

RX62N Group, RX621 Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family / RX600 Series

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. 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 categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - “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.
 - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
 - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. 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.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

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

4. Clock Signals

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

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

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

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

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 RX62N Group, RX621 Group. Before using any of the documents, please visit our web site to verify that you have the most up-to-date available version of the document.

Document Type	Contents	Document Title	Document No.
Short Sheet	Overview of hardware	—	—
Data Sheet	Overview of hardware and electrical characteristics	RX62N Group, RX621 Group Data sheet	R01DS0052EJ
User's manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	RX62N Group, RX621 Group User's manual: Hardware	This User's manual
User's manual: Software	Detailed descriptions of the CPU and instruction set	RX Family Series User's manual: Software	RJJ09B0465
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 xxxx xxxxxh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	...

Value after reset x 0 0 0 0 0 0 0

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

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

3. List of Abbreviations and Acronyms

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

Contents

1.	Features.....	44
1.	Overview.....	45
1.1	Outline of Specifications.....	45
1.2	List of Products.....	50
1.3	Block Diagram.....	52
1.4	Pin Assignments.....	53
1.5	Pin Functions.....	83
2.	CPU.....	90
2.1	Features.....	90
2.2	Register Set of the CPU.....	91
2.2.1	General-Purpose Registers (R0 to R15).....	92
2.2.2	Control Registers.....	92
2.2.2.1	Interrupt Stack Pointer (ISP)/User Stack Pointer (USP).....	93
2.2.2.2	Interrupt Table Register (INTB).....	93
2.2.2.3	Program Counter (PC).....	93
2.2.2.4	Processor Status Word (PSW).....	94
2.2.2.5	Backup PC (BPC).....	96
2.2.2.6	Backup PSW (BPSW).....	96
2.2.2.7	Fast Interrupt Vector Register (FINTV).....	96
2.2.2.8	Floating-Point Status Word (FPSW).....	97
2.2.3	Accumulator (ACC).....	99
2.3	Processor Mode.....	100
2.3.1	Supervisor Mode.....	100
2.3.2	User Mode.....	100
2.3.3	Privileged Instruction.....	100
2.3.4	Switching Between Processor Modes.....	100
2.4	Data Types.....	101
2.4.1	Integer.....	101
2.4.2	Floating-Point.....	101
2.4.3	Bitwise Operations.....	102
2.4.4	Strings.....	102
2.5	Endian.....	103
2.5.1	Switching the Endian.....	103
2.5.2	Access to I/O Registers.....	107
2.5.3	Notes on Access to I/O Registers.....	107
2.5.4	Data Arrangement.....	108
2.5.4.1	Data Arrangement in Registers.....	108
2.5.4.2	Data Arrangement in Memory.....	108
2.5.5	Notes on the Allocation of Instruction Codes.....	108
2.6	Vector Table.....	109
2.6.1	Fixed Vector Table.....	109

2.6.2	Relocatable Vector Table	110
2.7	Operation of Instructions.....	111
2.7.1	Data Prefetching by the RMPA Instruction and the String-Manipulation Instructions	111
2.8	Pipeline.....	112
2.8.1	Overview	112
2.8.2	Instructions and Pipeline Processing.....	113
2.8.2.1	Instructions Converted into Single Micro-Operation and Pipeline Processing	113
2.8.2.2	Instructions Converted into Multiple Micro-Operations and Pipeline Processing	115
2.8.2.3	Pipeline Basic Operation	118
2.8.3	Calculation of the Instruction Processing Time	120
2.8.4	Numbers of Cycles for Response to Interrupts.....	121
3.	Operating Modes	122
3.1	Operating Mode Types and Selection	122
3.2	Register Descriptions.....	123
3.2.1	Mode Monitor Register (MDMONR)	123
3.2.2	Mode Status Register (MDSR).....	124
3.2.3	System Control Register 0 (SYSCR0).....	125
3.2.4	System Control Register 1 (SYSCR1).....	126
3.3	Details of Operating Modes	127
3.3.1	Single-Chip Mode.....	127
3.3.2	On-Chip ROM Enabled Extended Mode.....	127
3.3.3	On-Chip ROM Disabled Extended Mode	127
3.3.4	Boot Mode	127
3.3.5	USB Boot Mode	127
3.4	Transitions of Operating Modes.....	128
3.4.1	Operating Mode Transitions According to Mode Pin Setting.....	128
3.4.2	Operating Mode Transitions According to Register Setting	129
4.	Address Space.....	130
4.1	Address Space	130
4.2	External Address Space	131
5.	I/O Registers	132
5.1	I/O Register Addresses (Address Order).....	134
5.2	I/O Register Bits.....	170
6.	Resets.....	223
6.1	Overview.....	223
6.2	Register Descriptions.....	226
6.2.1	Reset Status Register (RSTSR)	226
6.2.2	Reset Control/Status Register (RSTCSR)	226
6.2.3	IWDT Status Register (IWDTSR).....	226
6.3	Operation.....	227

6.3.1	Pin Reset	227
6.3.2	Power-On Reset	227
6.3.3	Voltage-Monitoring Reset	228
6.3.4	Deep Software Standby Reset	228
6.3.5	Independent Watchdog Timer Reset	228
6.3.6	Watchdog Timer Reset	228
6.4	Determination of Reset Generation Source	229
6.5	Usage Notes	229
6.5.1	Notes on Board Design	229
7.	Voltage Detection Circuit (LVD)	230
7.1	Overview	230
7.2	Register Descriptions	231
7.2.1	Reset Status Register (RSTSR)	231
7.2.2	Key Code Register for Low-Voltage Detection Control Register (LVDKEYR)	231
7.2.3	Low-Voltage Detection Control Register (LVDCR)	232
7.3	Voltage Detection Circuit	234
7.3.1	Voltage Monitoring Reset	234
7.3.2	Voltage Monitoring Interrupt	236
7.3.3	Cancellation of Deep Software Standby Mode by the Voltage Detection Circuit	239
8.	Clock Generation Circuit	240
8.1	Overview	240
8.2	Register Descriptions	242
8.2.1	System Clock Control Register (SCKCR)	243
8.2.2	External Bus Clock Control Register (BCKCR)	245
8.2.3	Oscillation Stop Detection Control Register (OSTDCR)	246
8.2.4	Sub-Clock Oscillator Control Register (SUBOSCCR)	247
8.3	Main Clock Oscillator	248
8.3.1	Connecting a Crystal Resonator	248
8.3.2	External Clock Input	249
8.4	Sub-Clock Oscillator	250
8.4.1	Connecting 32.768-kHz Crystal Resonator	250
8.4.2	Handling of Pins when Sub-Clock is Not Used	251
8.5	On-Chip Oscillator	251
8.6	Oscillation Stop Detection Circuit	251
8.7	Internal Oscillation Circuit	251
8.8	PLL Circuit	251
8.9	Frequency Divider	251
8.10	Internal Clock	252
8.10.1	System Clock (ICLK)	252
8.10.2	Peripheral Module Clock (PCLK)	252

8.10.3	External Bus Clock (BCLK).....	253
8.10.4	SDRAM Clock (SDCLK).....	253
8.10.5	Dedicated USB Clock (UCLK).....	253
8.10.6	Dedicated RTC Clock (SUBCLK).....	253
8.10.7	On-Chip Oscillator Clock (IWDTCLK).....	253
8.11	Oscillation Stop Detection Function.....	254
8.11.1	Detection of Oscillation Stop and Operation after the Detection.....	254
8.11.2	Oscillation Stop Detection Interrupt.....	255
8.11.3	Note on Oscillation Stop Detection Function.....	255
8.12	Usage Notes.....	256
8.12.1	Notes on Clock Generation Circuit.....	256
8.12.2	Notes on Resonator.....	257
8.12.3	Notes on Board Design.....	257
9.	Low Power Consumption.....	258
9.1	Overview.....	258
9.2	Register Descriptions.....	261
9.2.1	Standby Control Register (SBYCR).....	263
9.2.2	Module Stop Control Register A (MSTPCRA).....	265
9.2.3	Module Stop Control Register B (MSTPCRB).....	267
9.2.4	Module Stop Control Register C (MSTPCRC).....	269
9.2.5	Deep Standby Control Register (DPSBYCR).....	270
9.2.6	Deep Standby Wait Control Register (DPSWCR).....	272
9.2.7	Deep Standby Interrupt Enable Register (DPSIER).....	273
9.2.8	Deep Standby Interrupt Flag Register (DPSIFR).....	274
9.2.9	Deep Standby Interrupt Edge Register (DPSIEGR).....	276
9.2.10	Reset Status Register (RSTSR).....	277
9.2.11	Deep Standby Backup Register (DPSBKRY) (y = 0 to 31).....	279
9.3	Multi-Clock Function.....	280
9.4	Module Stop Function.....	280
9.5	Low Power Consumption Modes.....	281
9.5.1	Sleep Mode.....	281
9.5.1.1	Transition to Sleep Mode.....	281
9.5.1.2	Canceling Sleep Mode.....	281
9.5.2	All-Module Clock Stop Mode.....	282
9.5.2.1	Transition to All-Module Clock Stop Mode.....	282
9.5.2.2	Canceling All-Module Clock Stop Mode.....	283
9.5.3	Software Standby Mode.....	284
9.5.3.1	Transition to Software Standby Mode.....	284
9.5.3.2	Canceling Software Standby Mode.....	285
9.5.3.3	Setting Oscillation Settling Time after Software Standby Mode is Canceled.....	286
9.5.3.4	Example of Software Standby Mode Application.....	287

9.5.4	Deep Software Standby Mode.....	288
9.5.4.1	Transition to Deep Software Standby Mode	288
9.5.4.2	Canceling Deep Software Standby Mode	289
9.5.4.3	Pin States when Deep Software Standby Mode is Canceled	290
9.5.4.4	Setting Oscillation Settling Time after Deep Software Standby Mode is Canceled	291
9.5.4.5	Canceling Deep Software Standby Mode by USB	292
9.5.4.6	Example of Deep Software Standby Mode Application	292
9.5.4.7	Flowchart to Use Deep Software Standby Mode	294
9.6	BCLK and SDCLK Output Control.....	295
9.7	Usage Notes.....	296
9.7.1	I/O Port States.....	296
9.7.2	Module Stop State of the DMACA, DTC, EXDMAC, and EDMAC.....	296
9.7.3	On-Chip Peripheral Module Interrupts.....	296
9.7.4	Write-Access to MSTPCRA, MSTPCRB, and MSTPCRC.....	296
9.7.5	Input Buffer Control by DIRQnE Bit (n = 3 to 0).....	296
9.7.6	Conflict between Transition to Deep Software Standby Mode and Interrupt	296
9.7.7	Timing of Wait Instructions	296
10.	Exceptions	297
10.1	Types of Exceptions	297
10.1.1	Undefined Instruction Exception.....	298
10.1.2	Privileged Instruction Exception	298
10.1.3	Access Exceptions	298
10.1.4	Floating-Point Exceptions	298
10.1.5	Reset	298
10.1.6	Non-Maskable Interrupt	298
10.1.7	Interrupts.....	298
10.1.8	Unconditional Trap.....	298
10.2	Exception Handling Procedure	299
10.3	Acceptance of Exceptions	301
10.3.1	Timing of Acceptance and Saved PC Values.....	301
10.3.2	Vector and Site for Saving the Values in the PC and PSW.....	302
10.4	Hardware Processing for Accepting and Returning from Exceptions.....	303
10.5	Hardware Pre-Processing.....	304
10.5.1	Undefined Instruction Exception.....	304
10.5.2	Privileged Instruction Exception	304
10.5.3	Floating-Point Exceptions	304
10.5.4	Access Exceptions	304
10.5.5	Reset	304
10.5.6	Non-Maskable Interrupt	304
10.5.7	Interrupts.....	306
10.5.8	Unconditional Trap.....	306

10.6	Return from Exception Processing Routines	307
10.7	Order of Priority for Exceptions.....	307
11.	Interrupt Control Unit (ICUa).....	308
11.1	Overview	308
11.2	Register Descriptions.....	310
11.2.1	Interrupt Request Register i (IRi) (i = interrupt vector number).....	320
11.2.2	Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh)	322
11.2.3	Interrupt Priority Register m (IPRm) (m = 00h to 8Fh)	323
11.2.4	Fast Interrupt Register (FIR)	324
11.2.5	Software Interrupt Activation Register (SWINTR).....	325
11.2.6	DTC Activation Enable Register n (DTCERn) (n = interrupt vector number)	326
11.2.7	DMACA Activation Source Select Register n (DMRSRn) (n = DMACA channel number) ...	327
11.2.8	IRQ Control Register n (IRQCRn) (n = 0 to 15).....	328
11.2.9	Non-Maskable Interrupt Status Register (NMISR)	329
11.2.10	Non-Maskable Interrupt Enable Register (NMIER)	331
11.2.11	Non-Maskable Interrupt Clear Register (NMICLR)	332
11.2.12	NMI Pin Interrupt Control Register (NMICR).....	333
11.3	Vector Table	334
11.3.1	Interrupt Vector Table	334
11.3.2	Fast Interrupt Vector.....	339
11.3.3	Non-maskable Interrupt Vector	339
11.4	Interrupt Operation.....	340
11.4.1	Detecting Interrupts	340
11.4.1.1	Operation of Status Flags for Edge-Detected Interrupts	340
11.4.1.2	Operation of Status Flags for Level-Detected Interrupts	342
11.4.2	Enabling and Disabling Interrupt Sources.....	343
11.4.3	Selecting Interrupt Request Destinations	344
11.4.4	Determining Priority.....	346
11.4.5	Fast Interrupt.....	346
11.4.6	External Pin Interrupts.....	346
11.5	Non-maskable Interrupt Operation	347
11.6	Return from Power-Down States	348
11.6.1	Return from Sleep Mode	348
11.6.2	Return from All-Module Clock Stop Mode	348
11.6.3	Return from Software Standby Mode.....	348
11.7	Usage Notes	349
12.	Buses.....	352
12.1	Overview.....	352
12.2	Description of Buses.....	354
12.2.1	CPU Buses.....	354

