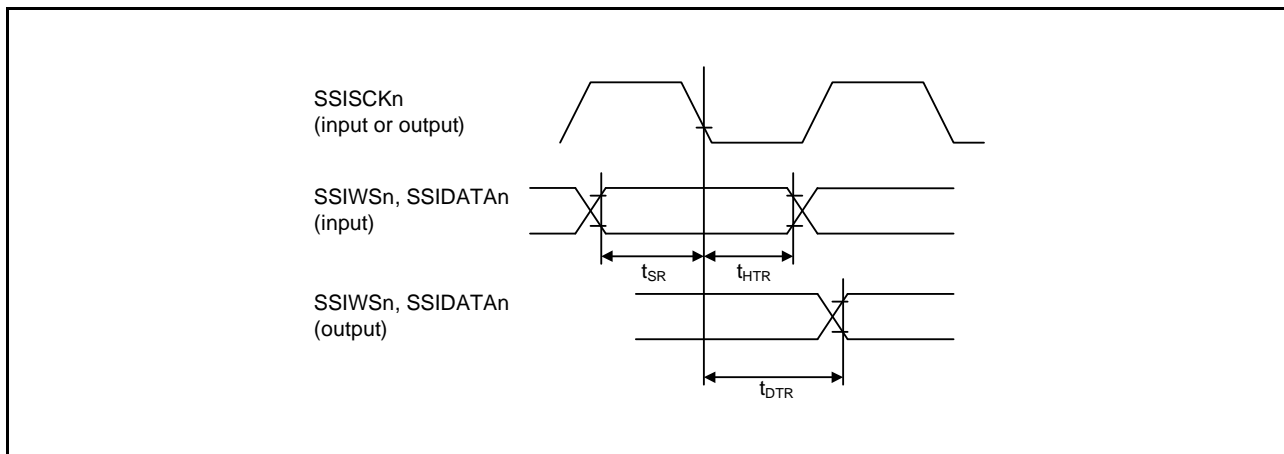


Figure 42.52 Transmission/Reception Timing (Synchronized with SSISCKn Falling Edge)

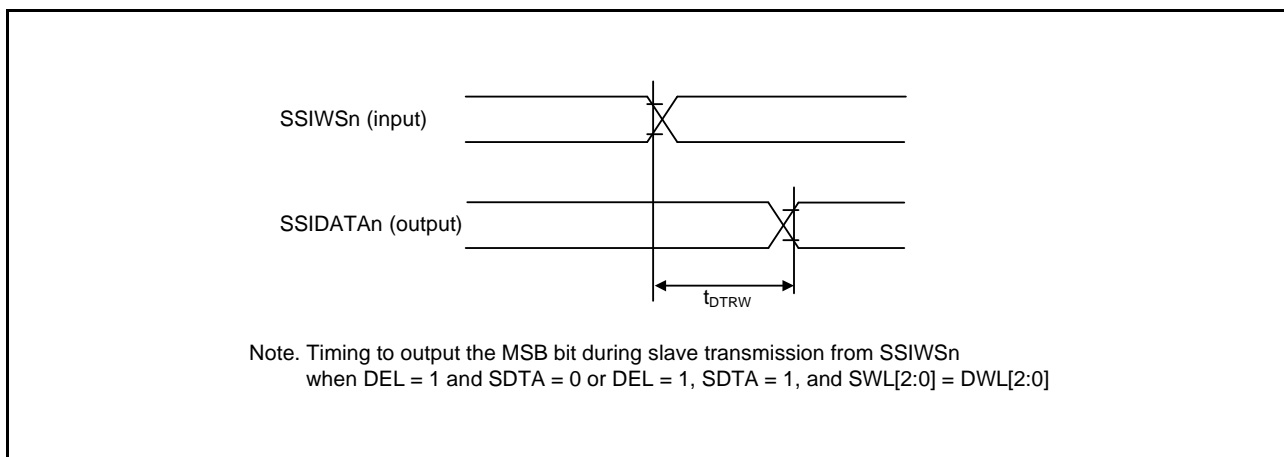


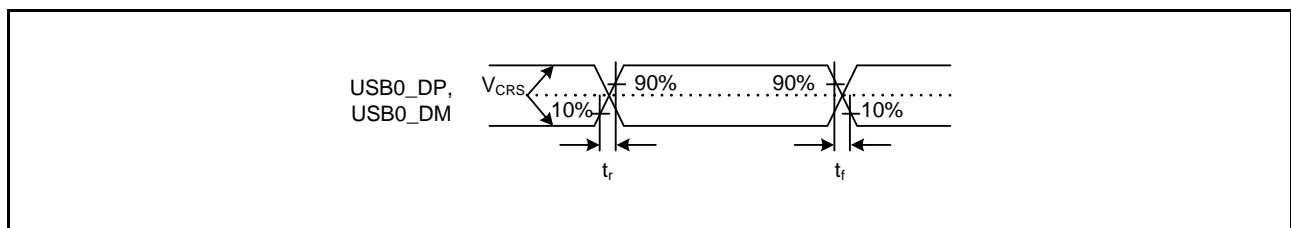
Figure 42.53 SSIDATA Output Delay After SSIWSn Changing Edge

42.4 USB Characteristics

**Table 42.36 USB Characteristics (USB0\_DP and USB0\_DM Pin Characteristics)**

Conditions: VCC = AVCC0 = VCC\_USB = 3.0 to 3.6 V, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

Item		Symbol	Min.	Max.	Unit	Test Conditions	
Input characteristics	Input high level voltage	V <sub>IH</sub>	2.0	—	V		
	Input low level voltage	V <sub>IL</sub>	—	0.8	V		
	Differential input sensitivity	V <sub>DI</sub>	0.2	—	V	USB0_DP – USB0_DM	
	Differential common mode range	V <sub>CM</sub>	0.8	2.5	V		
Output characteristics	Output high level voltage	V <sub>OH</sub>	2.8	VCC_USB	V	I <sub>OH</sub> = -200 μA	
	Output low level voltage	V <sub>OL</sub>	0.0	0.3	V	I <sub>OL</sub> = 2 mA	
	Cross-over voltage	V <sub>CRS</sub>	1.3	2.0	V	Figure 42.54 Figure 42.55	
	Rise time	FS	t <sub>r</sub>	4	20		ns
		LS		75	300		
	Fall time	FS	t <sub>f</sub>	4	20		ns
		LS		75	300		
	Rise/fall time ratio	FS	t <sub>r</sub> /t <sub>f</sub>	90	111.11		%
LS		80		125			
Output resistance		Z <sub>DRV</sub>	28	44	Ω	(Adjusting the resistance of external elements is not necessary.)	
VBUS characteristics	VBUS input voltage	V <sub>IH</sub>	VCC × 0.8	—	V		
		V <sub>IL</sub>	—	VCC × 0.2	V		
	VBUS (P16) input leakage current	I <sub>VBUSIN</sub>	—	10	μA	USB0_VBUS = 5.5 V	
Pull-up, pull-down	Pull-down resistor	R <sub>PD</sub>	14.25	24.80	kΩ		
	Pull-up resistor	R <sub>PUI</sub>	0.9	1.575	kΩ	During idle state	
		R <sub>PUA</sub>	1.425	3.09	kΩ	During reception	
Battery Charging Specification Ver 1.2	USB0_DP sink current	I <sub>DP_SINK</sub>	25	175	μA		
	USB0_DM sink current	I <sub>DM_SINK</sub>	25	175	μA		
	DCD source current	I <sub>DP_SRC</sub>	7	13	μA		
	Data detection voltage	V <sub>DAT_REF</sub>	0.25	0.4	V		
	USB0_DP source current	V <sub>DP_SRC</sub>	0.5	0.7	V	Output current = 250 μA	
	USB0_DM source current	V <sub>DM_SRC</sub>	0.5	0.7	V	Output current = 250 μA	



**Figure 42.54 USB0\_DP and USB0\_DM Output Timing**

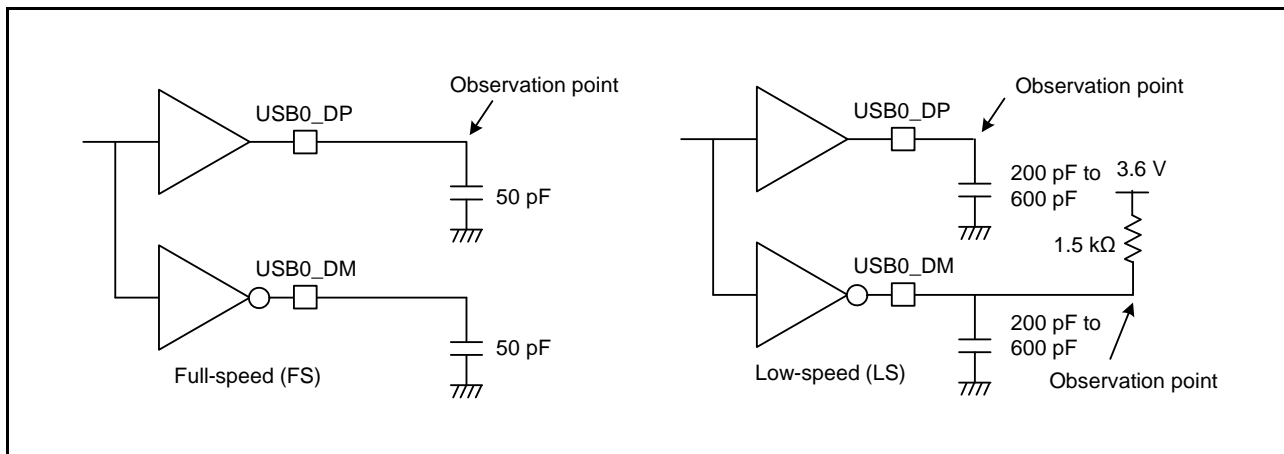


Figure 42.55 Test Circuit

42.5 A/D Conversion Characteristics

**Table 42.37 A/D Conversion Characteristics (1)**

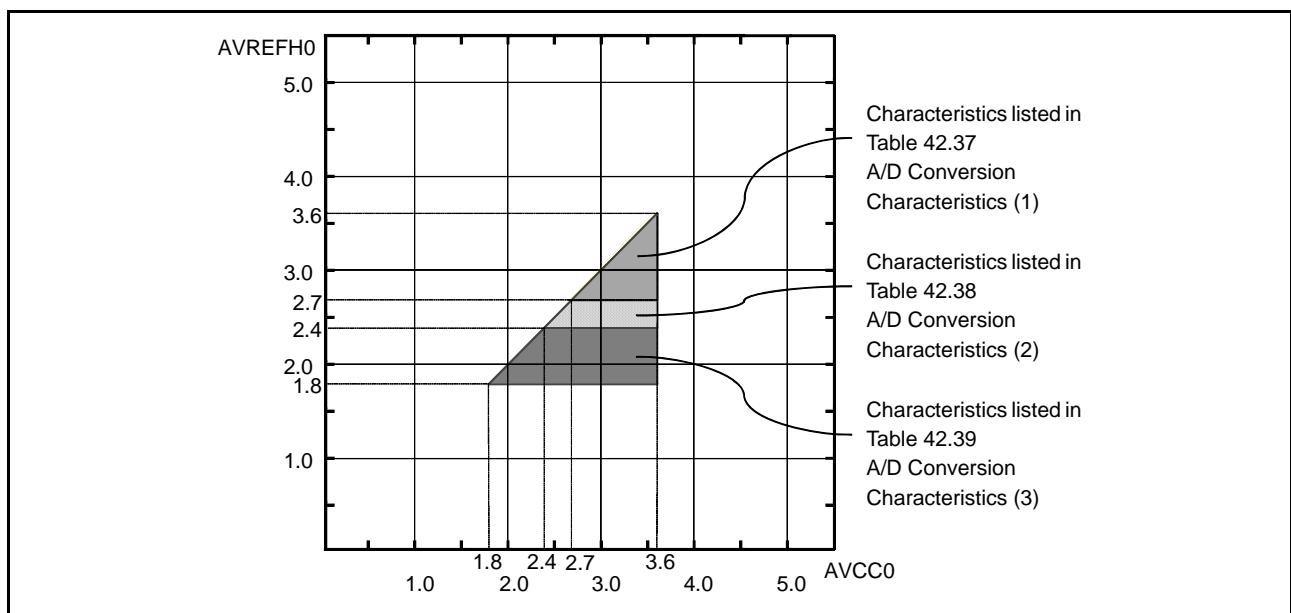
Conditions: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

Item	Min.	Typ.	Max.	Unit	Test Conditions	
Frequency	4	—	32	MHz		
Resolution	—	—	12	Bit		
Conversion time*1 (Operation at PCLKD = 32 MHz)	Permissible signal source impedance (Max.) = 0.3 kΩ	1.031 (0.313)*2	—	—	μs	High-precision channel ADCSR.ADHSC bit = 1 ADSSTRn.SST[7:0] bits = 09h
		1.375 (0.641)*2	—	—	μs	Normal-precision channel ADCSR.ADHSC bit = 1 ADSSTRn.SST[7:0] bits = 14h
Offset error		±0.5	±4.5	LSB	High-precision channel PJ6PFS.ASEL bit = 1 PJ7PFS.ASEL bit = 1	
				±6.0	LSB	Other than above
Full-scale error		±0.75	±4.5	LSB	High-precision channel PJ6PFS.ASEL bit = 1 PJ7PFS.ASEL bit = 1	
				±6.0	LSB	Other than above
Quantization error	—	±0.5	—	LSB		
Absolute accuracy		±1.25	±5.0	LSB	High-precision channel PJ6PFS.ASEL bit = 1 PJ7PFS.ASEL bit = 1	
				±8.0	LSB	Other than above
DNL differential nonlinearity error	—	±1.0	—	LSB		
INL integral nonlinearity error	—	±1.0	±3.0	LSB		

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The value in parentheses indicates the sampling time.



**Figure 42.56 AVCC0 to AVREFH0 Voltage Range**

**Table 42.38 A/D Conversion Characteristics (2)**Conditions:  $V_{CC} = AV_{CC0} = V_{REFH0} = 2.4$  to  $2.7$  V,  $V_{SS} = AV_{SS0} = V_{REFL0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		4	—	16	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (Operation at PCLKD = 16 MHz)	Permissible signal source impedance (Max.) = 1.0 k $\Omega$	2.062 (0.625)*2	—	—	$\mu\text{s}$	High-precision channel ADCSR.ADHSC bit = 1 ADSSTRn.SST[7:0] bits = 09h
		2.750 (1.313)*2	—	—	$\mu\text{s}$	Normal-precision channel ADCSR.ADHSC bit = 1 ADSSTRn.SST[7:0] bits = 14h
Offset error		—	$\pm 0.5$	$\pm 6.0$	LSB	
Full-scale error		—	$\pm 1.25$	$\pm 6.0$	LSB	
Quantization error		—	$\pm 0.5$	—	LSB	
Absolute accuracy		—	$\pm 3.0$	$\pm 8.0$	LSB	
DNL differential nonlinearity error		—	$\pm 1.0$	—	LSB	
INL integral nonlinearity error		—	$\pm 1.5$	$\pm 3.0$	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The value in parentheses indicates the sampling time.

**Table 42.39 A/D Conversion Characteristics (3)**Conditions:  $V_{CC} = AV_{CC0} = V_{REFH0} = 1.8$  to  $2.4$  V,  $V_{SS} = AV_{SS0} = V_{REFL0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	8	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (Operation at PCLKD = 8 MHz)	Permissible signal source impedance (Max.) = 5.0 k $\Omega$	4.875 (1.250)*2	—	—	$\mu\text{s}$	High-precision channel ADCSR.ADHSC bit = 0 ADSSTRn.SST[7:0] bits = 09h
		6.250 (2.625)*2	—	—	$\mu\text{s}$	Normal-precision channel ADCSR.ADHSC bit = 0 ADSSTRn.SST[7:0] bits = 14h
Offset error		—	$\pm 0.5$	$\pm 24.0$	LSB	
Full-scale error		—	$\pm 1.25$	$\pm 24.0$	LSB	
Quantization error		—	$\pm 0.5$	—	LSB	
Absolute accuracy		—	$\pm 2.75$	$\pm 32.0$	LSB	
DNL differential nonlinearity error		—	$\pm 1.0$	—	LSB	
INL integral nonlinearity error		—	$\pm 1.25$	$\pm 12.0$	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The value in parentheses indicates the sampling time.

**Table 42.40 A/D Conversion Characteristics (4)**

Conditions:  $V_{CC} = AVCC0 = 2.0$  to  $3.6$  V,  $V_{SS} = AVSS0 = VSS\_USB = 0$  V,  $ADHVREFCNT.OCSVSEL = 1$  (internal reference voltage selected as high-side reference voltage),  $PJ7PFS.ASEL = 0$  ( $AVSS0$  pin selected as low-side reference power supply ground pin)  
 $T_a = -40$  to  $+105^\circ\text{C}$

Item	Min.	Typ.	Max.	Unit	Test Conditions	
Frequency	1	—	2	MHz		
Resolution	—	—	12	Bit		
Internal reference voltage	1.36	1.43	1.50	V		
Conversion time*1 (Operation at PCLKD = 2 MHz)	Permissible signal source impedance (Max.) = 5.0 k $\Omega$	16 (1.5)*2	—	—	$\mu\text{s}$	High-precision channel ADCSR.ADHSC bit = 0 ADSSTRn.SST[7:0] bits = 02h
		17.5 (3.0)*2	—	—		Normal-precision channel ADCSR.ADHSC bit = 0 ADSSTRn.SST[7:0] bits = 05h
Offset error	—	—	$\pm 24.0$	LSB		
DNL differential nonlinearity error	—	$\pm 16.0$	—	LSB		
INL integral nonlinearity error	—	$\pm 16.0$	$\pm 32.0$	LSB		

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The value in parentheses indicates the sampling time.

**Table 42.41 A/D Converter Channel Classification**

Classification	Channel	Conditions	Remarks
High-precision channel	AN000 to AN007, AN021	$AVCC0 = 1.8$ to $3.6$ V	Pins AN000 to AN007 and AN021 cannot be used as digital outputs when the A/D converter is in use.
Normal-precision channel	AN008 to AN015		
Internal reference voltage input channel	Internal reference voltage	$AVCC0 = 2.0$ to $3.6$ V	
Temperature sensor input channel	Temperature sensor output	$AVCC0 = 2.0$ to $3.6$ V	

**Table 42.42 A/D Internal Reference Voltage Characteristics**

Conditions:  $V_{CC} = AVCC0 = VCC\_USB = 2.0$  to  $3.6$  V\*1,  $V_{SS} = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

Item	Min.	Typ.	Max.	Unit	Test Conditions
Internal reference voltage input channel*2	1.36	1.43	1.50	V	

Note 1. The internal reference voltage cannot be selected for input channels when  $AVCC0 < 2.0$  V.

Note 2. The A/D internal reference voltage indicates the voltage when the internal reference voltage is input to the A/D converter.



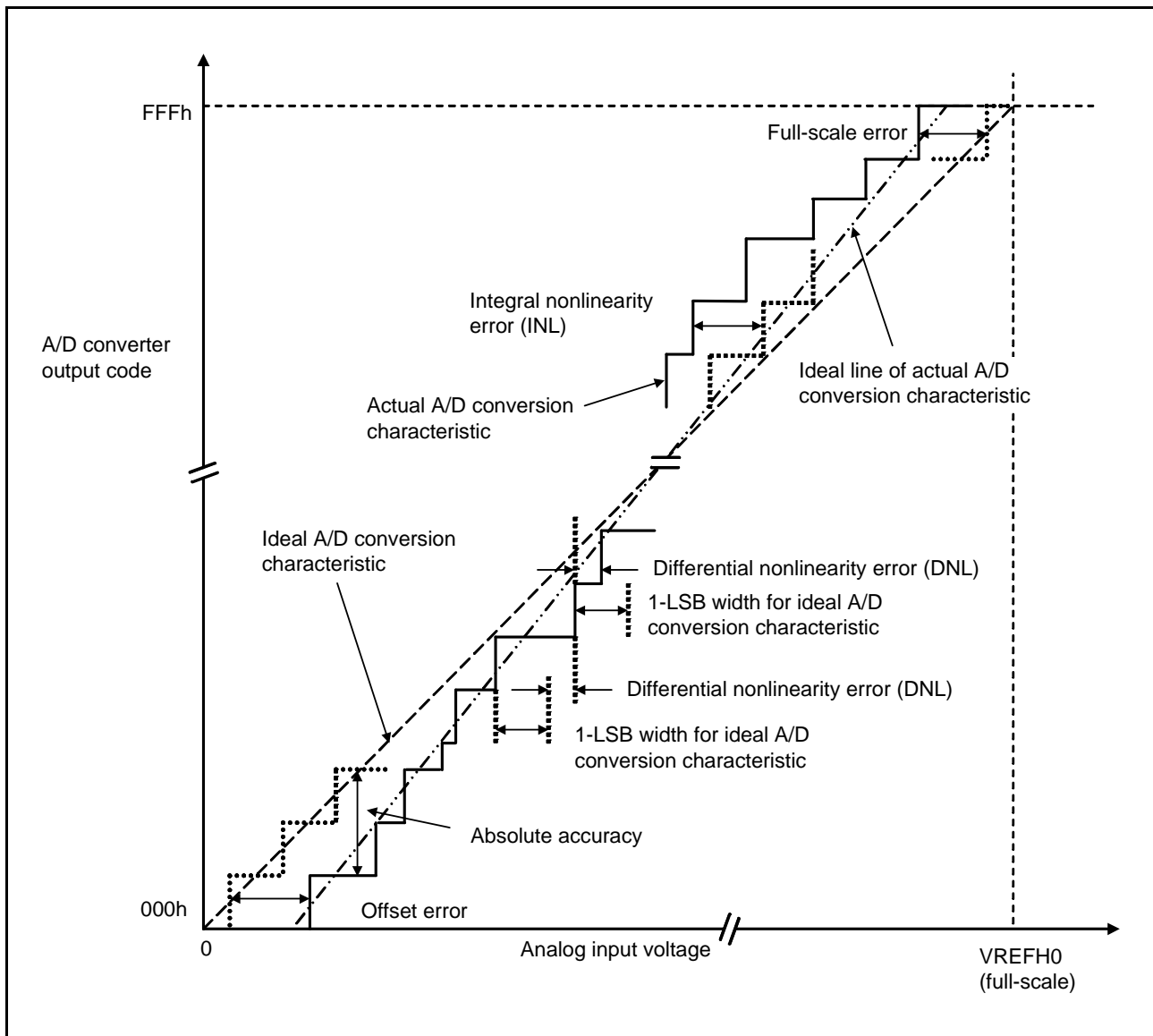


Figure 42.57 Illustration of A/D Converter Characteristic Terms

### Absolute accuracy

Absolute accuracy is the difference between output code based on the theoretical A/D conversion characteristics, and the actual A/D conversion result. When measuring absolute accuracy, the voltage at the midpoint of the width of analog input voltage (1-LSB width), that can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as an analog input voltage. For example, if 12-bit resolution is used and if reference voltage ( $V_{REFH0} = 3.072 \text{ V}$ ), then 1-LSB width becomes 0.75 mV, and 0 mV, 0.75 mV, 1.5 mV, ... are used as analog input voltages.

If analog input voltage is 6 mV, absolute accuracy =  $\pm 5 \text{ LSB}$  means that the actual A/D conversion result is in the range of 003h to 00Dh though an output code, 008h, can be expected from the theoretical A/D conversion characteristics.

### Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

**Differential nonlinearity error (DNL)**

Differential nonlinearity error is the difference between 1-LSB width based on the ideal A/D conversion characteristics and the width of the actual output code.

**Offset error**

Offset error is the difference between a transition point of the ideal first output code and the actual first output code.

**Full-scale error**

Full-scale error is the difference between a transition point of the ideal last output code and the actual last output code.

## 42.6 D/A Conversion Characteristics

**Table 42.43 D/A Conversion Characteristics (1)**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = 1.8$  to  $3.6$  V,  $V_{SS} = AVSS0 = VREFL = VSS\_USB = 0$  V,  
 $T_a = -40$  to  $+105^\circ\text{C}$   
 Reference voltage = VREFH or VREFL selected

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	—	—	12	Bit	
Resistive load	30	—	—	k $\Omega$	
Capacitive load	—	—	50	pF	
Output voltage range*1	0.35	—	AVCC0 - 0.47	V	
DNL differential nonlinearity error	—	$\pm 0.5$	$\pm 1.0$	LSB	
INL integral nonlinearity error	—	$\pm 2.0$	$\pm 8.0$	LSB	
Offset error	—	—	$\pm 20$	mV	
Full-scale error	—	—	$\pm 20$	mV	
Output resistance	—	75	—	$\Omega$	
Conversion time	—	—	30	$\mu\text{s}$	

Note 1. There are restrictions on AVCC0 and VCC depending on the usage conditions for the 12-bit D/A converter and I/O ports.  
 When using ports J0 and J2 as DA0 and DA1 output, make sure that  $V_{CC} \geq$  D/A output voltage.