12.2.2	Memory Buses.....	354
12.2.3	Internal Main Buses.....	354
12.2.4	Internal Peripheral Buses.....	355
12.2.5	External Bus	356
12.2.6	Parallel Operation.....	358
12.2.7	Limitations.....	358
12.3	Register Descriptions.....	359
12.3.1	CSn Control Register (CSnCR) (n = 0 to 7).....	361
12.3.2	CSn Recovery Cycle Register (CSnREC) (n = 0 to 7).....	362
12.3.3	CSn Mode Register (CSnMOD) (n = 0 to 7).....	364
12.3.4	CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 7)	366
12.3.5	CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 7)	369
12.3.6	SDC Control Register (SDCCR).....	372
12.3.7	SDC Mode Register (SDCMOD).....	373
12.3.8	SDRAM Access Mode Register (SDAMOD).....	373
12.3.9	SDRAM Self-Refresh Control Register (SDSELF).....	374
12.3.10	SDRAM Refresh Control Register (SDRFCR).....	375
12.3.11	SDRAM Auto-Refresh Control Register (SDRFEN).....	376
12.3.12	SDRAM Initialization Sequence Control Register (SDICR)	377
12.3.13	SDRAM Initialization Register (SDIR)	378
12.3.14	SDRAM Address Register (SDADR)	380
12.3.15	SDRAM Timing Register (SDTR).....	381
12.3.16	SDRAM Mode Register (SDMOD)	383
12.3.17	SDRAM Status Register (SDSR)	384
12.3.18	Bus Error Status Clear Register (BERCLR).....	385
12.3.19	Bus Error Monitoring Enable Register (BEREN)	385
12.3.20	Bus Error Status Register 1 (BERSR1).....	386
12.3.21	Bus Error Status Register 2 (BERSR2).....	387
12.4	Endian and Data Alignment.....	388
12.4.1	Data Alignment Control for CS Area.....	388
12.4.2	Data Alignment Control for SDRAM Area.....	394
12.5	Operation of CS Area Controller.....	402
12.5.1	Timing of CS Area Access	402
12.5.2	External Wait Function.....	418
12.5.3	No Access State	420
12.5.4	Insertion of Recovery Cycles	420
12.5.5	Write Buffer Function	421
12.5.6	Notes on Usage.....	422
12.6	SDRAM Area Controller Operation.....	424
12.6.1	Enabling/Disabling SDRAM Access and Setting SDRAM Bus Width.....	424

12.6.2	No Access State	424
12.6.3	Insertion of Recovery Cycles	424
12.6.4	Write Buffer Function	425
12.6.5	SDRAM Commands.....	425
12.6.6	Conditions for Setting SDRAMC Registers.....	426
12.6.7	Self-Refresh.....	427
12.6.8	Auto-Refresh	430
12.6.9	Initialization Sequencer	432
12.6.10	Read/Write Access	433
12.6.11	Setting Mode Register	436
12.6.12	SDRAMC Setting Examples	437
12.6.12.1	SDRAMC Access Procedure	437
12.6.12.2	Procedure for Transition to and Recovery from Self-Refresh Mode	438
12.6.12.3	Timing Register Settings and Access Timing	440
12.6.13	Address Multiplexing	449
12.6.14	Examples for Connecting with SDRAMs	450
12.6.14.1	32-Bit Bus Space	450
12.6.14.2	16-Bit Bus Space	453
12.6.15	Restrictions	456
12.7	Bus Error Monitoring Section.....	457
12.7.1	Types of Bus Error	457
12.7.1.1	Illegal Address Access	457
12.7.1.2	Timeout	457
12.7.2	Operations When a Bus Error Occurs	457
12.7.3	Conditions Leading to Bus Errors	458
13.	Memory-Protection Unit (MPU).....	460
13.1	Overview.....	460
13.1.1	Types of Access Control.....	462
13.1.2	Regions for Access Control.....	462
13.1.3	Background Region.....	462
13.1.4	Overlap between Regions.....	462
13.1.5	Instructions and Data that Span Regions.....	462
13.2	Register Descriptions.....	463
13.2.1	Region-n Start Page Number Register (RSPAGEn) (n = 0 to 7)	464
13.2.2	Region-n End Page Number Register (REPAGEn) (n = 0 to 7)	465
13.2.3	Memory-Protection Enable Register (MPEN)	466
13.2.4	Background Access Control Register (MPBAC).....	467
13.2.5	Memory-Protection Error Status-Clearing Register (MPECLR)	468
13.2.6	Memory-Protection Error Status Register (MPESTS)	469
13.2.7	Data Memory-Protection Error Address Register (MPDEA).....	470

13.2.8	Region Search Address Register (MPSA).....	471
13.2.9	Region Search Operation Register (MPOPS).....	471
13.2.10	Region Invalidation Operation Register (MPOPI).....	472
13.2.11	Instruction-Hit Region Register (MHITI).....	473
13.2.12	Data-Hit Region Register(MHITD).....	475
13.3	Functions.....	477
13.3.1	Memory Protection.....	477
13.3.2	Region Search.....	477
13.3.3	Protection of Registers Related to the Memory-Protection Unit.....	477
13.3.4	Flow for Determination of Access by the Memory-Protection Function.....	478
13.4	Procedures for Using Memory Protection.....	480
13.4.1	Setting Access-Control Information.....	480
13.4.2	Enabling Memory Protection.....	480
13.4.3	Transition to User Mode.....	480
13.4.4	Processing in Response to Memory-Protection Errors.....	480
14.	DMA Controller (DMACA).....	482
14.1	Overview.....	482
14.2	Register Descriptions.....	484
14.2.1	DMA Source Address Register (DMSAR).....	486
14.2.2	DMA Destination Address Register (DMDAR).....	486
14.2.3	DMA Transfer Count Register (DMCRA).....	487
14.2.4	DMA Block Transfer Count Register (DMCRB).....	488
14.2.5	DMA Transfer Mode Register (DMTMD).....	489
14.2.6	DMA Interrupt Setting Register (DMINT).....	490
14.2.7	DMA Address Mode Register (DMAMD).....	492
14.2.8	DMA Offset Register (DMOFR).....	495
14.2.9	DMA Transfer Enable Register (DMCNT).....	496
14.2.10	DMA Software Start Register (DMREQ).....	497
14.2.11	DMA Status Register (DMSTS).....	498
14.2.12	DMA Activation Source Flag Control Register (DMCSL).....	500
14.2.13	DMA Module Start Register (DMAST).....	501
14.3	Operation.....	502
14.3.1	Transfer Mode.....	502
14.3.2	Extended Repeat Area Function.....	506
14.3.3	Address Update Function using Offset.....	508
14.3.4	Activation Sources.....	512
14.3.5	Activating the DMACA.....	514
14.3.6	Starting DMA Transfer.....	515
14.3.7	Registers during DMA Transfer.....	515
14.3.8	Channel Priority.....	517

14.3.9	Operation Timing	517
14.3.10	DMACA Execution Cycles	518
14.4	Ending DMA Transfer	519
14.4.1	Transfer End by Completion of Specified Total Number of Transfer Operations	519
14.4.2	Transfer End by Repeat Size End Interrupt.....	519
14.4.3	Transfer End by Interrupt on Extended Repeat Area Overflow	519
14.5	Interrupts.....	520
14.6	Low-Power Consumption Function.....	522
14.7	Usage Notes	523
14.7.1	DMA Transfer to External Devices.....	523
14.7.2	DMA Transfer to Peripheral Modules.....	523
14.7.3	Access to the Registers during DMACA Transfer	523
14.7.4	DMA Transfer to Reserved Areas	523
14.7.5	Interrupt Request by the DMACA Activation Source Flag Control Register (DMCSL) at the End of each Transfer.....	523
14.7.6	Setting of DMACA Activation Source Select Register of the Interrupt Control Unit (ICU.DMRSRn)	523
14.7.7	Suspending or Restarting DMACA Activation.....	523
14.7.8	DMA Controller (DMACA).....	523
15.	EXDMA Controller (EXDMAC).....	524
15.1	Overview.....	524
15.2	Register Descriptions.....	526
15.2.1	EXDMA Source Address Register (EDMSAR).....	527
15.2.2	EXDMA Destination Address Register (EDMDAR).....	527
15.2.3	EXDMA Transfer Count Register (EDMCRA)	528
15.2.4	EXDMA Block Transfer Count Register (EDMCRB).....	530
15.2.5	EXDMA Transfer Mode Register (EDMTMD).....	531
15.2.6	EXDMA Output Setting Register (EDMOMD).....	532
15.2.7	EXDMA Interrupt Setting Register (EDMINT).....	533
15.2.8	EXDMA Address Mode Register (EDMAMD).....	535
15.2.9	EXDMA Offset Register (EDMOFR).....	539
15.2.10	EXDMA Transfer Enable Register (EDMCNT)	540
15.2.11	EXDMA Software Start Register (EDMREQ).....	541
15.2.12	EXDMA Status Register (EDMSTS).....	542
15.2.13	EXDMA External Request Sense Mode Register (EDMRMD)	544
15.2.14	EXDMA External Request Flag Register (EDMERF).....	545
15.2.15	EXDMA Peripheral Request Flag Register (EDMPRF)	546
15.2.16	EXDMA Module Start Register (EDMAST)	547
15.2.17	Cluster Buffer Register i (CLSBRI) (i = 0 to 6).....	548
15.3	Operation	549
15.3.1	Transfer Mode	549

15.3.2	Extended Repeat Area Function.....	555
15.3.3	Address Update Function using Offset.....	557
15.3.4	Address Modes	561
15.4	Transfer Operation.....	562
15.4.1	Normal/Repeat Transfer Operation	562
15.4.2	Block Transfer Operation	564
15.4.3	Cluster Transfer Operation	566
15.5	Activation Sources and Procedures for Activation.....	569
15.5.1	Activation Sources.....	569
15.5.2	Activating the EXDMAC	571
15.5.3	Starting DMA Transfer.....	572
15.5.4	Registers during DMA Transfer	572
15.5.5	Channel Priority.....	573
15.6	Ending DMA Transfer	574
15.6.1	Transfer End by Completion of Specified Total Number of Transfer Operations	574
15.6.2	Transfer End by Repeat Size End Interrupt.....	575
15.6.3	Transfer End by Interrupt on Extended Repeat Area Overflow	575
15.7	Interrupts.....	576
15.8	Low-Power Consumption Function.....	578
15.9	EDACK Operation in Single Address Mode.....	579
15.9.1	EDACK Operation Example in Normal-Transfer (CS Area) Single Address Mode	579
15.9.2	EDACK Operation Example in Normal-Transfer (SDRAM Area) Single Address Mode.....	581
15.9.3	EDACK Operation Example in Block-Transfer (CS Area) Single Address Mode.....	583
15.9.4	EDACK Operation Example in Block-Transfer (SDRAM Area) Single Address Mode	585
15.10	Usage Notes	588
15.10.1	Cluster Buffers.....	588
15.10.2	Access to the Registers during DMA Transfer.....	588
15.10.3	DMA Transfer to Reserved Areas	588
16.	Data Transfer Controller (DTCa).....	589
16.1	Overview.....	589
16.2	Register Descriptions.....	591
16.2.1	DTC Mode Register A (MRA).....	592
16.2.2	DTC Mode Register B (MRB)	593
16.2.3	DTC Transfer Source Address register (SAR)	594
16.2.4	DTC Transfer Destination Address Register (DAR).....	594
16.2.5	DTC Transfer Count Register A (CRA).....	595
16.2.6	DTC Transfer Count Register B (CRB)	596
16.2.7	DTC Control Register (DTCCR).....	596
16.2.8	DTC Vector Base Register (DTCVBR)	597
16.2.9	DTC Address Mode Register (DTCADM0D).....	597

16.2.10	DTC Module Start Register (DTCST).....	598
16.2.11	DTC Status Register (DTCSTS).....	599
16.3	Sources of Activation	600
16.3.1	Allocating Transfer Data and DTC Vector Table	600
16.3.2	Startup source and Vector Address	602
16.4	Operation.....	605
16.4.1	Transfer Data Read Skip Function	608
16.4.2	Transfer Data Write-Back Skip Function.....	608
16.4.3	Normal Transfer Mode	609
16.4.4	Repeat Transfer Mode	610
16.4.5	Block Transfer Mode.....	611
16.4.6	Chain Transfer	612
16.4.7	Operation Timing	613
16.4.8	Execution Cycle of the DTC	616
16.4.9	DTC Bus Mastership Release Timing.....	616
16.5	DTC Setting Procedure.....	617
16.6	Examples of DTC Usage	618
16.6.1	Normal Transfer	618
16.6.2	Chain Transfer	619
16.6.3	Chain Transfer when Counter = 0	621
16.7	Interrupt Source	622
16.8	Low-Power Consumption Function.....	622
16.9	Usage Notes	623
16.9.1	Transfer Information Data Start Address	623
16.9.2	Allocating Transfer Data	623
16.9.3	Setting the DTC Activation Enable Register (ICU.DTCERi) of the Interrupt Control Unit	624
16.9.4	Selecting Communication Function Interrupt as DTC Activation Source	624
17.	I/O Ports.....	625
17.1	I/O Ports [for 176-Pin LFBGA].....	625
17.1.1	Overview	625
17.1.2	Register Descriptions.....	632
17.1.2.1	Data Direction Register (DDR)	638
17.1.2.2	Data Register (DR)	639
17.1.2.3	Port Register (PORT)	640
17.1.2.4	Input Buffer Control Register (ICR)	641
17.1.2.5	Open Drain Control Register (ODR)	642
17.1.2.6	Pull-Up Resistor Control Register (PCR)	643
17.1.2.7	Port Function Control Register 0 (PF0CSE)	645
17.1.2.8	Port Function Control Register 1 (PF1CSS)	646
17.1.2.9	Port Function Control Register 2 (PF2CSS)	647
17.1.2.10	Port Function Control Register 3 (PF3BUS)	648

17.1.2.11	Port Function Control Register 4 (PF4BUS)	649
17.1.2.12	Port Function Control Register 5 (PF5BUS)	650
17.1.2.13	Port Function Control Register 6 (PF6BUS)	651
17.1.2.14	Port Function Control Register 7 (PF7DMA)	652
17.1.2.15	Port Function Control Register 8 (PF8IRQ)	653
17.1.2.16	Port Function Control Register 9 (PF9IRQ)	654
17.1.2.17	Port Function Control Register A (PFAADC)	655
17.1.2.18	Port Function Control Register B (PFBTMR)	655
17.1.2.19	Port Function Control Register C (PFCMTU)	656
17.1.2.20	Port Function Control Register D (PFDMTU)	657
17.1.2.21	Port Function Control Register E (PFENET)	658
17.1.2.22	Port Function Control Register F (PFFSCI)	660
17.1.2.23	Port Function Control Register G (PFGSPI)	661
17.1.2.24	Port Function Control Register H (PFHSPI)	663
17.1.2.25	Port Function Control Register J (PFJCAN)	665
17.1.2.26	Port Function Control Register K (PFKUSB)	666
17.1.2.27	Port Function Control Register L (PFLUSB)	668
17.1.2.28	Port Function Control Register M (PFMPOE)	670
17.1.2.29	Port Function Control Register N (PFNPOE)	671
17.1.3	Settings of Ports.....	672
17.1.4	List of Output Enable Settings.....	673
17.1.5	Treatment of Unused Pins	693
17.2	I/O Ports [for 145-Pin TFLGA/144-Pin LQFP].....	694
17.2.1	Overview	694
17.2.2	Register Descriptions.....	700
17.2.2.1	Data Direction Register (DDR)	706
17.2.2.2	Data Register (DR)	707
17.2.2.3	Port Register (PORT)	708
17.2.2.4	Input Buffer Control Register (ICR)	709
17.2.2.5	Open Drain Control Register (ODR)	710
17.2.2.6	Pull-Up Resistor Control Register (PCR)	711
17.2.2.7	Port Function Control Register 0 (PF0CSE)	713
17.2.2.8	Port Function Control Register 1 (PF1CSS)	714
17.2.2.9	Port Function Control Register 2 (PF2CSS)	715
17.2.2.10	Port Function Control Register 3 (PF3BUS)	716
17.2.2.11	Port Function Control Register 4 (PF4BUS)	717
17.2.2.12	Port Function Control Register 5 (PF5BUS)	718
17.2.2.13	Port Function Control Register 6 (PF6BUS)	719
17.2.2.14	Port Function Control Register 7 (PF7DMA)	720
17.2.2.15	Port Function Control Register 8 (PF8IRQ)	721
17.2.2.16	Port Function Control Register 9 (PF9IRQ)	722