**Table 42.44 D/A Conversion Characteristics (2)**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = 1.8$  to  $3.6$  V,  $V_{SS} = AVSS0 = VREFL = VSS\_USB = 0$  V,  
 $T_a = -40$  to  $+105^\circ\text{C}$   
 Reference voltage = AVCC0 or AVSS0 selected

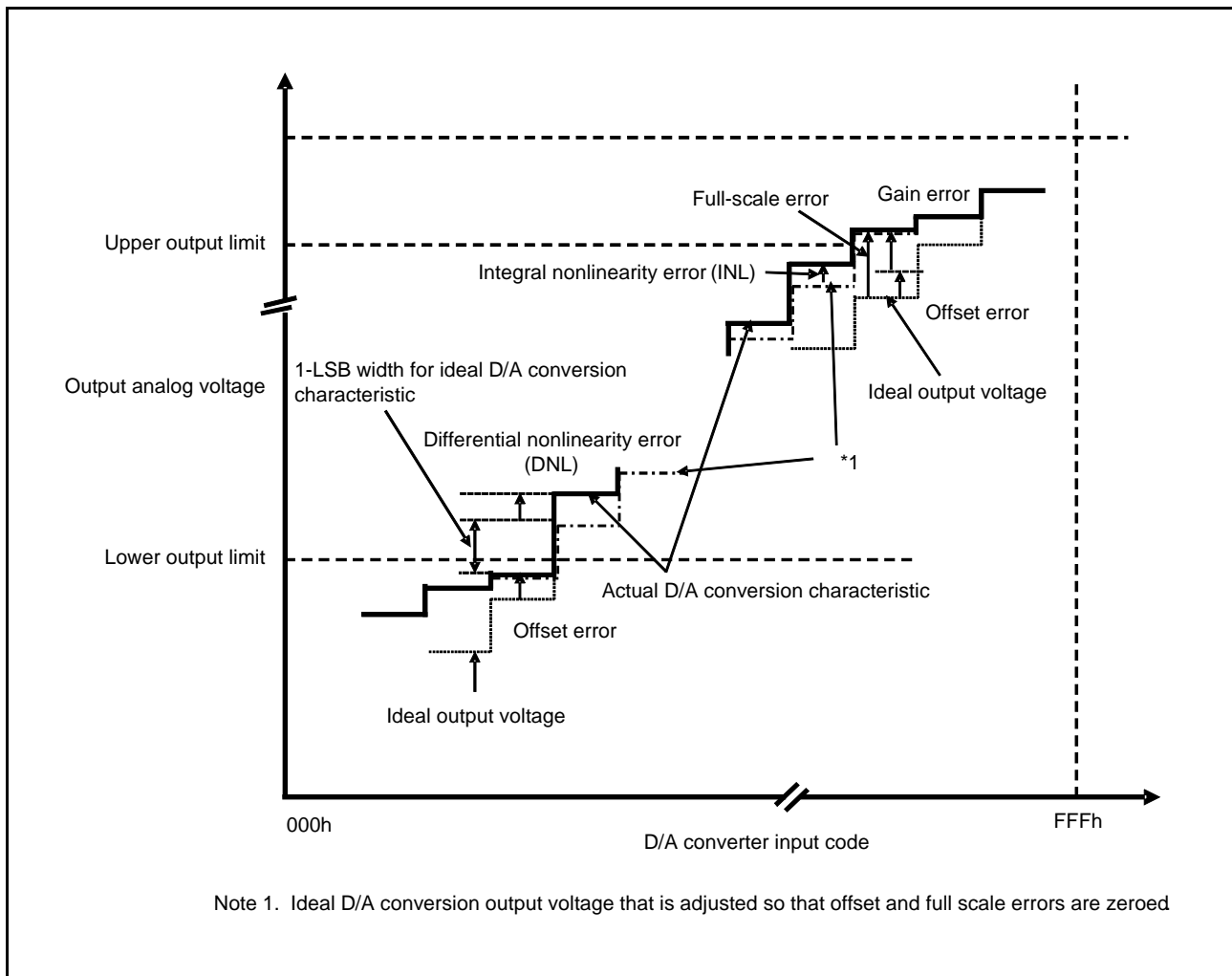
Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	—	—	12	Bit	
Resistive load	30	—	—	k $\Omega$	
Capacitive load	—	—	50	pF	
Output voltage range*1	0.35	—	AVCC0 - 0.47	V	
DNL differential nonlinearity error	—	$\pm 0.5$	$\pm 2.0$	LSB	
INL integral nonlinearity error	—	$\pm 2.0$	$\pm 8.0$	LSB	
Offset error	—	—	$\pm 30$	mV	
Full-scale error	—	—	$\pm 30$	mV	
Output resistance	—	75	—	$\Omega$	
Conversion time	—	—	30	$\mu\text{s}$	

Note 1. There are restrictions on AVCC0 and VCC depending on the usage conditions for the 12-bit D/A converter and I/O ports.  
 When using ports J0 and J2 as DA0 and DA1 output, make sure that  $V_{CC} \geq$  D/A output voltage.

**Table 42.45 D/A Conversion Characteristics (3)**Conditions:  $V_{CC} = AVCC0 = VCC\_USB = 2.0$  to  $3.6$  V,  $V_{SS} = AVSS0 = VSS\_USB = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Reference voltage = internal reference voltage selected

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	—	—	12	Bit	
Internal reference voltage (Vbgr)	1.36	1.43	1.50	V	
Resistive load	30	—	—	k $\Omega$	
Capacitive load	—	—	50	pF	
Output voltage range	0.35	—	Vbgr	V	
DNL differential nonlinearity error	—	$\pm 2.0$	$\pm 16.0$	LSB	
INL integral nonlinearity error	—	$\pm 8.0$	$\pm 16.0$	LSB	
Offset error	—	—	$\pm 30$	mV	
Output resistance	—	75	—	$\Omega$	
Conversion time	—	—	30	$\mu\text{s}$	



**Figure 42.58 Illustration of D/A Converter Characteristic Terms**

### Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal output voltage based on the ideal conversion characteristic when the measured offset and full-scale errors are zeroed, and the actual output voltage.

### Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between 1-LSB voltage width based on the ideal D/A conversion characteristics and the width of the actual output voltage.

### Offset error

Offset error is the difference between the highest actual output voltage that falls below the lower output limit and the ideal output voltage based on the input code.

### Full-scale error

Full-scale error is the difference between the lowest actual output voltage that exceeds the upper output limit and the ideal output voltage based on the input code.

## 42.7 Temperature Sensor Characteristics

**Table 42.46 Temperature Sensor Characteristics**

Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 2.0$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Relative accuracy	—	—	$\pm 1.5$	—	°C	2.4 V or above
		—	$\pm 2.0$	—		Below 2.4 V
Temperature slope	—	—	-3.65	—	mV/°C	
Output voltage (25°C)	—	—	1.05	—	V	$V_{CC} = 3.3$ V
Temperature sensor start time	$t_{START}$	—	—	5	$\mu\text{s}$	
Sampling time	—	5	—	—	$\mu\text{s}$	

## 42.8 Comparator Characteristics

**Table 42.47 Comparator Characteristics**

Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 1.8$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
CVREFB0 or CVREFB1 input reference voltage	VREF	0	—	$V_{CC} - 1.4$	V	
CMPB0 or CMPB1 input voltage	VI	-0.3	—	$V_{CC} + 0.3$	V	
Offset	Comparator high-speed mode	—	—	50	mV	
	Comparator high-speed mode Window function enabled	—	—	60	mV	
	Comparator low-speed mode	—	—	40	mV	
Comparator output delay time	Comparator high-speed mode	$T_d$	—	1.2	$\mu\text{s}$	$V_{CC} = 3$ V, input slew rate $\geq 50$ mV/ $\mu\text{s}$
	Comparator high-speed mode Window function enabled	$T_{dw}$	—	2	$\mu\text{s}$	
	Comparator low-speed mode	$T_d$	—	5	$\mu\text{s}$	
High-side reference voltage (comparator high-speed mode, window function enabled)	VRFH	—	0.76 VCC	—	V	
Low-side reference voltage (comparator high-speed mode, window function enabled)	VRFL	—	0.24 VCC	—	V	
Operation stabilization wait time	$T_{cmp}$	100	—	—	$\mu\text{s}$	

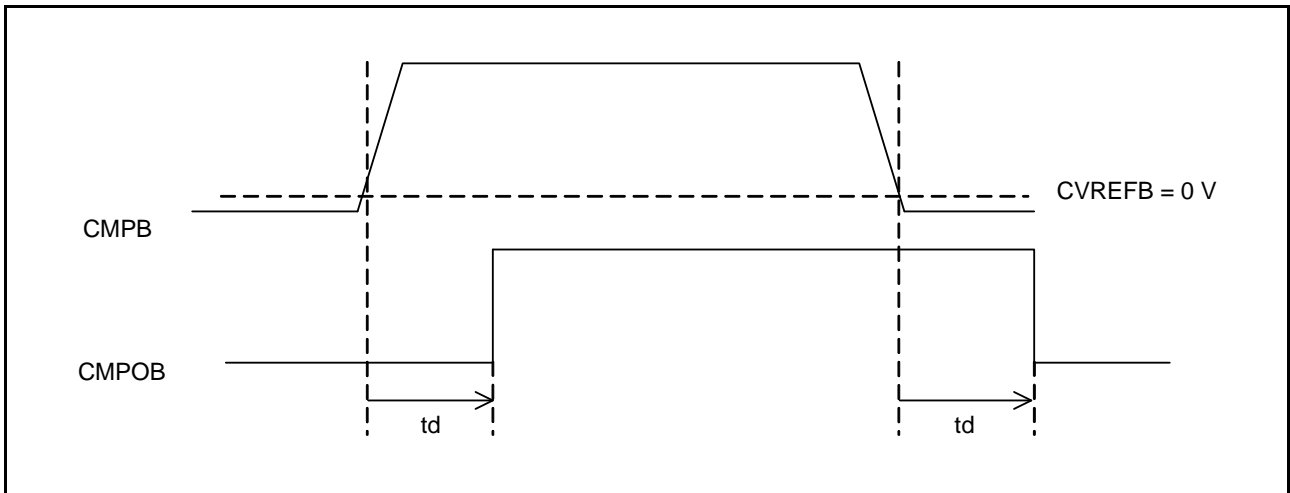


Figure 42.59 Comparator Output Delay Time in Comparator High-Speed Mode and Low-Speed Mode

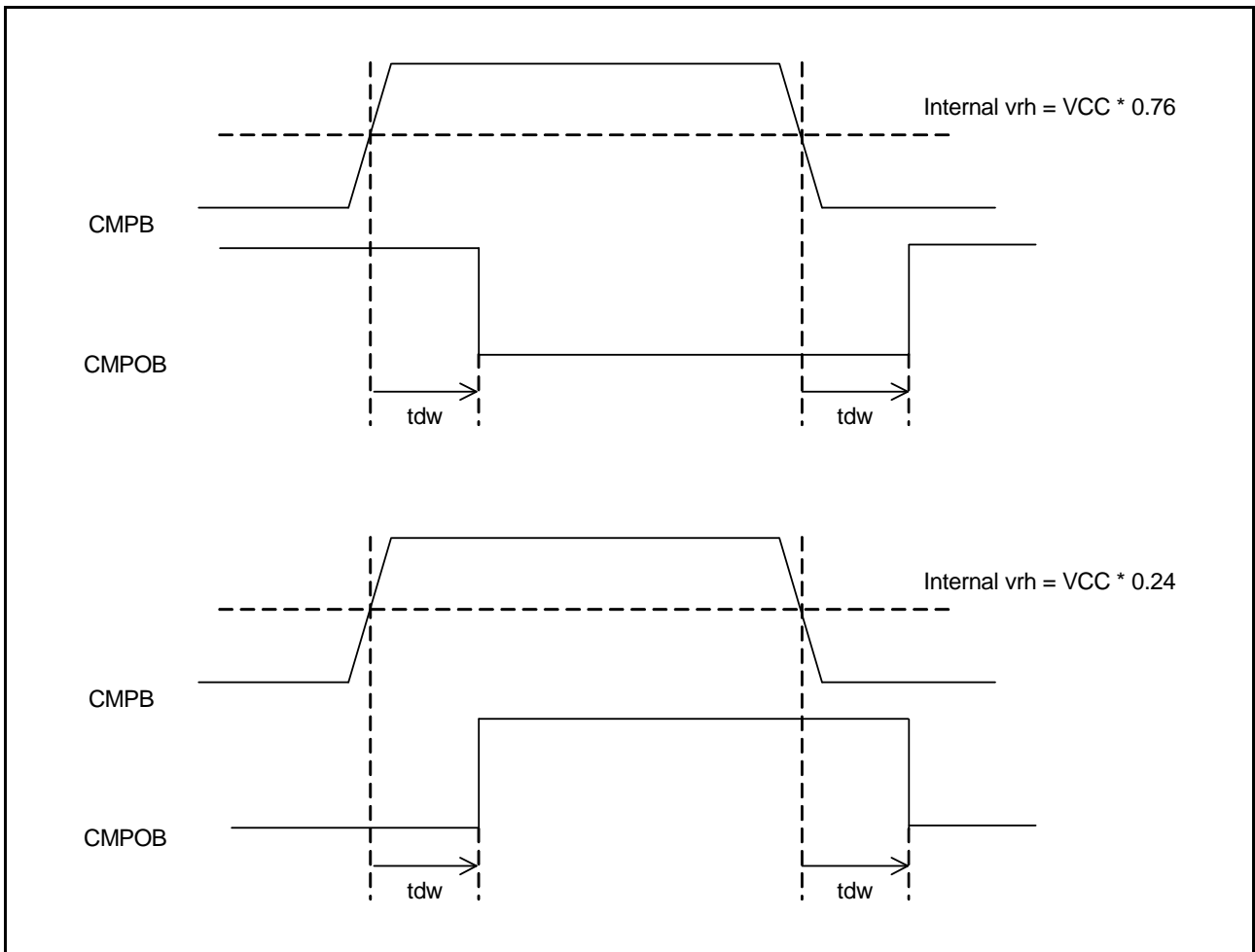


Figure 42.60 Comparator Output Delay Time in High-Speed Mode with Window Function Enabled