17.2.2.17	Port Function Control Register A (PFAADC)	723
17.2.2.18	Port Function Control Register B (PFBTMR)	724
17.2.2.19	Port Function Control Register C (PFCMTU)	725
17.2.2.20	Port Function Control Register D (PFDMTU)	726
17.2.2.21	Port Function Control Register E (PFENET)	727
17.2.2.22	Port Function Control Register F (PFFSCI)	729
17.2.2.23	Port Function Control Register G (PFGSPI)	730
17.2.2.24	Port Function Control Register H (PFHSPI)	732
17.2.2.25	Port Function Control Register J (PFJCAN)	734
17.2.2.26	Port Function Control Register K (PFKUSB)	735
17.2.2.27	Port Function Control Register M (PFMPOE)	737
17.2.2.28	Port Function Control Register N (PFNPOE)	738
17.2.3	Settings of Ports.....	739
17.2.4	List of Output Enable Settings.....	740
17.2.5	Treatment of Unused Pins	758
17.3	I/O Ports [for 100-Pin LQFP].....	759
17.3.1	Overview	759
17.3.2	Register Descriptions.....	764
17.3.2.1	Data Direction Register (DDR)	768
17.3.2.2	Data Register (DR)	769
17.3.2.3	Port Register (PORT)	770
17.3.2.4	Input Buffer Control Register (ICR)	771
17.3.2.5	Open Drain Control Register (ODR)	772
17.3.2.6	Pull-Up Resistor Control Register (PCR)	773
17.3.2.7	Port Function Control Register 0 (PF0CSE)	775
17.3.2.8	Port Function Control Register 3 (PF3BUS)	776
17.3.2.9	Port Function Control Register 4 (PF4BUS)	777
17.3.2.10	Port Function Control Register 5 (PF5BUS)	778
17.3.2.11	Port Function Control Register 6 (PF6BUS)	778
17.3.2.12	Port Function Control Register 8 (PF8IRQ)	779
17.3.2.13	Port Function Control Register 9 (PF9IRQ)	780
17.3.2.14	Port Function Control Register A (PFAADC)	781
17.3.2.15	Port Function Control Register C (PFCMTU)	782
17.3.2.16	Port Function Control Register D (PFDMTU)	783
17.3.2.17	Port Function Control Register E (PFENET)	784
17.3.2.18	Port Function Control Register F (PFFSCI)	786
17.3.2.19	Port Function Control Register G (PFGSPI)	787
17.3.2.20	Port Function Control Register H (PFHSPI)	789
17.3.2.21	Port Function Control Register J (PFJCAN)	791
17.3.2.22	Port Function Control Register K (PFKUSB)	792
17.3.2.23	Port Function Control Register M (PFMPOE)	794

17.3.2.24	Port Function Control Register N (PFNPOE)	795
17.3.3	Settings of Ports.....	796
17.3.4	List of Output Enable Settings.....	797
17.3.5	Treatment of Unused Pins	813
17.4	I/O Ports [for 85-Pin TFLGA]	814
17.4.1	Overview	814
17.4.2	Register Descriptions.....	817
17.4.2.1	Data Direction Register (DDR)	821
17.4.2.2	Data Register (DR)	822
17.4.2.3	Port Register (PORT)	823
17.4.2.4	Input Buffer Control Register (ICR)	824
17.4.2.5	Open Drain Control Register (ODR)	825
17.4.2.6	Pull-Up Resistor Control Register (PCR)	826
17.4.2.7	Port Function Control Register 0 (PF0CSE)	827
17.4.2.8	Port Function Control Register 3 (PF3BUS)	828
17.4.2.9	Port Function Control Register 4 (PF4BUS)	829
17.4.2.10	Port Function Control Register 8 (PF8IRQ)	830
17.4.2.11	Port Function Control Register 9 (PF9IRQ)	831
17.4.2.12	Port Function Control Register D (PFDMTU)	832
17.4.2.13	Port Function Control Register F (PFFSCI)	833
17.4.2.14	Port Function Control Register G (PFGSPI)	834
17.4.2.15	Port Function Control Register H (PFHSPI)	835
17.4.2.16	Port Function Control Register J (PFJCAN)	836
17.4.2.17	Port Function Control Register K (PFKUSB)	837
17.4.3	Settings of Ports.....	839
17.4.4	List of Output Enable Settings.....	840
17.4.5	Treatment of Unused Pins	853
17.5	Configuration of I/O Port.....	854
17.6	Usage Notes.....	858
17.6.1	Setting the Input Buffer Control Register (PORTn.ICR).....	858
17.6.2	Setting the Port Function Control Register (PFCRm).....	858
17.6.3	Changing the Output Enable Settings	859
17.6.4	Setting Open-Drain Pins with TDO Function	859
17.6.5	Reading Port Registers (PORT)	859
18.	Multi-Function Timer Pulse Unit 2 (MTU2).....	860
18.1	Overview.....	860
18.2	Register Descriptions.....	871
18.2.1	Timer Control Register (TCR)	876
18.2.2	Timer Mode Register (TMDR).....	880
18.2.3	Timer I/O Control Register (TIOR).....	883
18.2.4	Timer Compare Match Clear Register (TCNTCMPCLR)	902

18.2.5	Timer Interrupt Enable Register (TIER)	903
18.2.6	Timer Status Register (TSR)	906
18.2.7	Timer Buffer Operation Transfer Mode Register (TBTM)	907
18.2.8	Timer Input Capture Control Register (TICCR)	908
18.2.9	Timer A/D Converter Start Request Control Register (TADCR)	909
18.2.10	Timer A/D Converter Start Request Cycle Set Register (TADCORA, TADCORB)	910
18.2.11	Timer A/D Converter Start Request Cycle Set Buffer Register (TADCORA, TADCORB)	910
18.2.12	Timer Counter (TCNT)	911
18.2.13	Timer General Register (TGR)	911
18.2.14	Timer Start Register (TSTR)	912
18.2.15	Timer Synchronous Register (TSYR)	914
18.2.16	Timer Read/Write Enable Register (TRWER)	915
18.2.17	Timer Output Master Enable Register (TOER)	916
18.2.18	Timer Output Control Register 1 (TOCR1)	917
18.2.19	Timer Output Control Register 2 (TOCR2)	919
18.2.20	Timer Output Level Buffer Register (TOLBR)	922
18.2.21	Timer Gate Control Register (TGCR)	923
18.2.22	Timer Subcounter (TCNTS)	925
18.2.23	Timer Dead Time Data Register (TDDR)	925
18.2.24	Timer Cycle Data Register (TCDR)	925
18.2.25	Timer Cycle Buffer Register (TCBR)	926
18.2.26	Timer Interrupt Skipping Set Register (TITCR)	926
18.2.27	Timer Interrupt Skipping Counter (TITCNT)	928
18.2.28	Timer Buffer Transfer Set Register (TBTER)	929
18.2.29	Timer Dead Time Enable Register (TDER)	930
18.2.30	Timer Waveform Control Register (TWCR)	931
18.2.31	Bus Master Interface	932
18.3	Operation	933
18.3.1	Basic Functions	933
18.3.2	Synchronous Operation	939
18.3.3	Buffer Operation	941
18.3.4	Cascaded Operation	946
18.3.5	PWM Modes	951
18.3.6	Phase Counting Mode	956
18.3.7	Reset-Synchronized PWM Mode	962
18.3.8	Complementary PWM Mode	965
18.3.9	A/D Converter Start Request Delaying Function	1001
18.3.10	External Pulse Width Measurement	1004
18.3.11	Dead Time Compensation	1005

18.3.12	TCNT Capture at Crest and/or Trough in Complementary PWM Operation	1007
18.4	Interrupt Sources	1008
18.4.1	Interrupt Sources and Priorities	1008
18.4.2	DTC and DMACA Activation.....	1010
18.4.3	A/D Converter Activation	1011
18.5	Operation Timing	1013
18.5.1	Input/Output Timing.....	1013
18.5.2	Interrupt Signal Timing	1020
18.6	Usage Notes	1023
18.6.1	Module Clock Stop Mode Setting	1023
18.6.2	Input Clock Restrictions	1023
18.6.3	Note on Cycle Setting.....	1023
18.6.4	Contention between TCNT Write and Clear Operations.....	1024
18.6.5	Contention between TCNT Write and Increment Operations	1024
18.6.6	Contention between TGR Write Operation and Compare Match	1025
18.6.7	Contention between Buffer Register Write Operation and Compare Match.....	1025
18.6.8	Contention between Buffer Register Write and TCNT Clear Operations.....	1026
18.6.9	Contention between TGR Read Operation and Input Capture	1027
18.6.10	Contention between TGR Write Operation and Input Capture	1028
18.6.11	Contention between Buffer Register Write Operation and Input Capture	1029
18.6.12	Contention between MTUn.TCNT Write Operation and Overflow/ Underflow in Cascaded Operation (n = 2 or 8)	1030
18.6.13	Counter Value when Stopped in Complementary PWM Mode	1031
18.6.14	Buffer Operation Setting in Complementary PWM Mode.....	1031
18.6.15	Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode	1032
18.6.16	Overflow Flags in Reset-Synchronized PWM Mode	1033
18.6.17	Contention between Overflow/Underflow and Counter Clearing	1034
18.6.18	Contention between TCNT Write Operation and Overflow/Underflow	1034
18.6.19	Note on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode	1035
18.6.20	Output Levels in Complementary PWM Mode and Reset-Synchronous PWM Mode	1035
18.6.21	Interrupts in the Module-Stop State.....	1035
18.6.22	Simultaneous Input Capture in MTUn.TCNT and MTUm.TCNT in Cascade Connection....	1035
18.6.23	Notes when Complementary PWM Mode Output Protection Functions are not Used	1035
18.6.24	Preventing Abnormal Operation of Synchronous Clearing in Complementary PWM Mode.	1036
18.7	MTU Output Pin Initialization	1038
18.7.1	Operating Modes	1038
18.7.2	Operation in Case of Re-Setting Due to Error during Operation	1038
18.7.3	Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation	1039

19.	Port Output Enable 2 (POE2)	1064
19.1	Overview	1064
19.2	Register Descriptions	1067
19.2.1	Input Level Control/Status Register 1 (ICSR1)	1068
19.2.2	Output Level Control/Status Register 1 (OCSR1)	1070
19.2.3	Input Level Control/Status Register 2 (ICSR2)	1071
19.2.4	Output Level Control/Status Register 2 (OCSR2)	1073
19.2.5	Input Level Control/Status Register 3 (ICSR3)	1074
19.2.6	Input Level Control/Status Register 4 (ICSR4)	1075
19.2.7	Software Port Output Enable Register (SPOER)	1076
19.2.8	Port Output Enable Control Register 1 (POECR1)	1078
19.2.9	Port Output Enable Control Register 2 (POECR2)	1080
19.3	Operation	1082
19.3.1	Input Level Detection Operation	1083
19.3.2	Output-Level Compare Operation	1084
19.3.3	High-Impedance Control Using Registers	1084
19.3.4	High-Impedance Control through Detection of Stopped Oscillation	1084
19.3.5	Release from High-Impedance State	1084
19.4	Interrupts	1085
19.5	Usage Notes	1085
20.	Programmable Pulse Generator (PPG)	1086
20.1	Overview	1086
20.2	Register Descriptions	1090
20.2.1	PPG Trigger Select Register (PTRSLR)	1091
20.2.2	Next Data Transfer Enable Registers H and L (NDERH, NDERL)	1092
20.2.3	Output Data Registers H and L (PODRH, PODRL)	1095
20.2.4	Next Data Registers H and L (NDRH, NDRL)	1097
20.2.5	PPG Output Control Register (PCR)	1102
20.2.6	PPG Output Mode Register (PMR)	1104
20.3	Operation	1106
20.3.1	Output Timing	1107
20.3.2	Sample Setup Procedure for Normal Pulse Output	1108
20.3.3	Example of Normal Pulse Output (Example of Five-Phase Pulse Output)	1110
20.3.4	Non-Overlapping Pulse Output	1111
20.3.5	Sample Setup Procedure for Non-Overlapping Pulse Output	1112
20.3.6	Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)	1114
20.3.7	Inverted Pulse Output	1116
20.3.8	Pulse Output Triggered by Input Capture	1117
20.4	Usage Note	1118
20.4.1	Module Stop Function Setting	1118

21.	8-Bit Timer (TMR)	1119
21.1	Overview	1119
21.2	Register Descriptions	1123
21.2.1	Timer Counter (TCNT)	1124
21.2.2	Time Constant Register A (TCORA)	1124
21.2.3	Time Constant Register B (TCORB)	1125
21.2.4	Timer Control Register (TCR)	1126
21.2.5	Timer Counter Control Register (TCCR)	1127
21.2.6	Timer Control/Status Register (TCSR)	1129
21.3	Operation	1131
21.3.1	Pulse Output	1131
21.3.2	Reset Input	1132
21.4	Operation Timing	1133
21.4.1	TCNT Count Timing	1133
21.4.2	Timing of the Interrupt for a Compare-Match	1134
21.4.3	Timing of Timer Output at Compare Match	1135
21.4.4	Timing of Counter Clear by Compare Match	1135
21.4.5	Timing of the External Reset for TCNT	1136
21.4.6	Timing of the Interrupt for an Overflow	1137
21.5	Operation with Cascaded Connection	1138
21.5.1	16-Bit Count Mode	1138
21.5.2	Compare Match Count Mode	1138
21.6	Interrupt Sources	1139
21.6.1	Interrupt Sources and DTC Activation	1139
21.6.2	A/D Converter Activation	1139
21.7	Usage Notes	1140
21.7.1	Module Stop State Setting	1140
21.7.2	Notes on Setting Cycle	1140
21.7.3	Conflict between TCNT Write and Counter Clear	1140
21.7.4	Conflict between TCNT Write and Increment	1141
21.7.5	Conflict between TCORA or TCORB Write and Compare Match	1141
21.7.6	Conflict between Compare Matches A and B	1142
21.7.7	Switching of Internal Clocks and TCNT Operation	1143
21.7.8	Clock Source Setting with Cascaded Connection	1144
22.	Compare Match Timer (CMT)	1145
22.1	Overview	1145
22.2	Register Descriptions	1146
22.2.1	Compare Match Timer Start Register 0 (CMSTR0)	1147
22.2.2	Compare Match Timer Start Register 1 (CMSTR1)	1148
22.2.3	Compare Match Timer Control Register (CMCR)	1149