## 42.9 LCD Characteristics

### 42.9.1 External Resistance Division Method

#### (1) Static Display Mode

**Table 42.48 LCD Characteristics**

Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 2.0$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
LCD drive voltage	$V_{L4}$	2.0	—	VCC	V	

#### (2) 1/2 Bias Method, 1/4 Bias Method

**Table 42.49 LCD Characteristics**

Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 2.7$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
LCD drive voltage	$V_{L4}$	2.7	—	VCC	V	

#### (3) 1/3 Bias Method

**Table 42.50 LCD Characteristics**

Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 2.5$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
LCD drive voltage	$V_{L4}$	2.5	—	VCC	V	



## 42.9.2 Internal Voltage Boosting Method

**Table 42.51 Internal Voltage Boosting Method**Conditions:  $V_{CC} = AVCC0 = V_{CC\_USB} = 1.8$  to  $3.6$  V,  $V_{SS} = AVSS0 = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
External capacitance connected between CAPH and CAPL pins	C1	0.33	0.47	0.61	$\mu\text{F}$	
External capacitor connected to $V_{L1}$ pin	C2	0.33	0.47	0.61	$\mu\text{F}$	
External capacitor connected to $V_{L2}$ pin	C3	0.33	0.47	0.61	$\mu\text{F}$	
External capacitor connected to $V_{L3}$ pin	C4	0.33	0.47	0.61	$\mu\text{F}$	
External capacitor connected to $V_{L4}$ pin	C5	0.33	0.47	0.61	$\mu\text{F}$	

## (1) 1/3 Bias Method

**Table 42.52 Internal Voltage Boosting Method LCD Characteristics**Conditions:  $V_{CC} = AVCC0 = V_{CC\_USB} = 1.8$  V to  $3.6$  V,  $V_{SS} = AVSS0 = V_{SS\_USB} = 0$  V,  $T_a = -40$  to  $+105^\circ\text{C}$ 

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit	Test Conditions	
LCD output voltage variation range	$V_{L1}$	C1 to C4 connected	VLCD = 04h	0.9	1.0	1.08	V	
			VLCD = 05h	0.95	1.05	1.13	V	
			VLCD = 06h	1	1.1	1.18	V	
			VLCD = 07h	1.05	1.15	1.23	V	
			VLCD = 08h	1.1	1.2	1.28	V	
			VLCD = 09h	1.15	1.25	1.33	V	
			VLCD = 0Ah	1.2	1.3	1.38	V	
			VLCD = 0Bh	1.25	1.35	1.43	V	
			VLCD = 0Ch	1.3	1.4	1.48	V	
			VLCD = 0Dh	1.35	1.45	1.53	V	
			VLCD = 0Eh	1.4	1.5	1.58	V	
			VLCD = 0Fh	1.45	1.55	1.63	V	
			VLCD = 10h	1.5	1.6	1.68	V	
			VLCD = 11h	1.55	1.65	1.73	V	
VLCD = 12h	1.6	1.70	1.78	V				
VLCD = 13h	1.65	1.75	1.83	V				
Doubler output voltage	$V_{L2}$	C1 to C3, C5 connected	$2V_{L1} - 0.10$	$2V_{L1}$	$2V_{L1}$	V		
Tripler output voltage	$V_{L3}$	C1 to C5 connected	$3V_{L1} - 0.15$	$3V_{L1}$	$3V_{L1}$	V		
Reference voltage setup time*1	$t_{VL1S}$		5	—	—	ms		
LCD output voltage variation range*2	$t_{VLWT}$	C1 to C4 connected	500	—	—	ms		

Note 1. This is the required wait time from when the reference voltage is specified by the VLCD register (or when the internal voltage boosting method is selected (LCDM0.MDSET1 and MDSET0 = 01b) if the default reference voltage value is used) until voltage boosting starts (VLCON = 1).

Note 2. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

## (2) 1/4 Bias Method

**Table 42.53 Internal Voltage Boosting Method LCD Characteristics**Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 1.8\text{ V to }3.6\text{ V}$ ,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0\text{ V}$ ,  $T_a = -40\text{ to }+105^\circ\text{C}$ 

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit	Test Conditions	
LCD output voltage variation range	$V_{L1}$	C1 to C4 connected	VLCD = 04h	0.9	1.0	1.08	V	
			VLCD = 05h	0.95	1.05	1.13	V	
			VLCD = 06h	1	1.1	1.18	V	
			VLCD = 07h	1.05	1.15	1.23	V	
			VLCD = 08h	1.1	1.2	1.28	V	
			VLCD = 09h	1.15	1.25	1.33	V	
			VLCD = 0Ah	1.2	1.3	1.38	V	
Doubler output voltage	$V_{L2}$	C1 to C5 connected	$2V_{L1} - 0.08$	$2V_{L1}$	$2V_{L1}$	V		
Tripler output voltage	$V_{L3}$	C1 to C5 connected	$3V_{L1} - 0.12$	$3V_{L1}$	$3V_{L1}$	V		
Quadruply output voltage	$V_{L4}$	C1 to C5 connected	$4V_{L1} - 0.16$	$4V_{L1}$	$4V_{L1}$	V		
Reference voltage setup time*1	$t_{VL1S}$		5	—	—	ms		
Voltage boost wait time*2	$t_{VLWT}$	C1 to C5 connected	500	—	—	ms		

Note 1. This is the required wait time from when the reference voltage is specified by the VLCD register (or when the internal voltage boosting method is selected (LCDM0.MDSET1 and MDSET0 = 01b) if the default reference voltage value is used) until voltage boosting starts (VLCON = 1).

Note 2. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

## 42.9.3 Capacitor Split Method

**Table 42.54 Capacitor Split Method**Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 2.2\text{ to }3.6\text{ V}$ ,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0\text{ V}$ ,  $T_a = -40\text{ to }+105^\circ\text{C}$ 

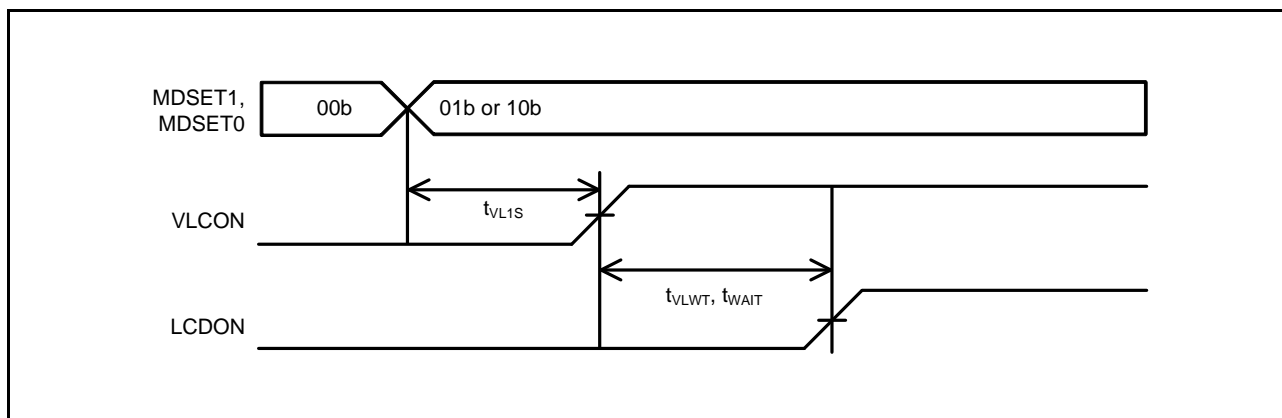
Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
External capacitance connected between CAPH and CAPL pins	C1	0.33	0.47	0.61	$\mu\text{F}$	
External capacitor connected to $V_{L1}$ pin	C2	0.33	0.47	0.61	$\mu\text{F}$	
External capacitor connected to $V_{L2}$ pin	C3	0.33	0.47	0.61	$\mu\text{F}$	
External capacitor connected to $V_{L3}$ pin	C4	0.33	0.47	0.61	$\mu\text{F}$	
External capacitor connected to $V_{L4}$ pin	C5	0.33	0.47	0.61	$\mu\text{F}$	

## (1) 1/3 Bias Method

**Table 42.55 Capacitor Split Method LCD Characteristics**Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 2.2\text{ to }3.6\text{ V}$ ,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0\text{ V}$ ,  $T_a = -40\text{ to }+105^\circ\text{C}$ 

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit	Test Conditions
$V_{L4}$ voltage*1	$V_{L4}$	C1 to C4 connected	—	VCC	—	V	
$V_{L2}$ voltage*1	$V_{L2}$	C1 to C4 connected	$2/3V_{L4} - 0.07$	$2/3V_{L4}$	$2/3V_{L4} + 0.07$	V	
$V_{L1}$ voltage**1	$V_{L1}$	C1 to C4 connected	$1/3V_{L4} - 0.08$	$2/3V_{L4}$	$2/3V_{L4} + 0.08$	V	
Capacitor split wait time*1	$t_{WAIT}$		100	—	—	ms	

Note 1. This is the wait time from when voltage bucking is started (VLCON = 1) until display is enabled (LCDON = 1).



**Figure 42.61 LCD Reference Voltage Setup Time, Voltage Boosting Wait Time, and Capacitor Split Wait Time**

### 42.10 CTSU Characteristics

**Table 42.56 CTSU Characteristics**

Conditions: VCC = AVCC0 = VCC\_USB = 1.8 to 3.6 V, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
External capacitance connected to TSCAP pin	C <sub>tscap</sub>	2.8	3.3	4	nF	
TS pin capacitive load	C <sub>base</sub>	—	—	50	pF	

## 42.11 Power-On Reset Circuit and Voltage Detection Circuit Characteristics

**Table 42.57 Power-On Reset Circuit and Voltage Detection Circuit Characteristics (1)**Conditions: VCC = AVCC0 = VCC\_USB, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Voltage detection level	Power-on reset (POR)	V <sub>POR</sub>	1.35	1.50	1.65	V	Figure 42.62, Figure 42.63
	Voltage detection circuit (LVD1)*1	V <sub>det1_4</sub>	3.00	3.10	3.20	V	Figure 42.64 At falling edge VCC
		V <sub>det1_5</sub>	2.91	3.00	3.09		
		V <sub>det1_6</sub>	2.81	2.90	2.99		
		V <sub>det1_7</sub>	2.70	2.79	2.88		
		V <sub>det1_8</sub>	2.60	2.68	2.76		
		V <sub>det1_9</sub>	2.50	2.58	2.66		
		V <sub>det1_A</sub>	2.40	2.48	2.56		
		V <sub>det1_B</sub>	1.99	2.06	2.13		
		V <sub>det1_C</sub>	1.90	1.96	2.02		
V <sub>det1_D</sub>	1.80	1.86	1.92				

Note: These characteristics apply when noise is not superimposed on the power supply. When a setting is made so that the voltage detection level overlaps with that of the voltage detection circuit (LVD2), it cannot be specified which of LVD1 and LVD2 is used for voltage detection.

Note 1. n in the symbol Vdet1\_n denotes the value of the LVDLVLR.LVD1LVL[3:0] bits.