22.2.4	Compare Match Timer Counter (CMCNT).....	1150
22.2.5	Compare Match Timer Constant Register (CMCOR).....	1150
22.3	Operation.....	1151
22.3.1	Periodic Count Operation.....	1151
22.3.2	CMCNT Count Timing	1151
22.4	Interrupts.....	1152
22.4.1	Interrupt Sources.....	1152
22.4.2	Timing of Compare Match Interrupt Generation	1152
22.5	Usage Notes.....	1153
22.5.1	Setting the Module Stop Function	1153
22.5.2	Conflict between Write and Compare-Match Processes of Compare Match Timer Counter (CMCNT).....	1153
22.5.3	Conflict between Write and Count-Up Processes of Compare Match Timer Counter (CMCNT).....	1153
22.5.4	Notes on Data Write to the Compare-Match Timer Control Register (CMCR)	1154
22.5.5	Notes on the Compare-Match Timer Counter (CMCNT) and the Compare-Match Constant Register (CMCOR)	1154
23.	Realtime Clock (RTC).....	1155
23.1	Overview.....	1155
23.2	Register Descriptions.....	1157
23.2.1	64-Hz Counter (R64CNT).....	1158
23.2.2	Second Counter (RSECCNT).....	1159
23.2.3	Minute Counter (RMINCNT).....	1159
23.2.4	Hour Counter (RHRCNT)	1160
23.2.5	Day-of-Week Counter (RWKCNT)	1161
23.2.6	Date Counter (RDAYCNT).....	1162
23.2.7	Month Counter (RMONCNT).....	1162
23.2.8	Year Counter (RYRCNT).....	1163
23.2.9	Second Alarm Register (RSECAR).....	1164
23.2.10	Minute Alarm Register (RMINAR)	1165
23.2.11	Hour Alarm Register (RHRAR).....	1166
23.2.12	Day-of-Week Alarm Register (RWKAR)	1167
23.2.13	Date Alarm Register (RDAYAR).....	1168
23.2.14	Month Alarm Register (RMONAR).....	1169
23.2.15	Year Alarm Register (RYRAR)	1170
23.2.16	Year Alarm Enable Register (RYRAREN)	1170
23.2.17	RTC Control Register 1 (RCR1)	1171
23.2.18	RTC Control Register 2 (RCR2)	1172
23.3	Operation.....	1173
23.3.1	Initial Settings of Registers after Power-On.....	1173
23.3.2	Setting Time	1173

23.3.3	30-Second Adjustment	1174
23.3.4	Reading 64-Hz Counter and Time.....	1175
23.3.5	Alarm Function.....	1176
23.3.6	Procedure for Disabling Alarm Interrupt.....	1177
23.4	Interrupt Sources	1178
23.5	Usage Notes	1180
23.5.1	Register Writing during Counting	1180
23.5.2	Use of Periodic Interrupts.....	1180
23.5.3	Transition to Low Power Consumption Modes after Setting Registers	1181
23.5.4	Points for Caution When Writing to and Reading from Registers	1181
24.	Watchdog Timer (WDT)	1182
24.1	Overview.....	1182
24.2	Register Descriptions.....	1184
24.2.1	Timer Counter (TCNT)	1184
24.2.2	Timer Control/Status Register (TCSR)	1185
24.2.3	Reset Control/Status Register (RSTCSR)	1186
24.2.4	Write Window A Register (WINA).....	1187
24.2.5	Write Window B Register (WINB).....	1187
24.3	Operation.....	1188
24.3.1	Watchdog Timer Mode.....	1188
24.3.2	Interval Timer Mode.....	1189
24.4	Interrupt Source	1189
24.5	Usage Notes	1190
24.5.1	Notes on Register Access	1190
24.5.2	Conflict between Timer Counter (TCNT) Write and Increment.....	1191
24.5.3	Changing Values of Bits CKS[2:0]	1191
24.5.4	Switching between Watchdog Timer Mode and Interval Timer Mode.....	1192
24.5.5	Internal Reset in Watchdog Timer Mode	1192
24.5.6	System Reset by WDTOVF# Signal	1192
24.5.7	Transition to Watchdog Timer Mode or Software Standby Mode.....	1192
25.	Independent Watchdog Timer (IWDT)	1193
25.1	Overview.....	1193
25.2	Register Descriptions.....	1194
25.2.1	IWDT Refresh Register (IWDTRR).....	1194
25.2.2	IWDT Control Register (IWDTCR).....	1195
25.2.3	IWDT Status Register (IWDTSR).....	1197
25.3	Description of Operation	1198
25.3.1	Count Operation of the Down-Counter	1198
25.3.2	Control over Writing to the IWDT Control Register (IWDTCR).....	1199
25.3.3	Control of Refreshing	1200

25.3.4	Status Flags.....	1201
25.4	Usage Notes.....	1201
25.4.1	Limitation on Transitions to Low-Power-Consumption Modes.....	1201
26.	Ethernet Controller (ETHERC).....	1202
26.1	Overview.....	1202
26.2	Register Descriptions.....	1204
26.2.1	ETHERC Mode Register (ECMR).....	1205
26.2.2	ETHERC Status Register (ECSR).....	1207
26.2.3	ETHERC Interrupt Permission Register (ECSIPR).....	1208
26.2.4	Receive Frame Length Register (RFLR).....	1209
26.2.5	PHY Interface Register (PIR).....	1210
26.2.6	MAC Address High Register (MAHR).....	1211
26.2.7	MAC Address Low Register (MALR).....	1211
26.2.8	PHY Status Register (PSR).....	1212
26.2.9	Transmit Retry Over Counter Register (TROCR).....	1212
26.2.10	Delayed Collision Detect Counter Register (CDCR).....	1213
26.2.11	Lost Carrier Counter Register (LCCR).....	1213
26.2.12	Carrier Not Detect Counter Register (CNDCCR).....	1214
26.2.13	CRC Error Frame Receive Counter Register (CEFCR).....	1214
26.2.14	Frame Receive Error Counter Register (FRECR).....	1215
26.2.15	Too-Short Frame Receive Counter Register (TSFRCCR).....	1215
26.2.16	Too-Long Frame Receive Counter Register (TLFRCCR).....	1216
26.2.17	Residual-Bit Frame Receive Counter Register (RFCR).....	1216
26.2.18	Multicast Address Frame Receive Counter Register (MAFCR).....	1217
26.2.19	IPG Register (IPGR).....	1217
26.2.20	Automatic PAUSE Frame Register (APR).....	1218
26.2.21	Manual PAUSE Frame Register (MPR).....	1218
26.2.22	Automatic PAUSE Frame Retransmit Count Register (TPAUSER).....	1219
26.2.23	Random Number Generation Counter Upper Limit Setting Register (RDMLR).....	1219
26.2.24	PAUSE Frame Receive Counter Register (RFCF).....	1220
26.2.25	PAUSE Frame Retransmit Counter Register (TPAUSECR).....	1221
26.2.26	Broadcast Frame Receive Count Setting Register (BCFRR).....	1221
26.3	Operation.....	1222
26.3.1	Transmission.....	1222
26.3.2	Reception.....	1224
26.3.3	Frame Timing.....	1225
26.3.3.1	MII Frame Timing.....	1225
26.3.3.2	RMII Frame Timing.....	1227
26.3.4	Accessing MII/RMII Registers.....	1228
26.3.4.1	MII/RMII Management Frame Format.....	1228

26.3.4.2	MII/RMII Register Access Procedure	1229
26.3.5	Magic Packet™ Detection	1231
26.3.6	Operation by IPG Setting	1231
26.3.7	Flow Control.....	1232
26.3.7.1	Automatic PAUSE Frame Transmission	1232
26.3.7.2	Manual PAUSE Frame Transmission	1232
26.3.7.3	PAUSE Frame Reception	1232
26.4	Connection to PHY-LSI.....	1233
26.5	Usage Notes.....	1234
26.5.1	Conditions for Setting LCHNG Bit	1234
26.5.2	RMII_RX_ER Pin Input when RMII is Selected	1234
27.	Ethernet Controller Direct Memory Access Controller (EDMAC).....	1235
27.1	Overview.....	1235
27.2	Register Descriptions.....	1236
27.2.1	EDMAC Mode Register (EDMR).....	1237
27.2.2	EDMAC Transmit Request Register (EDTRR)	1238
27.2.3	EDMAC Receive Request Register (EDRRR).....	1239
27.2.4	Transmit Descriptor List Start Address Register (TDLAR).....	1240
27.2.5	Receive Descriptor List Start Address Register (RDLAR).....	1241
27.2.6	ETHERC/EDMAC Status Register (EESR).....	1242
27.2.7	ETHERC/EDMAC Status Interrupt Permission Register (EESIPR)	1245
27.2.8	Transmit/Receive Status Copy Enable Register (TRSCER)	1247
27.2.9	Receive Missed-Frame Counter Register (RMFCR).....	1248
27.2.10	Transmit FIFO Threshold Register (TFTR)	1249
27.2.11	FIFO Depth Register (FDR).....	1250
27.2.12	Receiving Method Control Register (RMCR).....	1251
27.2.13	Transmit FIFO Underrun Counter (TFUCR)	1252
27.2.14	Receive FIFO Overflow Counter (RFOCR).....	1252
27.2.15	Receive Buffer Write Address Register (RBWAR)	1253
27.2.16	Receive Descriptor Fetch Address Register (RDFAR)	1253
27.2.17	Transmit Buffer Read Address Register (TBRAR).....	1254
27.2.18	Transmit Descriptor Fetch Address Register (TDFAR).....	1254
27.2.19	Flow Control Start FIFO Threshold Setting Register (FCFTR).....	1255
27.2.20	Receive Data Padding Insert Register (RPADIR).....	1256
27.2.21	Transmit Interrupt Setting Register (TRIMD).....	1257
27.2.22	Independent Output Signal Setting Register (IOSR).....	1258
27.3	Operation.....	1259
27.3.1	Descriptor Lists and Data Buffers	1259
27.3.1.1	Transmit Descriptor	1259
27.3.1.2	Receive Descriptor	1262

27.3.2	Transmission.....	1265
27.3.3	Reception.....	1266
27.3.4	Transmit/Receive Processing of Multi-Buffer Frame	1267
27.3.4.1	Multi-Buffer Frame Transmit Processing	1267
27.3.4.2	Receive Processing in Case of Multi-Buffer Frame	1268
28.	USB 2.0 Host/Function Module (USB).....	1269
28.1	Overview.....	1269
28.2	Register Descriptions.....	1272
28.2.1	System Configuration Control Register (SYSCFG).....	1276
28.2.2	System Configuration Status Register 0 (SYSSTS0).....	1278
28.2.3	Device State Control Register 0 (DVSTCTR0)	1279
28.2.4	CFIFO Port Register (CFIFO) D0FIFO Port Register (D0FIFO) D1FIFO Port Register (D1FIFO)	1282
28.2.5	CFIFO Port Select Register (CFIFOSEL) D0FIFO Port Select Register (D0FIFOSEL) D1FIFO Port Select Register (D1FIFOSEL).....	1284
28.2.6	CFIFO Port Control Register (CFIFOCTR) D0FIFO Port Control Register (D0FIFOCTR) D1FIFO Port Control Register (D1FIFOCTR)	1289
28.2.7	Interrupt Enable Register 0 (INTENB0)	1291
28.2.8	Interrupt Enable Register 1 (INTENB1)	1293
28.2.9	BRDY Interrupt Enable Register (BRDYENB).....	1295
28.2.10	NRDY Interrupt Enable Register (NRDYENB)	1296
28.2.11	BEMP Interrupt Enable Register (BEMPENB)	1297
28.2.12	SOF Output Configuration Register (SOFCFG)	1298
28.2.13	Interrupt Status Register 0 (INTSTS0).....	1299
28.2.14	Interrupt Status Register 1 (INTSTS1).....	1302
28.2.15	BRDY Interrupt Status Register (BRDYSTS)	1305
28.2.16	NRDY Interrupt Status Register (NRDYSTS).....	1306
28.2.17	BEMP Interrupt Status Register (BEMPSTS).....	1307
28.2.18	Frame Number Register (FRMNUM)	1308
28.2.19	Device State Change Register (DVCHGR).....	1309
28.2.20	USB Address Register (USBADDR)	1310
28.2.21	USB Request Type Register (USBREQ).....	1311
28.2.22	USB Request Value Register (USBVAL)	1312
28.2.23	USB Request Index Register (USBINDEX).....	1312
28.2.24	USB Request Length Register (USBLENG).....	1313
28.2.25	DCP Configuration Register (DCPCFG)	1314
28.2.26	DCP Maximum Packet Size Register (DCPMAXP).....	1315
28.2.27	DCP Control Register (DCPCTR).....	1316
28.2.28	Pipe Window Select Register (PIPESEL)	1320

28.2.29	Pipe Configuration Register (PIPECFG).....	1321
28.2.30	Pipe Maximum Packet Size Register (PIPEMAXP).....	1323
28.2.31	Pipe Cycle Control Register (PIPEPERI).....	1324
28.2.32	PIPE _n Control Registers (PIPE _n CTR) (n = 1 to 9).....	1325
28.2.33	PIPE _n Transaction Counter Enable Registers (PIPE _n TRE) (n = 1 to 5).....	1333
28.2.34	PIPE _n Transaction Counter Registers (PIPE _n TRN) (n = 1 to 5).....	1334
28.2.35	Device Address n Configuration Registers (DEVADD _n) (n = 0 to 5).....	1335
28.2.36	Deep Standby USB Transceiver Control/Pin Monitor Register (DPUSR0R).....	1336
28.2.37	Deep Standby USB Suspend/Resume Interrupt Register (DPUSR1R).....	1338
28.3	Operation.....	1342
28.3.1	System Control.....	1342
28.3.1.1	Starting Operation.....	1342
28.3.1.2	Controller Function Selection.....	1342
28.3.1.3	Example of USB External Connection Circuit.....	1342
28.3.1.4	Canceling Deep Software Standby Mode by a USB Suspend/Resume Interrupt.....	1347
28.3.2	Interrupt Sources.....	1351
28.3.3	Interrupt Descriptions.....	1355
28.3.3.1	BRDY Interrupt.....	1355
28.3.3.2	NRDY Interrupt.....	1358
28.3.3.3	BEMP Interrupt.....	1361
28.3.3.4	Device State Transition Interrupt.....	1363
28.3.3.5	Control Transfer Stage Transition Interrupt.....	1364
28.3.3.6	Frame Update Interrupt.....	1365
28.3.3.7	VBUS Interrupt.....	1365
28.3.3.8	Resume Interrupt.....	1365
28.3.3.9	OVRCR Interrupt.....	1365
28.3.3.10	BCHG Interrupt.....	1365
28.3.3.11	DTCH Interrupt.....	1366
28.3.3.12	SACK Interrupt.....	1366
28.3.3.13	SIGN Interrupt.....	1366
28.3.3.14	ATTCH Interrupt.....	1366
28.3.3.15	EOFERR Interrupt.....	1366
28.3.4	Pipe Control.....	1367
28.3.4.1	Pipe Control Register Switching Procedures.....	1368
28.3.4.2	Transfer Types.....	1369
28.3.4.3	Endpoint Number.....	1369
28.3.4.4	Maximum Packet Size Setting.....	1369
28.3.4.5	Transaction Counter (For PIPE1 to PIPE5 in Reading Direction).....	1369
28.3.4.6	Response PID.....	1370
28.3.4.7	Data PID Sequence Bit.....	1371
28.3.4.8	Response PID = NAK Function.....	1371

28.3.4.9	Auto Response Mode	1371
28.3.4.10	OUT-NAK Mode	1371
28.3.4.11	Null Auto Response Mode	1372
28.3.5	FIFO Buffer Memory	1373
28.3.5.1	FIFO Buffer Memory	1373
28.3.5.2	FIFO Buffer Clearing	1374
28.3.5.3	FIFO Port Functions	1374
28.3.5.4	DMA Transfers (D0FIFO and D1FIFO Ports)	1376
28.3.6	Control Transfers (DCP)	1377
28.3.6.1	Control Transfers when Host Controller Function is Selected	1377
28.3.6.2	Control Transfers when Function Controller Function is Selected	1378
28.3.7	Bulk Transfers (PIPE1 to PIPE5)	1379
28.3.8	Interrupt Transfers (PIPE6 to PIPE9)	1380
28.3.8.1	Interval Counter during Interrupt Transfers when Host Controller Function is Selected	1380
28.3.9	Isochronous Transfers (PIPE1 and PIPE2)	1381
28.3.9.1	Error Detection in Isochronous Transfers	1381
28.3.9.2	DATA-PID	1383
28.3.9.3	Interval Counter	1383
28.3.10	SOF Interpolation Function	1391
28.3.11	Pipe Schedule	1392
28.3.11.1	Conditions for Generating a Transaction	1392
28.3.11.2	Transfer Schedule	1393
28.3.11.3	Enabling USB Communication	1393
29.	Serial Communications Interface (SC1a)	1394
29.1	Overview	1394
29.2	Serial Communications Interface Mode	1398
29.2.1	Register Descriptions	1398
29.2.1.1	Receive Shift Register (RSR)	1400
29.2.1.2	Receive Data Register (RDR)	1400
29.2.1.3	Transmit Data Register (TDR)	1400
29.2.1.4	Transmit Shift Register (TSR)	1401
29.2.1.5	Serial Mode Register (SMR)	1401
29.2.1.6	Serial Control Register (SCR)	1403
29.2.1.7	Serial Status Register (SSR)	1406
29.2.1.8	Smart Card Mode Register (SCMR)	1409
29.2.1.9	Bit Rate Register (BRR)	1410
29.2.1.10	Serial Extended Mode Register (SEMR)	1416
29.2.2	Operation in Asynchronous Mode	1418
29.2.2.1	Serial Data Transfer Format	1419
29.2.2.2	Receive Data Sampling Timing and Reception Margin in Asynchronous Mode	1420

29.2.2.3	Clock	1421
29.2.2.4	SCI Initialization (Asynchronous Mode)	1422
29.2.2.5	Serial Data Transmission (Asynchronous Mode)	1423
29.2.2.6	Serial Data Reception (Asynchronous Mode)	1425
29.2.3	Multi-Processor Communications Function.....	1429
29.2.3.1	Multi-Processor Serial Data Transmission	1430
29.2.3.2	Multi-Processor Serial Data Reception	1431
29.2.4	Operation in Clock Synchronous Mode	1434
29.2.4.1	Clock	1434
29.2.4.2	SCI Initialization (Clock Synchronous Mode)	1435
29.2.4.3	Serial Data Transmission (Clock Synchronous Mode)	1436
29.2.4.4	Serial Data Reception (Clock Synchronous Mode)	1438
29.2.4.5	Serial Data Full-Duplex Communications (Clock Synchronous Mode)	1440
29.3	Smart-Card Interface Mode	1442
29.3.1	Register Descriptions.....	1442
29.3.1.1	Serial Mode Register (SMR)	1443
29.3.1.2	Serial Control Register (SCR)	1445
29.3.1.3	Serial Status Register (SSR)	1447
29.3.1.4	Bit Rate Register (BRR)	1450
29.3.2	Operation in Smart Card Interface Mode	1452
29.3.2.1	Sample Connection	1452
29.3.3	Data Format (Except in Block Transfer Mode).....	1453
29.3.3.1	Block Transfer Mode	1454
29.3.3.2	Receive Data Sampling Timing and Reception Margin	1455
29.3.3.3	Initialization of the Smart Card Interface	1456
29.3.3.4	Serial Data Transmission (Except in Block Transfer Mode)	1457
29.3.3.5	Serial Data Reception (Except in Block Transfer Mode)	1460
29.3.3.6	Clock Output Control	1462
29.4	Interrupt Sources	1464
29.4.1	Interrupts in Serial Communications Interface Mode	1464
29.4.2	Interrupts in Smart Card Interface Mode.....	1465
29.5	Usage Notes	1466
29.5.1	Setting the Module Stop Function	1466
29.5.2	Break Detection and Processing.....	1466
29.5.3	Mark State and Break Detection.....	1466
29.5.4	Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only).....	1466
29.5.5	Writing Data to TDR	1466
29.5.6	Restrictions on Clock Synchronous Transmission	1466
29.5.7	Restrictions on Using DTC or DMACA	1466
29.5.8	SCI Operations during Low Power Consumption State.....	1467
29.5.9	External Clock Input in Clock Synchronous Mode.....	1470

30.	CRC Calculator (CRC)	1471
30.1	Overview	1471
30.2	Register Descriptions	1472
30.2.1	CRC Control Register (CRCCR)	1472
30.2.2	CRC Data Input Register (CRCDIR)	1473
30.2.3	CRC Data Output Register (CRCDOR)	1473
30.3	Operation	1474
30.4	Usage Notes	1477
30.4.1	Module Stop Function Setting	1477
30.5	Note on Transmission	1477
31.	I2C Bus Interface (RIIC)	1478
31.1	Overview	1478
31.2	Register Descriptions	1481
31.2.1	I2C Bus Control Register 1 (ICCR1)	1483
31.2.2	I2C Bus Control Register 2 (ICCR2)	1486
31.2.3	I2C Bus Mode Register 1 (ICMR1)	1490
31.2.4	I2C Bus Mode Register 2 (ICMR2)	1491
31.2.5	I2C Bus Mode Register 3 (ICMR3)	1493
31.2.6	I2C Bus Function Enable Register (ICFER)	1496
31.2.7	I2C Bus Status Enable Register (ICSER)	1498
31.2.8	I2C Bus Interrupt Enable Register (ICIER)	1500
31.2.9	I2C Bus Status Register 1 (ICSR1)	1502
31.2.10	I2C Bus Status Register 2 (ICSR2)	1505
31.2.11	Slave Address Register Ly (SARLy) (y = 0 to 2)	1509
31.2.12	Slave Address Register Um (SARUm) (m = 0 to 2)	1510
31.2.13	I2C Bus Bit Rate Low-Level Register (ICBRL)	1511
31.2.14	I2C Bus Bit Rate High-Level Register (ICBRH)	1512
31.2.15	I2C Bus Transmit Data Register (ICDRT)	1514
31.2.16	I2C Bus Receive Data Register (ICDRR)	1514
31.2.17	I2C Bus Shift Register (ICDRS)	1514
31.3	Operation	1515
31.3.1	Communication Data Format	1515
31.3.2	Initial Settings	1516
31.3.3	Master Transmitter Operation	1517
31.3.4	Master Receiver Operation	1521
31.3.5	Slave Transmitter Operation	1526
31.3.6	Slave Receiver Operation	1529
31.4	SCL Synchronization Circuit	1532
31.5	Facility for Delaying SDA Output	1533
31.6	Digital Noise-Filter Circuits	1534
31.7	Address Match Detection	1535

31.7.1	Slave-Address Match Detection	1535
31.7.2	Detection of the General Call Address	1538
31.7.3	Device-ID Address Detection.....	1539
31.7.4	Host Address Detection	1541
31.8	Function to Automatically Hold SCLn Clock Low.....	1542
31.8.1	Function to Prevent Wrong Transmission of Transmit Data.....	1542
31.8.2	NACK Reception Transfer Suspension Function.....	1543
31.8.3	Function to Prevent Failure to Receive Data.....	1544
31.9	Arbitration-Lost Detection Functions.....	1546
31.9.1	Master Arbitration Lost Detection (MALE Bit).....	1546
31.9.2	Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)	1548
31.9.3	Slave Arbitration Lost Detection (SALE Bit)	1549
31.10	Start Condition/Restart Condition/Stop Condition Issuing Function	1550
31.10.1	Issuing a Start Condition	1550
31.10.2	Issuing a Restart Condition.....	1550
31.10.3	Issuing a Stop Condition.....	1551
31.11	Bus Hanging	1552
31.11.1	Timeout Detection Function.....	1552
31.11.2	Extra SCL Clock Cycle Output Function.....	1554
31.11.3	RIIC Reset and Internal Reset	1555
31.12	SMBus Operation	1555
31.12.1	SMBus Timeout Measurement.....	1555
31.12.2	Packet Error Code (PEC).....	1556
31.12.3	SMBus Host Notification Protocol/Notify ARP Master	1557
31.13	Interrupt Request.....	1558
31.14	Reset States	1559
31.15	Usage Notes	1560
31.15.1	Setting Module Stop Function.....	1560
31.15.2	Setting Input Buffer Control Register	1560
32.	CAN Module (CAN).....	1561
32.1	Overview.....	1561
32.2	Register Descriptions.....	1564
32.2.1	Control Register (CTRL).....	1565
32.2.2	Bit Configuration Register (BCR).....	1569
32.2.3	Mask Register i (MKRi) (i = 0 to 7).....	1571
32.2.4	FIFO Received ID Compare Registers 0 and 1 (FIDCR0, FIDCR1).....	1572
32.2.5	Mask Invalid Register (MKIVLR)	1574
32.2.6	Mailbox Register j (MBj) (j = 0 to 31)	1575
32.2.7	Mailbox Interrupt Enable Register (MIER).....	1579
32.2.8	Message Control Register j (MCTLj) (j = 0 to 31)	1581

32.2.9	Receive FIFO Control Register (RFCR)	1585
32.2.10	Receive FIFO Pointer Control Register (RFPCR)	1588
32.2.11	Transmit FIFO Control Register (TFCR).....	1589
32.2.12	Transmit FIFO Pointer Control Register (TFPCR).....	1591
32.2.13	Status Register (STR)	1592
32.2.14	Mailbox Search Mode Register (MSMR)	1595
32.2.15	Mailbox Search Status Register (MSSR)	1596
32.2.16	Channel Search Support Register (CSSR).....	1597
32.2.17	Acceptance Filter Support Register (AFSR)	1598
32.2.18	Error Interrupt Enable Register (EIER).....	1599
32.2.19	Error Interrupt Factor Judge Register (EIFR).....	1601
32.2.20	Receive Rrror Count Register (RECR)	1604
32.2.21	Transmit Error Count Register (TECR)	1604
32.2.22	Error Code Store Register (ECSR).....	1605
32.2.23	Time Stamp Register (TSR)	1607
32.2.24	Test Control Register (TCR)	1608
32.3	Operating Mode	1610
32.3.1	CAN Reset Mode.....	1611
32.3.2	CAN Halt Mode.....	1612
32.3.3	CAN Sleep Mode.....	1613
32.3.4	CAN Operation Mode (Excluding Bus-Off State)	1613
32.3.5	CAN Operation Mode (Bus-Off State).....	1614
32.4	CAN Communication Speed Setting.....	1615
32.4.1	CAN Clock Setting	1615
32.4.2	Bit Timing Setting	1615
32.4.3	Bit Rate.....	1616
32.5	Mailbox and Mask Register Structure.....	1617
32.6	Acceptance Filtering and Masking Functions	1619
32.7	Reception and Transmission	1622
32.7.1	Reception.....	1623
32.7.2	Transmission.....	1625
32.8	CAN Interrupt.....	1626
33.	Serial Peripheral Interface (RSPI).....	1627
33.1	Overview.....	1627
33.2	Register Descriptions.....	1630
33.2.1	RSPI Control Register (SPCR).....	1632
33.2.2	RSPI Slave Select Polarity Register (SSLP)	1634
33.2.3	RSPI Pin Control Register (SPPCR)	1635
33.2.4	RSPI Status Register (SPSR).....	1637
33.2.5	RSPI Data Register (SPDR)	1640