**Table 42.58 Power-On Reset Circuit and Voltage Detection Circuit Characteristics (2)**Conditions: VCC = AVCC0 = VCC\_USB, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Voltage detection level	Voltage detection circuit (LVD2)*1	V <sub>det2_0</sub>	2.71	2.90	3.09	V	Figure 42.65 At falling edge VCC
		V <sub>det2_1</sub>	2.43	2.60	2.77		
		V <sub>det2_2</sub>	1.87	2.00	2.13		
		V <sub>det2_3</sub> *2	1.69	1.80	1.91		
Wait time after power-on reset cancellation	At normal startup*3	t <sub>POR</sub>	—	9.1	—	ms	Figure 42.63
	During fast startup time*4	t <sub>POR</sub>	—	1.6	—		
Wait time after voltage monitoring 1 reset cancellation	Power-on voltage monitoring 1 reset disabled*3	t <sub>LVD1</sub>	—	568	—	μs	Figure 42.64
	Power-on voltage monitoring 1 reset enabled*4		—	100	—		
Wait time after voltage monitoring 2 reset cancellation	t <sub>LVD2</sub>	—	100	—	μs	Figure 42.65	
Response delay time	t <sub>det</sub>	—	—	350	μs	Figure 42.62	
Minimum VCC down time*5	t <sub>VOFF</sub>	350	—	—	μs	Figure 42.62, VCC = 1.0 V or above	
Power-on reset enable time	t <sub>W</sub> (POR)	1	—	—	ms	Figure 42.63, VCC = below 1.0 V	
LVD operation stabilization time (after LVD is enabled)	T <sub>d</sub> (E-A)	—	—	300	μs	Figure 42.64, Figure 42.65	
Hysteresis width (LVD1 and LVD2)	V <sub>LVH</sub>	—	70	—	mV	Vdet1_4 selected	
		—	60	—		Vdet1_5 to 9, LVD2 selected	
		—	50	—		When selection is from among Vdet1_A to B.	
		—	40	—		When selection is from among Vdet1_C to D.	

Note: These characteristics apply when noise is not superimposed on the power supply. When a setting is made so that the voltage detection level overlaps with that of the voltage detection circuit (LVD1), it cannot be specified which of LVD1 and LVD2 is used for voltage detection.

Note 1. n in the symbol Vdet2\_n denotes the value of the LVDLVLR.LVD2LVL[3:0] bits.

Note 2. Vdet2\_3 selection can be used only when the CMPA2 pin input voltage is selected and cannot be used when the power supply voltage (VCC) is selected.

Note 3. When OFS1.(STUPLVD1REN, FASTSTUP) = 11b.

Note 4. When OFS1.(STUPLVD1REN, FASTSTUP) ≠ 11b.

Note 5. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V<sub>POR</sub>, V<sub>det0</sub>, V<sub>det1</sub>, and V<sub>det2</sub> for the POR/LVD.

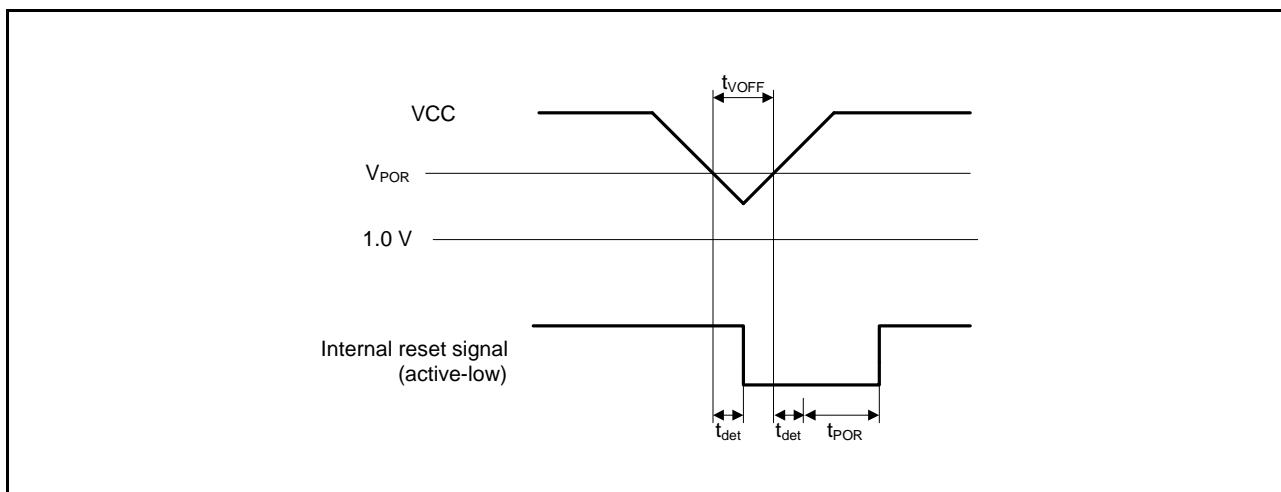
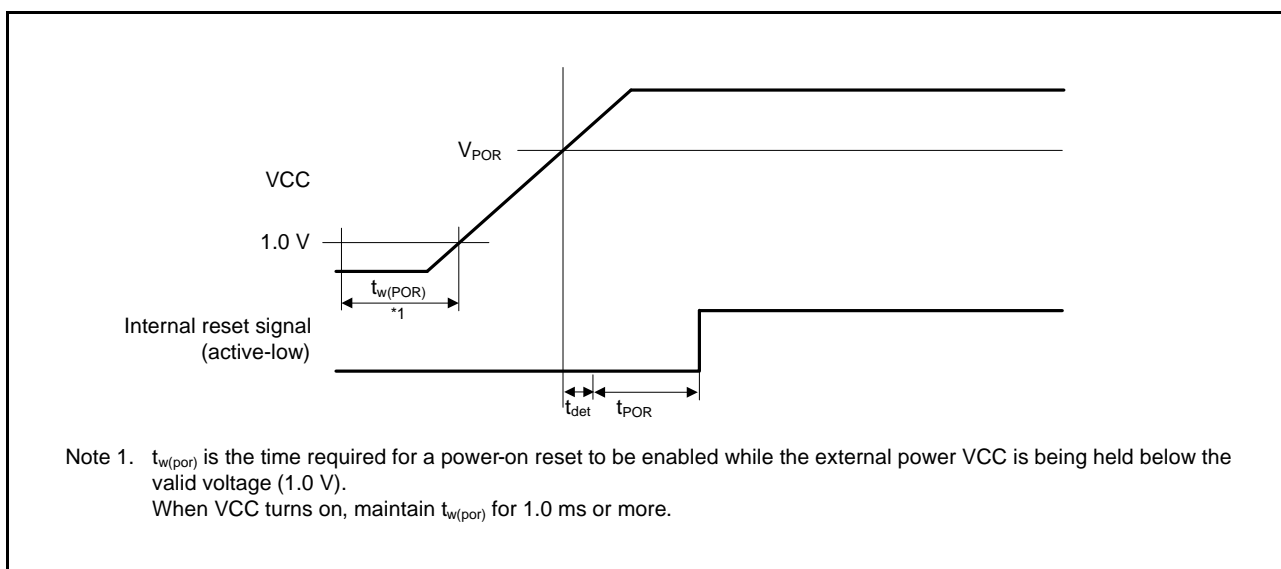


Figure 42.62 Voltage Detection Reset Timing



Note 1.  $t_{w(POR)}$  is the time required for a power-on reset to be enabled while the external power VCC is being held below the valid voltage (1.0 V).  
When VCC turns on, maintain  $t_{w(POR)}$  for 1.0 ms or more.

Figure 42.63 Power-On Reset Timing

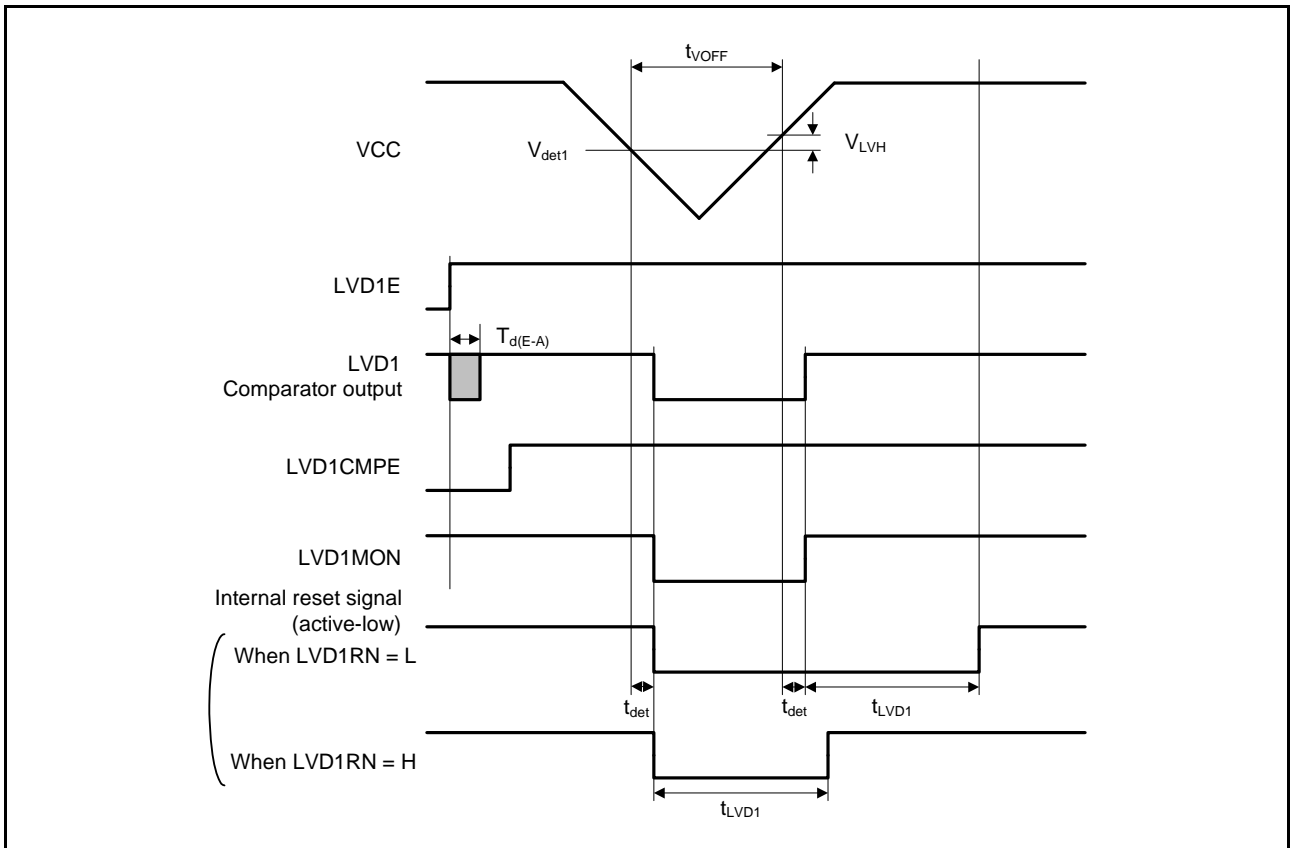


Figure 42.64 Voltage Detection Circuit Timing ( $V_{det1}$ )

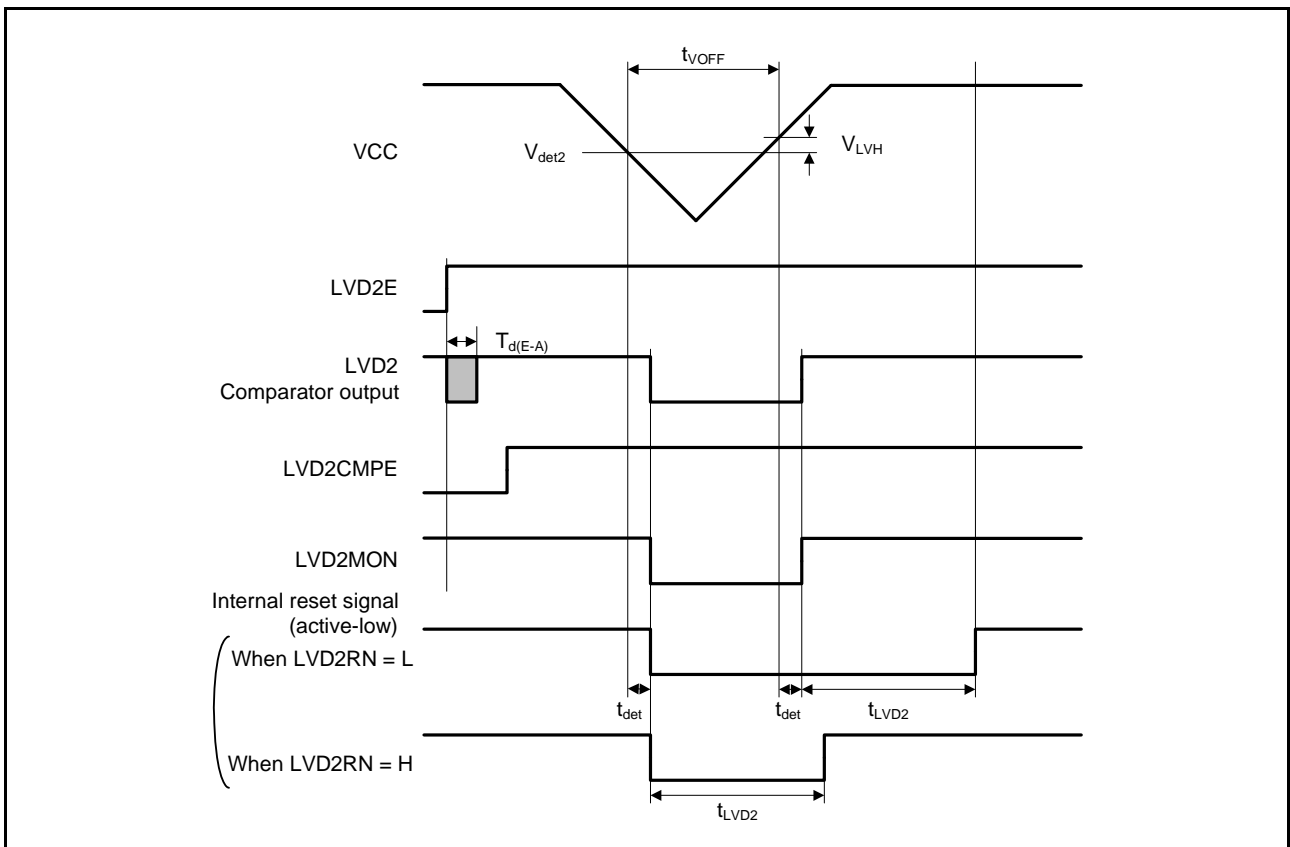


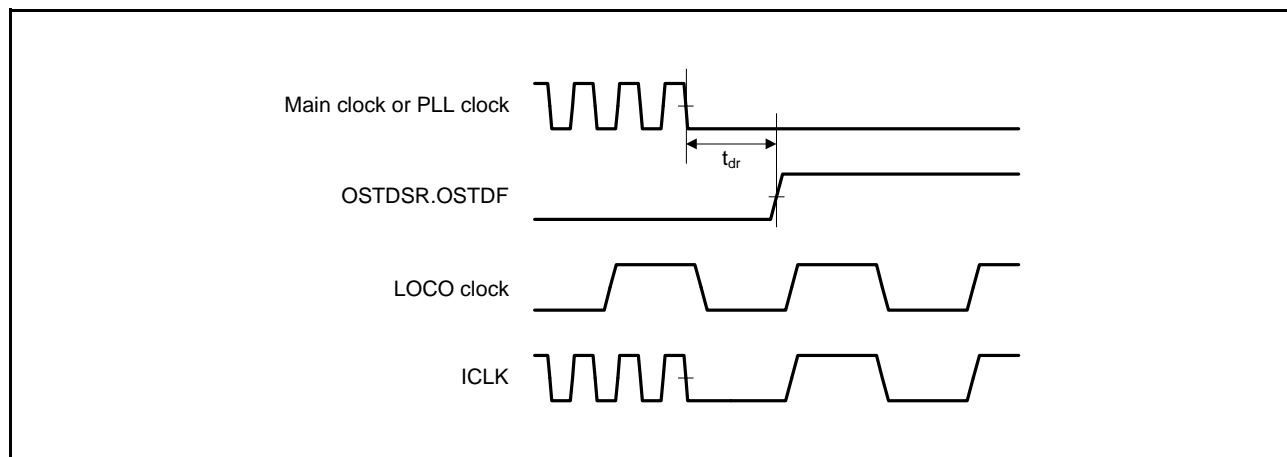
Figure 42.65 Voltage Detection Circuit Timing ( $V_{det2}$ )

42.12 Oscillation Stop Detection Timing

**Table 42.59 Oscillation Stop Detection Circuit Characteristics**

Conditions: VCC = AVCC0 = VCC\_USB = 1.8 to 3.6 V, VSS = AVSS0 = VSS\_USB = 0 V, T<sub>a</sub> = -40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t <sub>dr</sub>	—	—	1	ms	Figure 42.66



**Figure 42.66 Oscillation Stop Detection Timing**

## 42.13 ROM (Flash Memory for Code Storage) Characteristics

**Table 42.60 ROM (Flash Memory for Code Storage) Characteristics (1)**

Item		Symbol	Min.	Typ.	Max.	Unit	Conditions
Reprogramming/erasure cycle*1		$N_{PEC}$	1000	—	—	Times	
Data hold time	After 1000 times of $N_{PEC}$	$t_{DRP}$	20*2, *3	—	—	Year	$T_a = +85^\circ\text{C}$

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 4-byte programming is performed 256 times for different addresses in 1-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. Characteristic when using the flash memory programmer.

Note 3. This result is obtained from reliability testing.

**Table 42.61 ROM (Flash Memory for Code Storage) Characteristics (2)**

High-speed operating mode Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 2.7$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V

Temperature range for the programming/erasure operation:  $T_a = -40$  to  $+105^\circ\text{C}$

Item		Symbol	FCLK = 1 MHz			FCLK = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	4-byte	$t_{P4}$	—	103	931	—	52	489	$\mu\text{s}$
Erasure time	1-Kbyte	$t_{E1K}$	—	8.23	267	—	5.48	214	ms
Blank check time	4-byte	$t_{BC4}$	—	—	48	—	—	15.9	$\mu\text{s}$
	1-Kbyte	$t_{BC1K}$	—	—	1.58	—	—	0.127	ms
Erase operation forcible stop time		$t_{SED}$	—	—	21.6	—	—	12.8	$\mu\text{s}$
Start-up area switching setting time		$t_{SAS}$	—	12.6	543	—	6.16	432	ms
Access window time		$t_{AWS}$	—	12.6	543	—	6.16	432	ms
ROM mode transition wait time 1		$t_{DIS}$	2	—	—	2	—	—	$\mu\text{s}$
ROM mode transition wait time 2		$t_{MS}$	5	—	—	5	—	—	$\mu\text{s}$

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.

Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.

Note: The frequency accuracy of FCLK should be  $\pm 3.5\%$ . Confirm the frequency accuracy of the clock source.

**Table 42.62 ROM (Flash Memory for Code Storage) Characteristics (3)**

Middle-speed operating mode Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 1.8$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = V_{SS\_USB} = 0$  V

Temperature range for the programming/erasure operation:  $T_a = -40$  to  $+105^\circ\text{C}$

Item		Symbol	FCLK = 1 MHz			FCLK = 8 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	4-byte	$t_{P4}$	—	143	1330	—	96.8	932	$\mu\text{s}$
Erasure time	1-Kbyte	$t_{E1K}$	—	8.3	269	—	5.85	219	ms
Blank check time	4-byte	$t_{BC4}$	—	—	78	—	—	50	$\mu\text{s}$
	1-Kbyte	$t_{BC1K}$	—	—	1.61	—	—	0.369	ms
Erase operation forcible stop time		$t_{SED}$	—	—	33.6	—	—	25.6	$\mu\text{s}$
Start-up area switching setting time		$t_{SAS}$	—	13.2	549	—	7.6	445	ms
Access window time		$t_{AWS}$	—	13.2	549	—	7.6	445	ms
ROM mode transition wait time 1		$t_{DIS}$	2	—	—	2	—	—	$\mu\text{s}$
ROM mode transition wait time 2		$t_{MS}$	3	—	—	3	—	—	$\mu\text{s}$

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.

Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.

Note: The frequency accuracy of FCLK should be  $\pm 3.5\%$ . Confirm the frequency accuracy of the clock source.



## 42.14 E2 DataFlash Characteristics

**Table 42.63 E2 DataFlash Characteristics (1)**

Item		Symbol	Min.	Typ.	Max.	Unit	Conditions
Reprogramming/erasure cycle*1		$N_{DPEC}$	100000	1000000	—	Times	
Data hold time	After 10000 times of $N_{DPEC}$	$t_{DDRP}$	20*2, *3	—	—	Year	$T_a = +85^\circ\text{C}$
	After 100000 times of $N_{DPEC}$		5*2, *3	—	—	Year	
	After 1000000 times of $N_{DPEC}$		—	1*2, *3	—	Year	$T_a = +25^\circ\text{C}$

Note 1. 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 1-byte programming is performed 1000 times for different addresses in 1-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. Characteristics when using the flash memory programmer.

Note 3. These results are obtained from reliability testing.

**Table 42.64 E2 DataFlash Characteristics (2)  
: high-speed operating mode**

Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 2.7$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = USB\_V_{SS} = 0$  V

Temperature range for the programming/erasure operation:  $T_a = -40$  to  $+105^\circ\text{C}$

Item		Symbol	FCLK = 4 MHz			FCLK = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	1-byte	$t_{DP1}$	—	86	761	—	40.5	374	$\mu\text{s}$
Erasure time	1-Kbyte	$t_{DE1K}$	—	17.4	456	—	6.15	228	ms
Blank check time	1-byte	$t_{DBC1}$	—	—	48	—	—	15.9	$\mu\text{s}$
	1-Kbyte	$t_{DBC1K}$	—	—	1.58	—	—	0.127	ms
Erase operation forcible stop time		$t_{DSED}$	—	—	21.5	—	—	12.8	$\mu\text{s}$
DataFlash STOP recovery time		$t_{DSTOP}$	5	—	—	5	—	—	$\mu\text{s}$

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.

Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.

Note: The frequency accuracy of FCLK should be  $\pm 3.5\%$ . Confirm the frequency accuracy of the clock source.

**Table 42.65 E2 DataFlash Characteristics (3)  
: middle-speed operating mode**

Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = 1.8$  to  $3.6$  V,  $V_{SS} = AV_{SS0} = USB\_V_{SS} = 0$  V

Temperature range for the programming/erasure operation:  $T_a = -40$  to  $+105^\circ\text{C}$

Item		Symbol	FCLK = 4 MHz			FCLK = 8 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	1-byte	$t_{DP1}$	—	126	1160	—	85.4	818	$\mu\text{s}$
Erasure time	1-Kbyte	$t_{DE1K}$	—	17.5	457	—	7.76	259	ms
Blank check time	1-byte	$t_{DBC1}$	—	—	78	—	—	50	$\mu\text{s}$
	1-Kbyte	$t_{DBC1K}$	—	—	1.61	—	—	0.369	ms
Erase operation forcible stop time		$t_{DSED}$	—	—	33.5	—	—	25.5	$\mu\text{s}$
DataFlash STOP recovery time		$t_{DSTOP}$	720	—	—	720	—	—	ns

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.

Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.

Note: The frequency accuracy of FCLK should be  $\pm 3.5\%$ . Confirm the frequency accuracy of the clock source.

## Appendix 1. Port States in Each Processing Mode

**Table 1.1 Port States in Each Processing State (1/2)**

Port Name (Pin Name)	Reset	Software Standby Mode	
P02, P04, P07	Hi-Z	Keep-O	
P10 (IRQ6)	Hi-Z	Keep-O*1	
P11 (IRQ7)	Hi-Z	Keep-O*1	
P12 (IRQ2)	Hi-Z	Keep-O*1	
P13 (IRQ3)	Hi-Z	Keep-O*1	
P14 (OVRCURA/IRQ4)	Hi-Z	Keep-O*1, *2	
P15 (IRQ5/CLKOUT)	Hi-Z	CLKOUT selected	CLKOUT output
		Other than the above	Keep-O*1
P16 (VBUS/OVRCURB/IRQ6/RTCOUT)	Hi-Z	RTCOUT selected	RTCOUT output
		Other than the above	Keep-O*1, *2, *3
P17 (IRQ7)	Hi-Z	Keep-O*1	
P20 to P26	Hi-Z	Keep-O	
P27 (IRQ3)	Hi-Z	Keep-O*1	
P30 (IRQ0)	Hi-Z	Keep-O*1	
P31 (IRQ1)	Hi-Z	Keep-O*1	
P32 (IRQ2/RTCOUT)	Hi-Z	RTCOUT selected	RTCOUT output
		Other than the above	Keep-O*1
P35 (NMI)	Hi-Z	Keep*1	
P40 to P44, P46	Hi-Z	Keep-O	
P50 to P55	Hi-Z	Keep-O	
P56 (IRQ5)	Hi-Z	Keep-O*1	
P90 to P92	Hi-Z	Keep-O	
PA0	Hi-Z	Keep-O	
PA1 (RTCOUT)	Hi-Z	RTCOUT selected	RTCOUT output
		Other than the above	Keep-O
PA2	Hi-Z	Keep-O	
PA3 (IRQ6)	Hi-Z	Keep-O*1	
PA4 (IRQ5)	Hi-Z	Keep-O*1	
PA5	Hi-Z	Keep-O	
PA6 (IRQ3)	Hi-Z	Keep-O*1	
PA7	Hi-Z	Keep-O	
PB0 (IRQ2/RTCOUT)	Hi-Z	RTCOUT selected	RTCOUT output
		Other than the above	Keep-O*1
PB1 (IRQ4)	Hi-Z	Keep-O*1	
PB2	Hi-Z	Keep-O	
PB3 (OVRCURA)	Hi-Z	Keep-O*2	
PB4 to PB7	Hi-Z	Keep-O	
PC0 to PC3	Hi-Z	Keep-O	
PC4 (VBUS/IRQ2/CLKOUT)	Hi-Z	CLKOUT selected	CLKOUT output
		Other than the above	Keep-O*1, *2
PC5, PC6	Hi-Z	Keep-O	
PC7 (OVRCURB)	Hi-Z	Keep-O*2	
PD0 (IRQ0)	Hi-Z	Keep-O*1	