33.2.6	RSPI Sequence Control Register (SPSCR)	1641
33.2.7	RSPI Sequence Status Register (SPSSR)	1642
33.2.8	RSPI Bit Rate Register (SPBR).....	1643
33.2.9	RSPI Data Control Register (SPDCR)	1644
33.2.10	RSPI Clock Delay Register (SPCKD).....	1647
33.2.11	RSPI Slave Select Negation Delay Register (SSLND)	1648
33.2.12	RSPI Next-Access Delay Register (SPND).....	1649
33.2.13	RSPI Control Register 2 (SPCR2).....	1650
33.2.14	RSPI Command Registers 0 to 7 (SPCMD0 to SPCMD7)	1651
33.3	Operation	1654
33.3.1	Overview of RSPI Operations	1654
33.3.2	Controlling RSPI Pins	1656
33.3.3	RSPI System Configuration Examples.....	1657
33.3.3.1	Single Master/Single Slave (with This LSI Acting as Master)	1657
33.3.3.2	Single Master/Single Slave (with This LSI Acting as Slave)	1658
33.3.3.3	Single Master/Multi-Slave (with This LSI Acting as Master)	1659
33.3.3.4	Single Master/Multi-Slave (with This LSI Acting as Slave)	1660
33.3.3.5	Multi-Master/Multi-Slave (with This LSI Acting as Master)	1661
33.3.3.6	Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Master)	1662
33.3.3.7	Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Slave)	1662
33.3.4	Transfer Format	1663
33.3.4.1	CPHA = 0	1663
33.3.4.2	CPHA = 1	1664
33.3.5	Data Format	1665
33.3.5.1	MSB First Transfer (32-Bit Data)	1665
33.3.5.2	MSB First Transfer (24-Bit Data)	1667
33.3.5.3	LSB First Transfer (32-Bit Data)	1669
33.3.5.4	LSB First Transfer (24-Bit Data)	1671
33.3.6	Communications Operating Mode.....	1673
33.3.6.1	Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)	1673
33.3.6.2	Transmit Operations Only (SPCR.TXMD = 1)	1674
33.3.7	Transmit Buffer Empty/Receive Buffer Full Interrupts	1675
33.3.8	Error Detection	1677
33.3.8.1	Overflow Error	1678
33.3.8.2	Parity Error	1679
33.3.8.3	Mode Fault Error	1680
33.3.9	Initializing RSPI	1681
33.3.9.1	Initialization by Clearing the SPE Bit	1681
33.3.9.2	System Reset	1681
33.3.10	SPI Operation	1682

33.3.10.1	Master Mode Operation	1682
33.3.10.2	Slave Mode Operation	1689
33.3.11	Clock Synchronous Operation.....	1693
33.3.12	Master Mode Operation.....	1693
33.3.13	Slave Mode Operation.....	1697
33.3.14	Error Handling.....	1700
33.3.15	Loopback Mode.....	1702
33.3.16	Self-Diagnosis of Parity Bit Function	1703
33.3.17	Interrupt Sources.....	1704
33.4	Usage Note.....	1704
33.4.1	Transmit Operation when Parity Function is Enabled in Master Mode.....	1704
34.	12-Bit A/D Converter (S12AD).....	1705
34.1	Overview.....	1705
34.2	Register Descriptions.....	1708
34.2.1	A/D Data Registers n (ADDRn) (n = 0 to 7).....	1709
34.2.2	A/D Control Register (ADCSR).....	1711
34.2.3	A/D Channel Select Register (ADANS)	1712
34.2.4	A/D-Converted Value Addition Mode Select Register (ADADS).....	1713
34.2.5	A/D-Converted Value Addition Count Select Register (ADADC).....	1714
34.2.6	A/D Control Extended Register (ADCER)	1715
34.2.7	A/D Start Trigger Select Register (ADSTRGR)	1716
34.3	Operation.....	1717
34.3.1	Scan Conversion.....	1717
34.3.2	Single-Cycle Scan Mode	1717
34.3.3	Continuous Scan Mode.....	1718
34.3.4	Analog Input Sampling and Scan Conversion Time	1719
34.3.5	Usage Example of A/D Data Register n (ADDRn) Automatic Clearing Function	1720
34.3.6	A/D-Converted Value Addition Function	1720
34.3.7	Starting Scan Conversion with External Trigger.....	1720
34.3.8	Starting Scan Conversion with Trigger from Peripheral Modules.....	1721
34.3.8.1	A/D Converter Activation with TRG0AN_0 and TRG0BN_0 of MTU	1721
34.3.8.2	A/D Converter Activation with TRGAN_0 and TRGAN_1 of MTU	1721
34.3.8.3	A/D Converter Activation with TRG0EN_0 and TRG0FN_0 of MTU	1723
34.3.8.4	A/D Converter Activation with TRG4ABN_0 and TRG4ABN_1 of MTU	1724
34.3.8.5	A/D Converter Activation with TMTRG0AN_0 and TMTRG0AN_1 of TMR	1725
34.4	Interrupt Sources and DMA Transfer Request	1726
34.4.1	Interrupt Request on Completion of Each Scan Conversion.....	1726
34.5	Usage Notes.....	1727
34.5.1	Selecting 12-Bit A/D Converter or 10-Bit A/D Converter.....	1727
34.5.2	Module Stop Function Setting	1727
34.5.3	Notes on Restarting A/D Conversion	1727

34.5.4	Notes on Disabling A/D Conversion.....	1727
34.5.5	Notes on Entering Low Power Consumption States	1727
34.5.6	Notes on Canceling Software Standby Mode.....	1727
34.5.7	Notes on Using A/D Converter and D/A Converter Simultaneously.....	1728
35.	10-Bit A/D Converter (ADa)	1729
35.1	Overview.....	1729
35.2	Register Descriptions.....	1734
35.2.1	A/D Data Register n (ADDRn) (n = A to D).....	1735
35.2.2	A/D Control/Status Register (ADCSR).....	1736
35.2.3	A/D Control Register (ADCR).....	1738
35.2.4	ADDRn Format Select Register (ADDPR).....	1740
35.2.5	A/D Sampling State Register (ADSSTR).....	1740
35.2.6	A/D Self-Diagnostic Register (ADDIAGR).....	1741
35.3	Operation.....	1742
35.3.1	Single Mode.....	1742
35.3.2	Scan Mode	1743
35.3.2.1	Continuous Scan Mode	1743
35.3.2.2	One-Cycle Scan Mode	1744
35.3.3	Input Sampling and A/D Conversion Time.....	1745
35.3.4	A/D Converter Activation by External Triggers	1747
35.3.5	A/D Converter Activation with TRG0AN_0 and TRG0BN_0 of MTU.....	1748
35.3.6	A/D Converter Activation with TRGAN_0 and TRGAN_1 of MTU.....	1749
35.3.7	A/D Converter Activation with TRG4ABN_0 and TRG4ABN_1 of MTU	1751
35.3.8	A/D Converter Activation with TMTRG0AN_0 of TMR.....	1752
35.4	Interrupt Source	1753
35.5	A/D Conversion Accuracy Definitions	1753
35.6	Usage Notes	1755
35.6.1	Module Stop Function Setting.....	1755
35.6.2	Notes on Disabling A/D Conversion.....	1755
35.6.3	Notes on Restarting A/D Conversion	1755
35.6.4	Notes on Entering Power-Down States	1755
35.6.5	Permissible Impedance of Signal Sources.....	1756
35.6.6	Factors Affecting Absolute Accuracy	1756
35.6.7	Ranges of Settings for Analog Power Supply and Other Pins	1757
35.6.8	Point for Caution Regarding Board Design.....	1757
35.6.9	Point for Caution Regarding Countermeasures for Noise	1758
35.6.10	Realizing High-Speed Conversion	1759
35.6.11	Selecting 12-Bit A/D Converter or 10-Bit A/D Converter.....	1759
35.6.12	Notes on Using A/D Converter and D/A Converter Simultaneously.....	1760

36.	D/A Converter	1761
36.1	Overview.....	1761
36.2	Register Descriptions.....	1763
36.2.1	D/A Data Register m (DADRm) (m = 0, 1)	1763
36.2.2	D/A Control Register (DACR).....	1764
36.2.3	DADRm Format Select Register (DADPR).....	1765
36.3	Operation.....	1766
36.4	Usage Notes	1767
36.4.1	Module Stop Function Setting.....	1767
36.4.2	Operation of the D/A Converter in Module Stop State	1767
36.4.3	Operation of the D/A Converter in Software Standby Mode	1767
36.4.4	Note on Entering Deep Software Standby Mode	1767
37.	RAM.....	1768
37.1	Overview.....	1768
37.2	Operation	1768
37.2.1	Data Retention	1768
37.2.2	Power-Down Function.....	1768
38.	ROM (Flash Memory for Code Storage)	1769
38.1	Overview.....	1769
38.2	Register Descriptions.....	1772
38.2.1	Flash Mode Register (FMODR).....	1773
38.2.2	Flash Access Status Register (FASTAT)	1774
38.2.3	Flash Access Error Interrupt Enable Register (FAEINT)	1776
38.2.4	FCU RAM Enable Register (FCURAME).....	1777
38.2.5	Flash Status Register 0 (FSTATR0).....	1778
38.2.6	Flash Status Register 1 (FSTATR1).....	1781
38.2.7	Flash Ready Interrupt Enable Register (FRDYIE).....	1782
38.2.8	Flash P/E Mode Entry Register (FENTRYR)	1783
38.2.9	Flash Protection Register (FPROTR)	1785
38.2.10	Flash Reset Register (FRESETR).....	1786
38.2.11	FCU Command Register (FCMDR).....	1787
38.2.12	FCU Processing Switching Register (FCPSR).....	1788
38.2.13	Flash P/E Status Register (FPESTAT)	1789
38.2.14	Peripheral Clock Notification Register (PCKAR).....	1790
38.2.15	Flash Write Erase Protection Register (FWEPROR)	1791
38.3	Configuration of Memory Mats for the ROM	1792
38.4	Block Configuration.....	1793
38.5	Operating Modes Associated with the ROM.....	1794
38.6	Programming and Erasing the ROM.....	1796
38.6.1	FCU Modes.....	1796

38.6.1.1	ROM Read Modes	1797
38.6.1.2	ROM P/E Modes	1797
38.6.2	FCU Commands	1798
38.6.3	Connections between FCU Modes and Commands	1800
38.6.4	FCU Command Usage.....	1801
38.6.4.1	Mode Transitions	1801
38.6.4.2	Programming and Erasure Procedures	1805
38.6.4.3	Error Processing	1814
38.6.4.4	Suspension and Resumption	1815
38.7	Suspending Operation	1818
38.7.1	Suspension during Programming.....	1818
38.7.2	Suspension during Erasure (Suspension Priority Mode).....	1819
38.7.3	Suspension during Erasure (Erasure Priority Mode).....	1820
38.8	Protection.....	1821
38.8.1	Software Protection	1821
38.8.2	Error Protection	1821
38.9	Boot Mode	1823
38.9.1	System Configuration	1823
38.9.2	ID Code Protection	1824
38.9.3	UB Code A	1826
38.9.4	State Transitions in Boot Mode.....	1827
38.9.5	Automatic Adjustment of the Bit Rate	1829
38.9.6	Inquiry/Selection Host Command Wait State	1830
38.9.7	ID Code Wait State.....	1841
38.9.8	Programming/Erasure Host Command Wait State.....	1842
38.10	USB (User) Boot Mode	1850
38.10.1	Features.....	1851
38.10.2	State Transitions	1852
38.10.3	Points to Note When Running Programs in USB Boot Mode.....	1853
38.11	ID Code Protection on Connection of the On-Chip Debugger	1854
38.12	ROM Code Protection.....	1855
38.13	Usage Notes	1856
39.	Data Flash Memory (Flash Memory for Data Storage)	1858
39.1	Overview.....	1858
39.2	Register Descriptions.....	1861
39.2.1	Flash Mode Register (FMODR).....	1862
39.2.2	Flash Access Status Register (FASTAT)	1863
39.2.3	Flash Access Error Interrupt Enable Register (FAEINT)	1865
39.2.4	Data Flash Read Enable Register 0 (DFLRE0).....	1866
39.2.5	Data Flash Read Enable Register 1 (DFLRE1).....	1867
39.2.6	Data Flash Programming/Erasure Enable Register 0 (DFLWE0).....	1868

39.2.7	Data Flash Programming/Erasure Enable Register 1 (DFLWE1).....	1869
39.2.8	Flash P/E Mode Entry Register (FENTRYR)	1870
39.2.9	Data Flash Blank Check Control Register (DFLBCCNT)	1872
39.2.10	Data Flash Blank Check Status Register (DFLBCSTAT).....	1873
39.3	Configuration of Memory Mat for the Data Flash Memory.....	1874
39.4	Block Configuration.....	1874
39.5	Operating Modes Associated with the Data Flash.....	1875
39.6	Programming and Erasing the Data Flash Memory.....	1876
39.6.1	FCU Modes.....	1876
39.6.1.1	ROM P/E Modes	1877
39.6.1.2	ROM/Data Flash Read Mode	1877
39.6.1.3	Data Flash P/E Modes	1877
39.6.2	FCU Commands	1878
39.6.3	Connections between FCU Modes and Commands	1880
39.6.4	FCU Command Usage.....	1881
39.7	Protection.....	1885
39.7.1	Software Protection	1885
39.7.2	Error Protection	1886
39.8	Boot Mode	1887
39.8.1	Inquiry/Selection Host Commands.....	1887
39.8.2	Programming/Erasing Host Commands	1889
39.9	Usage Notes	1891
40.	Boundary Scan	1892
40.1	Features.....	1892
40.2	Register Descriptions.....	1893
40.2.1	Instruction Register (JTIR).....	1894
40.2.2	ID Code Register (JTID)	1894
40.2.3	Bypass Register (JTBPR).....	1895
40.2.4	Boundary Scan Register (JTBSR).....	1895
40.3	Operations	1907
40.3.1	TAP Controller	1907
40.3.2	List of Commands	1908
40.4	Usage Notes	1909
41.	Electrical Characteristics.....	1911
41.1	Absolute Maximum Ratings	1911
41.2	DC Characteristics.....	1912
41.3	AC Characteristics	1916
41.3.1	Clock Timing	1917
41.3.2	Control Signal Timing	1921
41.3.3	Bus Timing	1922
41.3.4	EXDMAC Timing	1935

41.3.5	Timing of On-Chip Peripheral Modules.....	1936
41.4	USB Characteristics.....	1957
41.5	A/D Conversion Characteristics.....	1958
41.6	D/A Conversion Characteristics.....	1959
41.7	Power-on Reset Circuit, Voltage Detection Circuit Characteristics	1960
41.8	Oscillation Stop Detection Timing.....	1962
41.9	ROM (Flash Memory for Code Storage) Characteristics	1963
41.10	Data Flash (Flash Memory for Data Storage) Characteristics	1964
Appendix 1.	Port States in Each Processing Mode	1966
Appendix 2.	Package Dimensions	1975
REVISION HISTORY	1979

100 MHz 32-bit RX MCU with FPU, 165 DMIPS, up to 512-Kbyte Flash, Ethernet, USB 2.0 Full-Speed Host/Function/OTG, CAN, 12-bit ADC, TFT-LCD, RTC, up to 14 communication channels

Features

■32-bit RX CPU Core

- Delivers 165 DMIPS at a maximum operating frequency of 100 MHz
- Single Precision 32-bit IEEE-754 Floating Point
- Accumulator: 32 × 32 to 64-bit result, one instruction
- Mult/Divide Unit, 32 × 32 Multiply in one CPU clock for multiple instructions
- Interrupt response in as few as 5 CPU clock cycles
- CISC-Harvard Architecture with 5-stage pipeline
- Variable length instructions, ultra compact code
- Supports the Memory Protection Unit (MPU)
- Background JTAG debug plus high-speed trace