**Table 1.1 Port States in Each Processing State (2/2)**

Port Name (Pin Name)	Reset	Software Standby Mode	
PD1 (IRQ1)	Hi-Z	Keep-O*1	
PD2 (IRQ2)	Hi-Z	Keep-O*1	
PD3 (IRQ3)	Hi-Z	Keep-O*1	
PD4 (IRQ4)	Hi-Z	Keep-O*1	
PE0 (IRQ0)	Hi-Z	Keep-O*1	
PE1 (IRQ1)	Hi-Z	Keep-O*1	
PE2 (IRQ7)	Hi-Z	Keep-O*1	
PE3 (IRQ3)	Hi-Z	Keep-O*1	
PE4 (IRQ4)	Hi-Z	Keep-O*1	
PE5 (IRQ5/CMPOB1)	Hi-Z	CMPOB1 selected	CMPOB1 output
		Other than the above	Keep-O*1
PE6 (IRQ6)	Hi-Z	Keep-O*1	
PE7 (IRQ7/CMPOB0)	Hi-Z	CMPOB0 selected	CMPOB0 output
		Other than the above	Keep-O*1
PF6, PF7	Hi-Z	Keep-O	
PH7	Hi-Z	Keep	
PJ0 (DA0)	Hi-Z	DA0 output (DAOE0 = 1)	DA output retained
		Other than the above (DAOE0 = 0)	Keep-O
PJ2 (DA1)	Hi-Z	DA1 output (DAOE1 = 1)	DA output retained
		Other than the above (DAOE1 = 0)	Keep-O
PJ3	Hi-Z	Keep-O	
PJ6	Hi-Z	Keep-O	
PJ7	Hi-Z	Keep-O	

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. (pull-up and open-drain settings are also retained.)

Hi-Z: High-impedance

Note: The LCD output is retained when the LCD controller/driver pin functions (COM0 to COM7 and SEG00 to SEG24) are set by the PmnPFS.PSEL[4:0] bits and the sub-clock oscillator is selected by the LCDSCLKCR.LCDSCLKSEL bit.

Note 1. Input is enabled if the pin is specified as the software standby mode canceling source while it is used as an external interrupt pin.

Note 2. Input is enabled while the pin is used as the USB pin (VBUS/OVRCURA/OVRCURB).

Note 3. Do not input or output a high-level signal in software standby mode if the pin is selected as a peripheral function or external interrupt pin.

## Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings has been displayed in “Packages” on Renesas Electronics Corporation website.

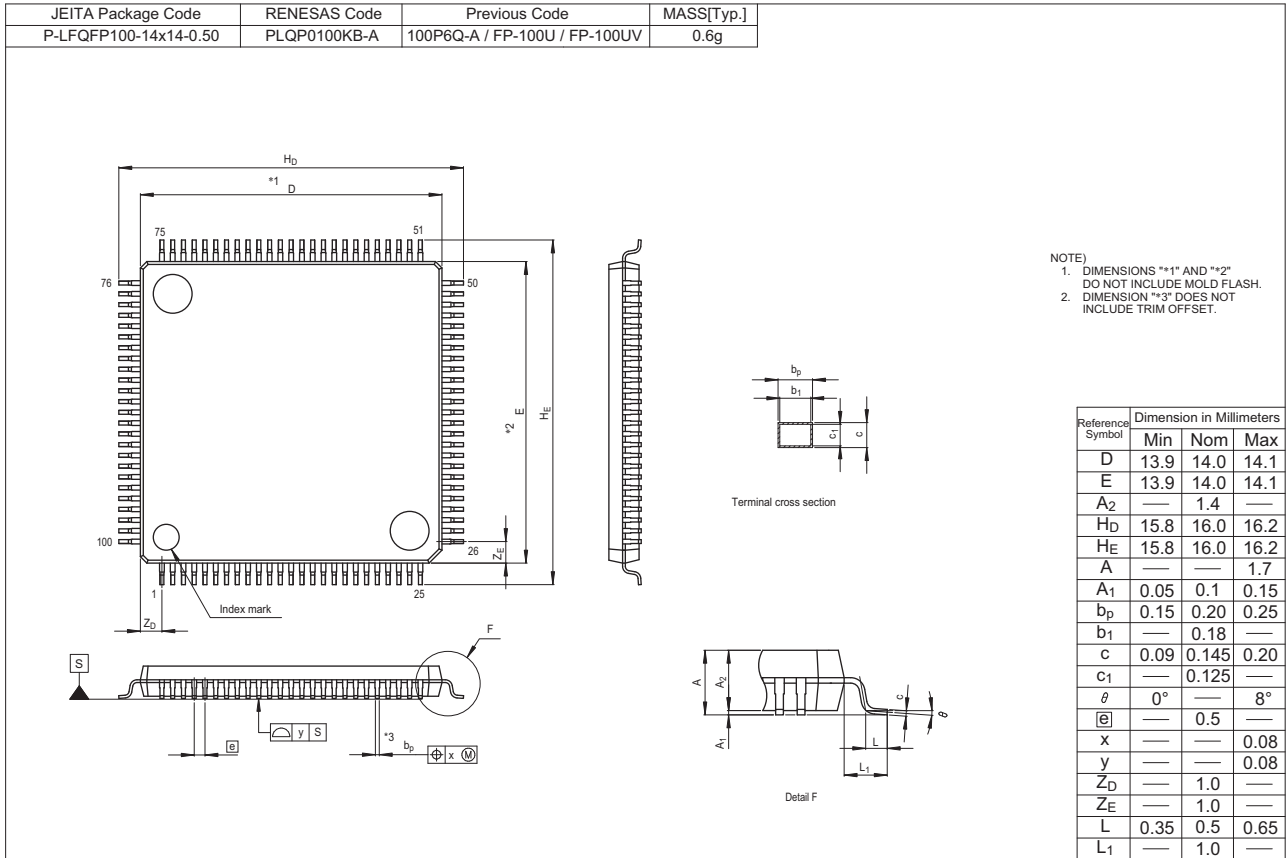


Figure A 100-Pin LFQFP (PLQP0100KB-A)

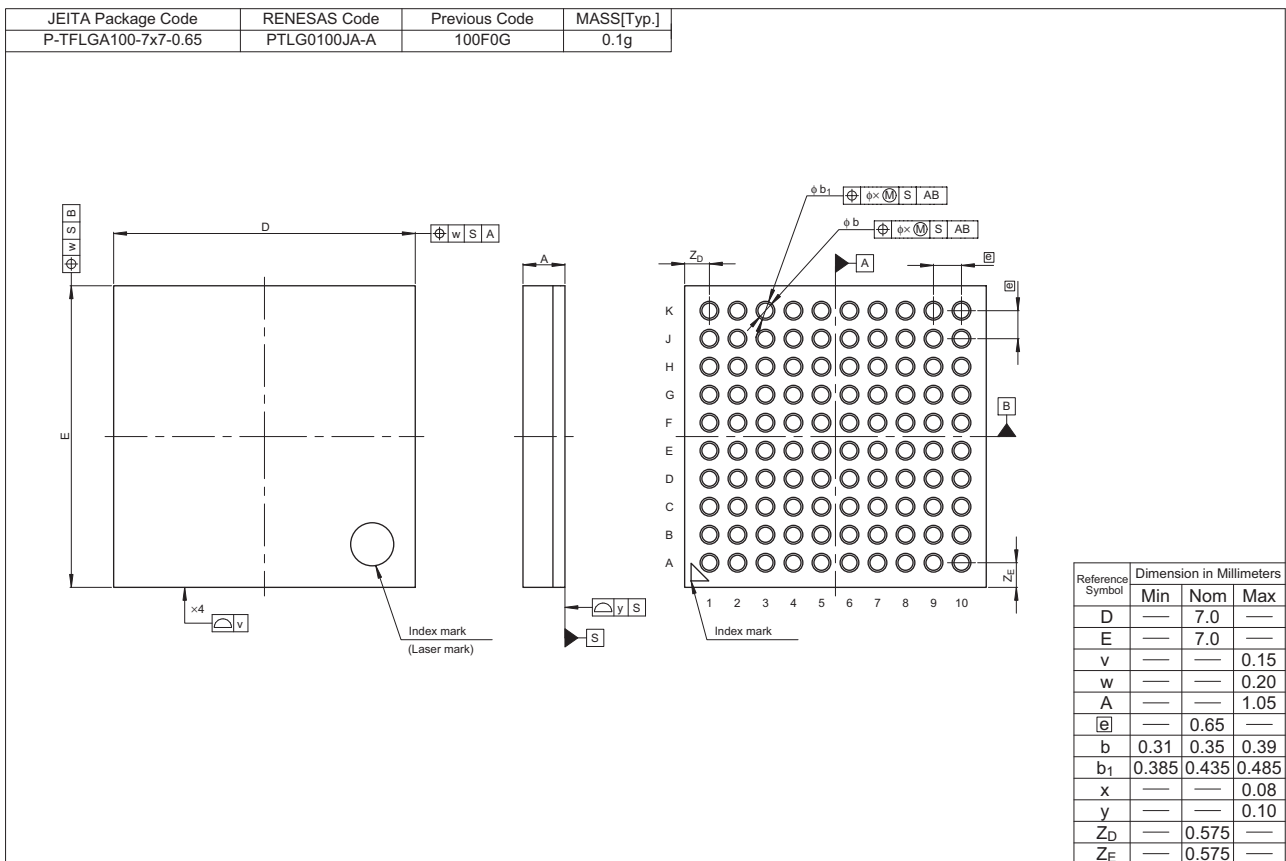


Figure B 100-Pin TFLGA (PTLG0100JA-A)

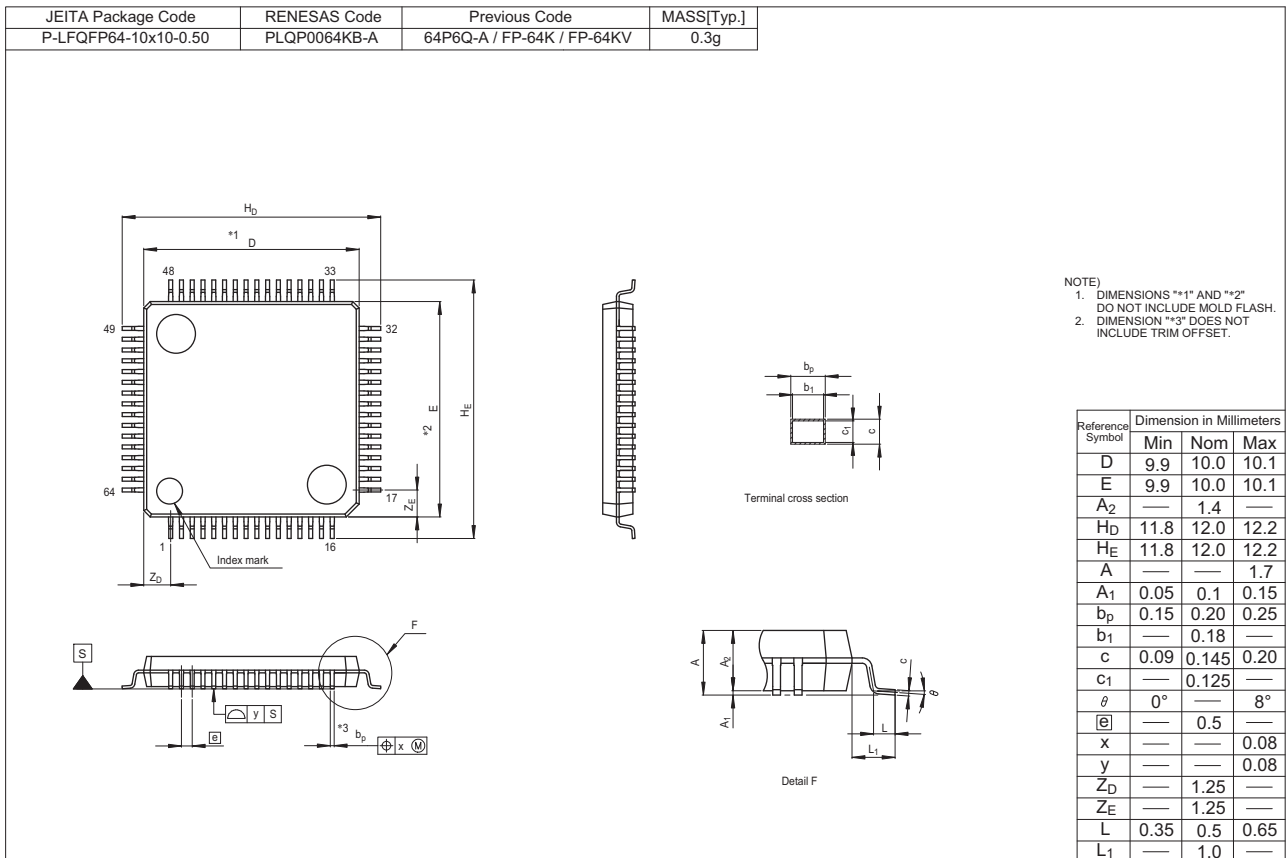


Figure C 64-Pin LFQFP (PLQP0064KB-A)

REVISION HISTORY	RX113 Group User's Manual: Hardware
------------------	-------------------------------------

Rev.	Date	Description	
		Page	Summary
1.00	May 15, 2014	—	First edition, issued
1.01	Aug 25, 2014	7. Option-Setting Memory	
		136	7.2.1 Option Function Select Register 0 (OFS0): Bit function table changed
		17. Event Link Controller (ELC)	
		335	Table 17.5 Operations of Modules When Event is Input: POE added
		18. I/O Ports	
		363	Table 18.5 Handling of Unused Pins, changed
		19. Multi-Function Pin Controller (MPC)	
		387	Table 19.13 Register Settings for I/O Pin Functions in 100-Pin, changed
		388	Table 19.14 Register Settings for I/O Pin Functions in 64-Pin, changed
		20. Multi-Function Timer Pulse Unit 2 (MTU2a)	
		397 to 584	Terms changed: Counter clock → count clock, preset → set, synchronous presetting → synchronous setting
		496	Figure 20.46 Example of Waveform Output in Complementary PWM Mode (1), changed
		497	Figure 20.47 Example of Waveform Output in Complementary PWM Mode (2) and Figure 20.48 Example of Waveform Output in Complementary PWM Mode (3), changed
		498	Figure 20.49 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1) and Figure 20.50 Example of 0% and 100% Waveform Output in Complementary PWM Mode (2), changed
		499	Figure 20.51 Example of 0% and 100% Waveform Output in Complementary PWM Mode (3) and Figure 20.52 Example of 0% and 100% Waveform Output in Complementary PWM Mode (4), changed
		500	Figure 20.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5), changed
		513	Figure 20.71 Example of Operation When Buffer Transfer is Linked with Interrupt Skipping (TBTER.BTE[1:0] = 10b), changed
		514	Figure 20.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period, changed
		21. Port Output Enable 2 (POE2a)	
		601	21.5.3 Specifying Pins Corresponding to the MTU, changed
		22. 8-Bit Timer (TMR)	
		602	Table 22.1 Specifications of TMR: Event link function (output) and event link function (input) changed
		603	Table 22.2 TMR Functions, changed
		25. Independent Watchdog Timer (IWDTCa)	
		689	25.3.3 Refresh Operation: [Sample refreshing timings] changed
		690	Figure 25.6 IWDTC Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b), changed
		26. USB 2.0 Host/Function Module (USBc)	
		702	26.2.4 CFIFO Port Register (CFIFO), D0FIFO Port Register (D0FIFO), D1FIFO Port Register (D1FIFO), changed
		703	Table 26.6 Endian Operation in 8-Bit Access, deleted
		716	26.2.13 Interrupt Status Register 0 (INTSTS0): Note 5 in bit function table changed
		27. Serial Communications Interface (SCIE, SCIf)	
		807	Figure 27.4 Block Diagram of SCIf (SC112), changed
821 to 824	27.2.7 Serial Status Register (SSR) (1) Non-Smart Card Interface Mode (SCMR.SMIF = 0): PER, FER, and ORER flag descriptions changed (2) Smart Card Interface Mode (SCMR.SMIF = 1): PER, ERS, and ORER flag descriptions changed		
886	27.6.5 SCI Initialization (Smart Card Interface Mode), changed		
28. IrDA Interface			
939	28.2.1 IrDA Control Register (IRCR): IRCKS[2:0] bit description changed		
940	28.3.1 IrDA Interface Setting Procedure, deleted		
942	28.4.2 SC15 Setting and 28.4.4 Notes on IrDA Initial Setting/Resetting, changed		

Rev.	Date	Description	
		Page	Summary
1.01	Aug 25, 2014	29. I <sup>2</sup> C Bus Interface (RIIC)	
		955, 956	29.2.5 I <sup>2</sup> C Bus Mode Register 3 (ICMR3): Bit function table changed
		975	29.2.18 Timeout Internal Counter (TMOCNT), changed
		1004	29.9.1 Master Arbitration-Lost Detection (MALE Bit), changed
		1014	29.12.1 SMBus Timeout Measurement, changed
		1015	Figure 29.41 SMBus Timeout Measurement, changed
		1016	Table 29.7 Interrupt Sources, changed
		30. Serial Sound Interface (SSI)	
		1022, 1023	30.2.1 Control Register (SSICR): Bit function table changed
		1026	30.2.2 Status Register (SSISR): Bit function table changed
		1028	30.2.3 FIFO Control Register (SSIFCR), changed
		1043	Figure 30.18 Transmission Using the DTC: Note 1 added
		1044	Figure 30.19 Transmission Using Interrupts: Note 1 added
		31. Serial Peripheral Interface (RSPI)	
		1057	31.2.4 RSPI Status Register (SPSR): Bit function table changed
		1064	31.2.8 RSPI Bit Rate Register (SPBR): Register description changed
		1113	Figure 31.37 Example of Initialization Flowchart in Slave Mode (SPI Operation), changed
		34. Capacitive Touch Sensing Unit (CTSU)	
		1194	34.2.13 CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register (CTSUSSC): Bit function table changed
		1195	34.2.14 CTSU Sensor Offset Register 0 (CTSUSO0): Bit function table changed
		1196	34.2.15 CTSU Sensor Offset Register 1 (CTSUSO1): Bit function table changed
		1199	34.2.18 CTSU Error Status Register (CTSUERRS): CTSUICOMP bit description changed
		1207	Figure 34.14 Software Flow and Operation Example of Self-Capacitance Multi-Scan Mode, changed
		1209	Figure 34.16 Software Flowchart and Operation Example of Mutual Capacitance Simple Scan Mode, changed
		1212	Figure 34.18 Software Flowchart and Operation Example of Mutual Capacitance Full Scan Mode, changed
		35. 12-Bit A/D Converter (S12ADb)	
		1244	35.2.17 A/D High-Side Reference Voltage Control Register (ADHVREFCNT): OCSVSEL bit description changed
		36. 12-Bit D/A Converter (R12DAA)	
		1268	36.2.5 D/A VREF Control Register (DAVREFCR): REF[2:0] bit description changed
		38. Comparator B (CMPBa)	
		1287	38.2.7 Comparator B Reference Input Voltage Select Register (CPBREF): Note 3 changed and Note 4 added
		41. Flash Memory	
		1304	Table 41.1 Flash Memory Specifications: Software commands changed
		1307	41.4.1 E2 DataFlash Control Register (DFLCTL), changed
		1308	41.4.2 Flash P/E Mode Entry Register (FENTRYR): FENTRY0 and FENTRYD bit descriptions changed
		1311	41.4.6 Flash Initial Setting Register (FISR): Note in PCKA[4:0] bit description deleted
		1313	41.4.7 Flash Reset Register (FRESETR): Register description changed
		1314, 1315	41.4.9 Flash Control Register (FCR): Bit function table and CMD[3:0] and DRC bit descriptions changed
		1315	41.4.10 Flash Extra Area Control Register (FEXCR): CMD[2:0] bit description changed
		1316	41.4.11 Flash Processing Start Address Register H (FSARH): Register description changed
		1317	41.4.12 Flash Processing Start Address Register L (FSARL) and 41.4.13 Flash Processing End Address Register H (FEARH): Register descriptions changed
		1318	41.4.14 Flash Processing End Address Register L (FEARL), 41.4.15 Flash Read Buffer Register H (FRBH), and 41.4.16 Flash Read Buffer Register L (FRBL): Register descriptions changed
		1320	41.4.19 Flash Status Register 0 (FSTATR0): Bit function table changed
		1322	41.4.20 Flash Status Register 1 (FSTATR1): DRRDY flag description added, FRDY flag description changed
		1323	41.4.21 Flash Error Address Monitor Register H (FEAMH) and 41.4.22 Flash Error Address Monitor Register L (FEAML): Register descriptions changed
		1325	41.4.26 Unique ID Register n (UIDRn) (n = 0 to 31), changed
		1326	Figure 41.3 Overview of the Start-Up Program Protection, changed



Rev.	Date	Description			
		Page	Summary		
1.01	Aug 25, 2014	1327	41.6 Area Protection and Figure 41.4 Area Protection Overview (Set Blocks 4 to 6 as the Access Window), changed		
		1328, 1329	41.7.1 Figure Sequencer Modes and Figure 41.5 Mode Transitions of the Flash Memory, changed 41.7.1.1 E2 DataFlash Access Disabled mode, added 41.7.1.2 Read Mode, changed		
		1329	41.7.1.3 P/E Modes and 41.7.3 Software Command Usage (1) Switching to ROM P/E Mode, changed		
		1330	Figure 41.6 Procedure for Transition to ROM P/E Mode, changed		
		1331	Figure 41.7 Procedure for Transition from ROM/E2 DataFlash Read Mode to E2 DataFlash P/E Mode, changed		
		1332	Figure 41.8 Procedure for Transition to ROM Read Mode, changed		
		1333	Figure 41.9 Procedure for Transition from E2 DataFlash P/E Mode to ROM/E2 DataFlash Read Mode, changed		
		1334	Figure 41.10 ROM Programming Procedure, changed		
		1335	Figure 41.11 E2 DataFlash Programming Procedure, changed		
		1336	Figure 41.12 ROM Block Erase Procedure, changed		
		1337	Figure 41.13 E2 DataFlash Block Erase Procedure, changed		
		1338	Figure 41.14 ROM Blank Check Procedure, changed		
		1339	Figure 41.15 E2 DataFlash Blank Check Procedure, changed		
		1340	Figure 41.16 Procedure for the Start-Up Area Information Program/Access Window Information Program, changed		
		1341	Figure 41.17 Simple Flowchart of the Procedure for Unique ID Read, changed		
		1342	(6) Forced Stop of Software Commands and Figure 41.18 Procedure for Forced Stop of Software Commands, added		
		1352	Table 41.11 Boot Mode ID Code Protection Specifications, changed		
		1353	Figure 41.25 Authentication for Boot Mode ID Code Protection and (2) ID Code 1 to ID Code 15, changed		
		1354	Table 41.12 On-Chip Debugging Emulator ID Code Protection Specifications, changed		
		1362	41.10.5.2 Operating Frequency Select: Section number changed		
		1363	41.10.5.3 Program/Erase Status Transition: Section number changed		
		1364	41.10.6 ID Code Authentication Command: Section number changed		
		1365	41.10.6.1 ID Code Check, changed		
		1365	41.10.6.2 Erase Ready, changed		
		1374	Figure 41.27 Transmit/Receive Data for Bit Rate Automatic Adjustment, changed		
		1384	41.10.9.11 Set the Access Window in the User Area, changed		
		1385	41.11 Rewriting by Self-Programming, 41.11.1 Overview, and Figure 41.39 Self-Programming Overview, changed		
		1386	41.12 Usage Notes (1) Access the Block Where Erase Operation is Forcibly Stopped and (2) Processing After Forced Stop of Erase Operation, changed		
		42. Electrical Characteristics			
		1418	Table 42.33 Timing of On-Chip Peripheral Modules (4): Note 2 deleted		
		1419	Table 42.34 Timing of On-Chip Peripheral Modules (5): Note 1 deleted Table 42.35 Timing of On-Chip Peripheral Modules (6) changed		
		1432	Table 42.40 A/D Conversion Characteristics (4): Analog input capacity deleted		
		1443	Table 42.56 CTSU Characteristics, changed		
1448	Table 42.61 ROM (Flash Memory for Code Storage) Characteristics (2) and Table 42.62 ROM (Flash Memory for Code Storage) Characteristics (3), changed				
1449	Table 42.64 E2 DataFlash Characteristics (2): high-speed operating mode changed				

---

RX113 Group User's Manual: Hardware

Publication Date: Rev.1.00 May 15, 2014  
Rev.1.01 Aug 25, 2014

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.**2801 Scott Boulevard Santa Clara, CA 95050-2549, 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: +44-1628-585-100, Fax: +44-1628-585-900**Renesas Electronics Europe GmbH**Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-6503-0, Fax: +49-211-6503-1327**Renesas Electronics (China) Co., Ltd.**Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679**Renesas Electronics (Shanghai) Co., Ltd.**Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333  
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999**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: +852-2265-6688, Fax: +852 2686-9022/9044**Renesas Electronics Taiwan Co., Ltd.**13F, No. 363, Fu Shing North Road, Taipei 10543, 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.**12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

RX113 Group