■Low Power Design and Architecture

- 2.7V to 3.6V operation from a single supply
- 480 μA/MHz Run Mode with all peripherals on
- Deep Software Standby Mode with RTC
- Four low power modes

■Main Flash Memory, no Wait-State

- 100 MHz operation, 10 nsec read cycle
- No wait states for read at full CPU speed
- 256K, 384K, 512K Byte size options
- For Instructions or Operands
- Programming from USB, SCI, JTAG, user code

■Data Flash Memory

- Up to 32K Bytes with 30K Erase Cycles
- Background Erase/Program does not stall CPU

■SRAM, no Wait-State

- 64K or 96K Byte size options
- For Operands or Instructions
- Back-up retention in Deep Software Standby Mode

■DMA

- Four fully programmable internal DMA channels
- Two EXDMA channels for external-to-external transfers
- Data Transfer Controller (DTC)

■Reset and Supply Management

- Power-On Reset (POR) monitor/generator
- Low Voltage Detect (LVD) with precision setting

■System Clocking with Clock Monitoring

- External crystal, 8 MHz to 14 MHz to Internal PLL
- PLL source to system, USB, and Ethernet
- Internal 125 kHz LOCO for IWDTC
- External crystal, 32 kHz for RTC

■Real Time Clock

- Full calendar function, BCD format

■Two Independent Watchdog Timers

- 125-kHz LOCO operation



■Up to 14 Communication Interfaces

- USB 2.0 Full-Speed interfaces with PHY (2ch)
Supports Host/Function/OTG
10 endpoints for types: Control, Interrupt, Bulk, Isochronous
- Ethernet MAC 10/100 Mbps, Half or Full Duplex Supported. (1ch)
Dedicated DMA with 2-Kbyte transmit and receive FIFOs.
RMII or MII interface to external PHY
- CAN ISO11898-1, supports 32 mailboxes (1ch)
- SCI channels: Asynchronous, clock sync, smartcard, and 9-bit modes (6ch)
- I²C interfaces up to 1M bps, SMBus support (2ch)
- RSPI (2ch)

■External Address Space

- Eight CS areas (8 × 16 Mbytes)
- 128-Mbyte SDRAM area
- 8-/16-/32-bit bus space selectable for each area

■TFT-LCD up to WQVGA resolution

■Up to 20 Extended Function Timers

- 16-bit MTU2
Input capture, Output Compare, PWM output, phase count mode (12ch)
- 8-bit TMR (4ch)
- 16-bit CMT (4ch)

■1-MHz ADC units with two combination choices

- 12-bit × 8 ch. unit with single sample/hold circuit
- or (2) 10-bit × 4 ch units each with a sample/hold circuit
- AD-converted value addition mode (12-bit A/D converter)

■10-bit DAC, 2 channels

■Up to 128 GPIO

- 5V tolerant, Open-Drain, Internal Pull-up

■Operation Temp

- -40°C to +85°C

1. Overview

1.1 Outline of Specifications

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

Table 1.1 Outline of Specifications (1 / 4)

Classification	Module/Function	Description
CPU	CPU	<ul style="list-style-type: none"> Maximum operating frequency: 100 MHz 32-bit RX CPU Minimum instruction execution time: One instruction per state (cycle of the system clock) Address space: 4-Gbyte linear Register set of the CPU <ul style="list-style-type: none"> General purpose: Sixteen 32-bit registers Control: Nine 32-bit registers Accumulator: One 64-bit register Basic instructions: 73 Floating-point instructions: 8 DSP instructions: 9 Addressing modes: 10 Data arrangement <ul style="list-style-type: none"> Instructions: Little endian Data: Selectable as little endian or big endian On-chip 32-bit multiplier: 32 x 32 → 64 bits On-chip divider: 32 / 32 → 32 bits Barrel shifter: 32 bits Memory-protection unit (MPU)
	FPU	<ul style="list-style-type: none"> Single precision (32-bit) floating point Data types and floating-point exceptions in conformance with the IEEE754 standard
Memory	ROM	<ul style="list-style-type: none"> ROM capacity: 512 Kbytes (max.) Two on-board programming modes <ul style="list-style-type: none"> Boot mode (The user MAT is programmable via the SCI and USB.) User program mode Parallel programmer mode (for off-board programming)
	RAM	RAM capacity: 96 Kbytes (max.)
	Data flash	Data flash capacity: 32 Kbytes
MCU operating modes		<ul style="list-style-type: none"> Single-chip mode, on-chip ROM enabled expansion mode, and on-chip ROM disabled expansion mode (software switching)
Clock	Clock generation circuit	<ul style="list-style-type: none"> Two circuits: Main clock oscillator and subclock oscillator Internal oscillator: Low-speed on-chip oscillator Structure of a PLL frequency synthesizer and frequency divider for selectable operating frequency Oscillation stoppage detection Independent frequency-division and multiplication settings for the system clock (ICLK), peripheral module clock (PCLK), and external bus clock (BCLK) <ul style="list-style-type: none"> The CPU and other bus masters run in synchronization with the system clock (ICLK): 8 to 100 MHz Peripheral modules run in synchronization with the peripheral module clock (PCLK): 8 to 50 MHz Devices connected to the external bus run in synchronization with the external bus clock (BCLK pin): 8 to 50 MHz*1
Reset		<ul style="list-style-type: none"> Pin reset, power-on reset, voltage-monitoring reset, watchdog timer reset, independent watchdog timer reset, and deep software standby reset
Voltage detection circuit		<ul style="list-style-type: none"> When the voltage on VCC falls below the voltage detection level (Vdet), an internal reset or internal interrupt is generated.
Low power consumption	Low power consumption facilities	<ul style="list-style-type: none"> Module stop function Four low power consumption modes <ul style="list-style-type: none"> Sleep mode, all-module clock stop mode, software standby mode, and deep software standby mode

Table 1.1 Outline of Specifications (2 / 4)

Classification	Module/Function	Description
Interrupt	Interrupt control unit	<ul style="list-style-type: none"> Peripheral function interrupts: 146 sources External interrupts: 16 (pins IRQ0 to IRQ15) Non-maskable interrupts: 3 (the NMI pin, oscillation stop detection interrupt, and voltage-monitoring interrupt) Sixteen levels specifiable for the order of priority
	User break controller (as an optional function)	<ul style="list-style-type: none"> Two breakpoint channels Address breaks in fetch cycles are specifiable (enabling ROM correction)
External bus extension		<ul style="list-style-type: none"> The external address space can be divided into nine areas (CS0 to CS7, SDCS), each with independent control of access settings. Capacity of each area: 16 Mbytes (CS0 to CS7), 128 Mbytes (SDCS) A chip-select signal (CS0# to CS7#, SDCS#) can be output for each area. Each area is specifiable as an 8-, 16-, or 32-bit bus space (however, only 176-pin versions support 32-bit bus spaces). The data arrangement in each area is selectable as little or big endian (only for data). SDRAM interface connectable Bus format: Separate buses Wait control Write buffer facility
DMA	DMA controller	<ul style="list-style-type: none"> 4 channels Three transfer modes: Normal transfer, repeat transfer, and block transfer Activation sources: Software trigger, external interrupts, and interrupt requests from peripheral functions
	EXDMA controller	<ul style="list-style-type: none"> 2 channels Four transfer modes: Normal transfer, repeat transfer, block transfer, and cluster transfer Single-address transfer enabled with the EDACK signal Capable of direct data transfer to TFT LCD panels Activation sources: Software trigger, external DMA transfer requests (EDREQ), and interrupt requests from peripheral functions
	Data transfer controller	<ul style="list-style-type: none"> Three transfer modes: Normal transfer, repeat transfer, and block transfer Activation sources: Software trigger, external interrupts and interrupt requests from peripheral functions
I/O ports	Programmable I/O ports	<ul style="list-style-type: none"> I/O ports for the 176-pin LFBGA/145-pin TFLGA/144-pin LQFP/100-pin LQFP/85-pin TFLGA I/O pins: 126/103/103/72/58 Input pins: 2/2/2/2/2 Pull-up resistors: 56/44/44/40/28 Open-drain outputs: 35/33/33/27/23 5-V tolerance: 11/11/11/7/6
Timers	Multi-function timer pulse unit	<ul style="list-style-type: none"> (16 bits x 6 channels) x 2 units Time bases for the 12 16-bit timer channels can be provided via up to 32 pulse-input/output lines and six pulse-input lines Select from among eight counter-input clock signals for each channel (PCLK/1, PCLK/4, PCLK/16, PCLK/64, MTCLKA, MTCLKB, MTCLKC, MTCLKD) other than channel 5, for which only four signals are available. Input capture function 21 output compare/input capture registers Pulse output mode Complementary PWM output mode Reset synchronous PWM mode Phase-counting mode Generation of triggers for A/D converter conversion
	Port output enable	<ul style="list-style-type: none"> Controls the high-impedance state of the MTU's waveform output pins

Table 1.1 Outline of Specifications (3 / 4)

Classification	Module/Function	Description
Timers	Programmable pulse generator	<ul style="list-style-type: none"> • (4 bits x 4 groups) x 2 units • Pulse output with the MTU output as a trigger • Maximum of 32-bit pulse output possible
	8-bit timers	<ul style="list-style-type: none"> • (8 bits x 2 channels) x 2 units • Select from among seven internal clock signals (PCLK, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192) and one external clock signal • Capable of output of pulse trains with desired duty cycles or of PWM signals • The 2 channels of each unit can be cascaded to create a 16-bit timer • Generation of triggers for A/D converter conversion • Capable of generating baud-rate clocks for SCI5 and SCI6
	Compare match timer	<ul style="list-style-type: none"> • (16 bits x 2 channels) x 2 units • Select from among four internal clock signals (PCLK/8, PCLK/32, PCLK/128, PCLK/512)
	Watchdog timer	<ul style="list-style-type: none"> • 8 bits x 1 channel • Select from among eight counter-input clock signals (PCLK/4, PCLK/64, PCLK/128, PCLK/512, PCLK/2048, PCLK/8192, PCLK/32768, PCLK/131072) • Switchable between watchdog timer mode and interval timer mode
	Independent watchdog timer	<ul style="list-style-type: none"> • 14 bits x 1 channel • Counter-input clock: Dedicated on-chip oscillator
Realtime clock		<ul style="list-style-type: none"> • Clock source: Subclock • Time/calendar <p>Interrupt sources: Alarm interrupt, periodic interrupt, and carry interrupt</p>
Communication function	Ethernet controller	<ul style="list-style-type: none"> • Input and output of Ethernet/IEEE 802.3 frames • Transfer at 10 or 100 Mbps • Full- and half-duplex modes • MII (Media Independent Interface) or RMII (Reduced Media Independent Interface) as defined in IEEE 802.3u • Detection of Magic Packets™* or output of a "wake-on-LAN" signal (WOL) • Compliance with flow control as defined in IEEE 802.3x standards <p>Note: * Magic Packet™ is a registered trademark of Advanced Micro Devices, Inc.</p>
	DMA controller for Ethernet controller	<ul style="list-style-type: none"> • Alleviation of CPU loads by the descriptor control method • Transmission FIFO: 2 Kbytes; Reception FIFO: 2 Kbytes
	USB 2.0 host/function module	<ul style="list-style-type: none"> • Includes a UDC (USB Device Controller) and transceiver for USB 2.0 • Single port (176-pin products: two ports) • Compliance with the USB 2.0 specification • Transfer rate: Full speed (12 Mbps) • Self-power mode and bus power are selectable • OTG (On the Go) operation is possible • Incorporates 2 Kbytes of RAM as a transfer buffer
	Serial communications interfaces	<ul style="list-style-type: none"> • 6 channels • Serial communications modes: Asynchronous, clock synchronous, and smart-card interface • Multi-processor communications function • On-chip baud rate generator allows selection of the desired bit rate • Choice of LSB-first or MSB-first transfer • Average transfer rate clock can be input from TMR timers for SCI5 and SCI6

Table 1.1 Outline of Specifications (4 / 4)

Classification	Module/Function	Description
Communication function	I ² C bus interfaces	<ul style="list-style-type: none"> 2 channels (100-pin version: 1 channel) Communications formats I²C bus format/SMBus format Master/slave selectable (For multi-master operation)
	CAN module	<ul style="list-style-type: none"> 1 channel 32 mailboxes
	Serial peripheral interfaces	<ul style="list-style-type: none"> 2 channels RSPI 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) Capable of handling serial transfer as a master or slave Data formats Switching between MSB first and LSB first The number of bits in each transfer can be changed to any number of bits from 8 to 16, or to 20, 24, or 32 bits. 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) Buffered structure Double buffers for both transmission and reception Max. transfer rate In master mode: 18 Mbps In slave mode: 6.25 Mbps
12-bit A/D converter 10-bit A/D converter		<ul style="list-style-type: none"> 12 bits x 1 unit (1 unit x 8 channels) or 10 bits x 2 units (2 units x 4 channels); 12- and 10-bit A/D converters can be exclusively used. 10- or 12-bit resolution Conversion time: 1.0 μs per channel (in operation with PCLK at 50 MHz) Two operating modes Single mode Scan mode (one-cycle scan mode or continuous scan mode) Sample-and-hold function Three ways to start A/D conversion Conversion can be started by software, a conversion start trigger from a timer (MTU or TMR), or an external trigger signal. Self-diagnostic functions
D/A converter		<ul style="list-style-type: none"> 2 channels (1 channel for 100-pin products) 10-bit resolution Output voltage: 0 V to VREFH
CRC calculator		<ul style="list-style-type: none"> CRC code generation for arbitrary amounts of data in 8-bit units Select any of three generating polynomials: $X^8 + X^2 + X + 1$, $X^{16} + X^{15} + X^2 + 1$, or $X^{16} + X^{12} + X^5 + 1$. Generation of CRC codes for use with LSB-first or MSB-first communications is selectable.
Operating frequency		8 to 100 MHz
Power supply voltage		VCC = PLLVCC = AVCC = 2.7 to 3.6V, VREFH = 2.7 to AVCC
Operating temperature		-40 to +85°C
Package		176-pin LFBGA (PLBG0176GA-A), 145-pin TFLGA (PTLG0145JB-A), 144-pin LQFP (PLQP0144KA-A), 100-pin LQFP (PLQP0100KB-A)*2 85-pin TFLGA (PTLG0085JA-A)*2,*3

Note 1. For products in the 100-pin LQFP and 85-pin TFLGA, the synchronizing frequency is 8 to 25 MHz.

Note 2. The 100-pin LQFP and 85-pin TFLGA do not support the SDRAM area controller and EXDMA controller.

Note 3. The 85-pin TFLGA does not support the port-output enabling.

Table 1.2 Functions of RX62N Group and RX621 Group Products

Functions		RX62N Group								RX621 Group				
		R5F562NxBxxx*				R5F562NxAxxx*				R5F5621xBxxx*				
Package		176-pin LFBGA	145-pin TFLGA	144-pin LQFP	100-pin LQFP	176-pin LFBGA	145-pin TFLGA	144-pin LQFP	100-pin LQFP	176-pin LFBGA	145-pin TFLGA	144-pin LQFP	100-pin LQFP	85-pin TFLGA
External bus	SDRAM area controller		○		—		○		—		○		—	
DMA	DMA controller		○				○				○			
	EXDMA controller		○		—		○		—		○		—	
	Data transfer controller		○				○				○			
Timers	Multi-function timer pulse unit		○				○				○			
	Port output enable		○				○			○			—	
	Programmable pulse generator		○				○				○			
	8-bit timers		○				○				○			
	Compare match timer		○				○				○			
	Realtime clock		○				○				○			
	Watchdog timer		○				○				○			
	Independent watchdog timer		○				○				○			
Communication function	Ethernet controller/ DMA controller for Ethernet controller		○				○				—			
	USB 2.0 host/function module		○				○				○			
	Serial communications interfaces		○				○				○			
	I ² C bus interfaces		○				○				○			
	CAN module		○				—				○			
	Serial peripheral interfaces		○				○				○			
A/D converter		○				○				○				
D/A converter		○				○				○				
CRC calculator		○				○				○				

[Legend]

○: Supported, —: Not supported

Note: * For details on part numbers, see Table 1.3.

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.

Table 1.3 List of Products

Group	Part No.	Package	ROM Capacity	RAM Capacity	Data Flash	Operating Frequency (Max.)
RX62N	R5F562N8BDBG	PLBG0176GA-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F562N8BDLE	PTLG0145JB-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F562N8BDFB	PLQP0144KA-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F562N8BDFF	PLQP0100KB-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F562N7BDBG	PLBG0176GA-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F562N7BDLE	PTLG0145JB-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F562N7BDFB	PLQP0144KA-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F562N7BDFF	PLQP0100KB-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F562N8ADBG	PLBG0176GA-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F562N8ADLE	PTLG0145JB-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F562N8ADFB	PLQP0144KA-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F562N8ADFF	PLQP0100KB-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F562N7ADBG	PLBG0176GA-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F562N7ADLE	PTLG0145JB-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F562N7ADFB	PLQP0144KA-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F562N7ADFF	PLQP0100KB-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
RX621	R5F56218BDBG	PLBG0176GA-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F56218BDLE	PTLG0145JB-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F56218BDFB	PLQP0144KA-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F56218BDFF	PLQP0100KB-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F56218BDLD	PTLG0085JA-A	512 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F56217BDBG	PLBG0176GA-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56217BDLE	PTLG0145JB-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56217BDFB	PLQP0144KA-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56217BDFF	PLQP0100KB-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56217BDLD	PTLG0085JA-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56216BDBG	PLBG0176GA-A	256 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56216BDLE	PTLG0145JB-A	256 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56216BDFB	PLQP0144KA-A	256 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56216BDFF	PLQP0100KB-A	256 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56216BDLD	PTLG0085JA-A	256 Kbytes	64 Kbytes	32 Kbytes	100 MHz

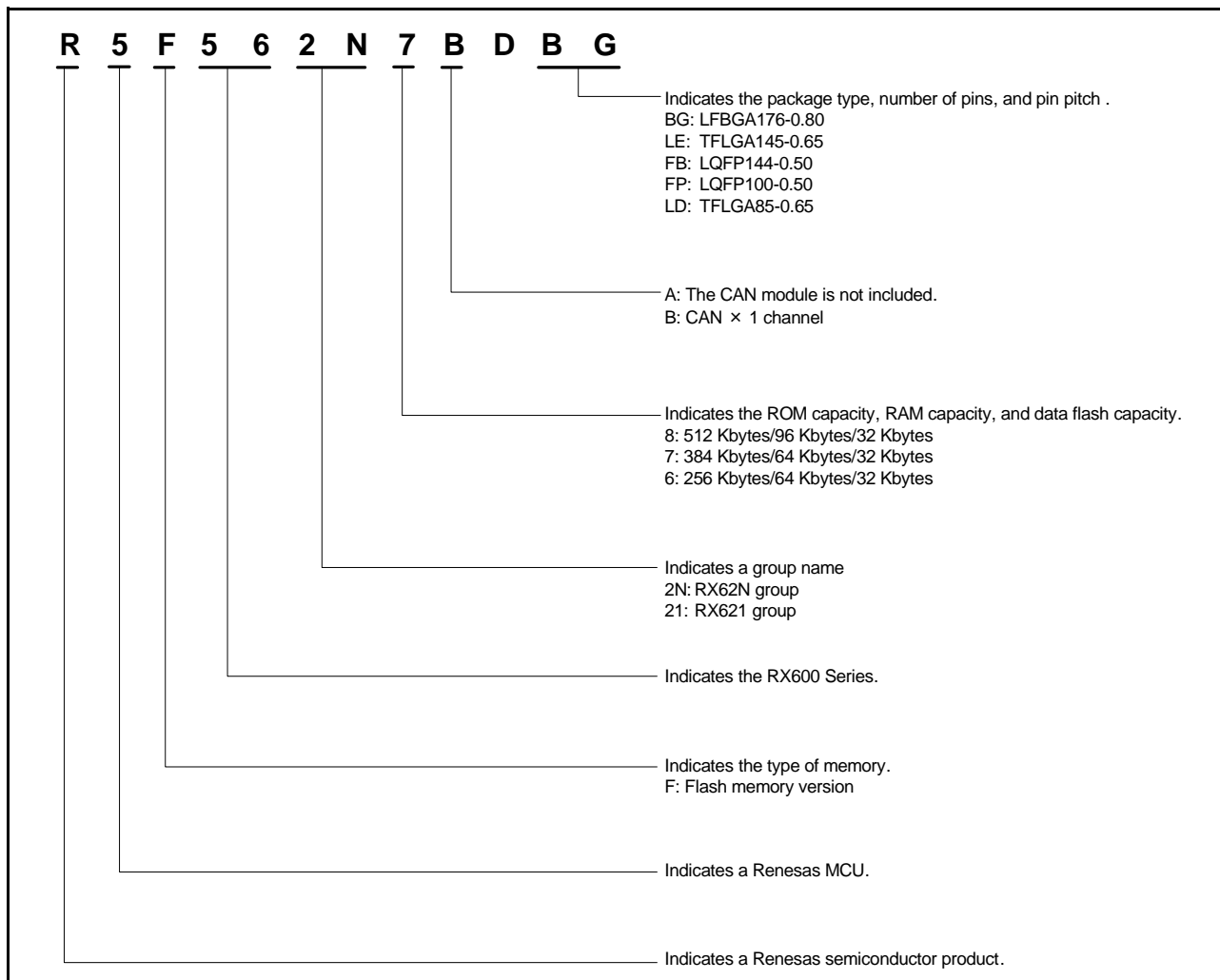


Figure 1.1 How to Read the Product Part No.

1.3 Block Diagram

Figure 1.2 shows a block diagram.

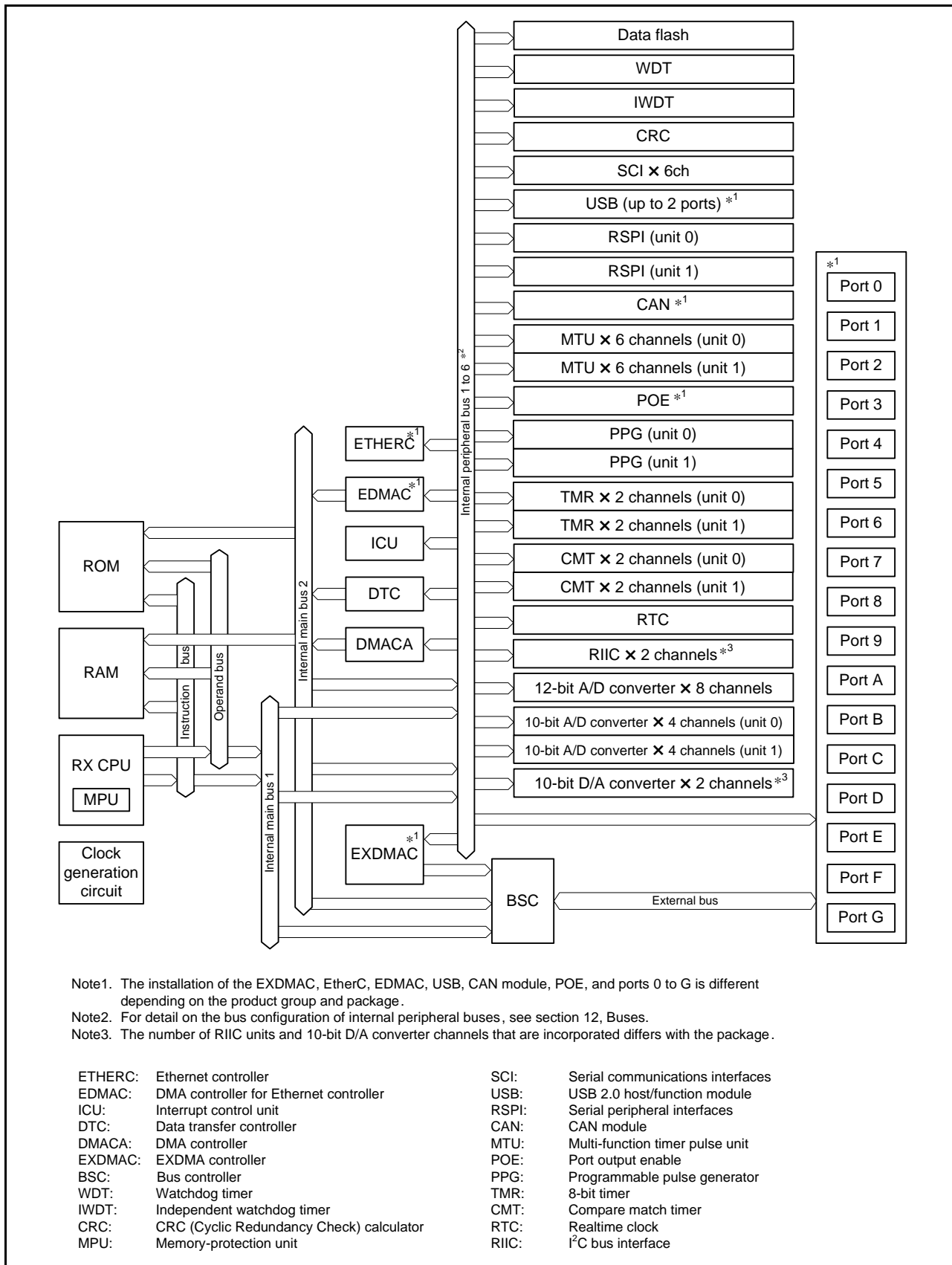


Figure 1.2 Block Diagram

1.4 Pin Assignments

Figure 1.3 to Figure 1.9 show the pins assignments. Table 1.4 to Table 1.8 show the list of pins and pin functions.

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R									
15	PE1	P70	PE6	P65	P67	PG5	PA1	PA3	PA6	PB0	VCC	PB2	PB5	PB7	P75	15								
14	P63	PE2	PE5	PE7	P66	PA0	PG6	PA4	PA7	P72	PB3	PB6	P73	PC1	P77	14								
13	P61	P64	PE3	PE4	VCC	PG3	VCC	PA2	PA5	P71	PB4	VCC	P74	P76	P80	13								
12	PD7	P62	PE0	VSS	PG2	PG4	VSS	PG7	VSS	PB1	VSS	PC0	PC2	PC4	PC7	12								
11	PG0	P60	VCC	VSS	RX62N Group RX621 Group PLBG0176GA-A (176-pin LFBGA) (Upper perspective view)								P81	PC3	P82	P83	11							
10	PD4	PD6	PD5	PG1									PC6	PC5	P50	P53	10							
9	PD3	P97	VCC	VSS									VSS	VCC	P84	P85	9							
8	PD2	P96	P94	P95									P51	P52	VCC_USB	USB1_DP	8							
7	PD0	PD1	P92	P93									P54	P10	P56	USB1_DM	7							
6	P90	P91	VCC	VSS									P55	P57	VCC_USB	VSS_USB	6							
5	P46	P47	P40	P43									P11	P15	P13	USB0_DP	5							
4	P45	P44	P07	P41									VSS	VSS	MDE	RES#	P34	PF4	P30	VSS	P17	P14	USB0_DM	4
3	P42	VREFL	P05	VCC									BSCANP	VCL	MD0	VCC	PF3	PF0	VCC	P22	P20	P16	P12	3
2	AVCC	VREFH	P03	P01	CNVSS	WDTOVF#	MD1	P35	P32	P31	P27	P25	P23	PLLVCC	PLLVSS	2								
1	AVSS	P02	P00	EMLE	XCIN	XCOUT	VSS	XTAL	EXTAL	P33	PF2	PF1	P26	P24	P21	1								
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R									

■ : NC pin

Figure 1.3 Pin Assignment of the 176-Pin LFBGA

	A	B	C	D	E	F	G	H	J	K	L	M	N	
13	P64	PE4	P70	PE6	P66	PA2	PA4	PA7	P72	PB3	PB6	VSS	P74	13
12	P62	PE1	PE3	PE7	PA0	VCC	PA6	PB1	PB5	PC0	VCC	PC1	P76	12
11	P60	PE2	PE5	VCC	P67	PA3	PA5	P71	PB4	P73	P75	PC2	PC4	11
10	PD6	PE0	P63	VSS	P65	PA1	VSS	PB0	PB2	PB7	P77	P80	PC5	10
9	PD3	VSS	P61	VCC	RX62N Group RX621 Group PTLG0145JB-A (145-pin TFLGA) (Upper perspective view)					PC3	P81	PC6	VCC	9
8	PD0	PD5	PD7	PD4						P82	P83	P50	P51	8
7	P91	PD1	PD2	P93						PC7	P52	P55	P54	7
6	P47	P90	P92	VSS						VSS	P56	VSS_USB	USB0_DP	6
5	P44	P45	P46	VCC	NC	P53	VCC_USB	P14	USB0_DM	5				
4	P42	P40	P41	P43	BSCANP	MDE	MD0	RES#	P32	P26	P12	P15	P13	4
3	VREFL	VREFH	VSS	P02	P00	WDTOVF#	MD1	VCC	P35	P31	P17	PLLVCC	PLLVSS	3
2	AVCC	P07	P05	VCC	VSS	XCOUT	VSS	P34	P27	P24	P22	P20	P16	2
1	AVSS	P03	P01	EMLE	VCL	XCIN	XTAL	EXTAL	P33	P30	P25	P23	P21	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	

: NC pin

Figure 1.4 Pin Assignment of the 145-Pin TFLGA

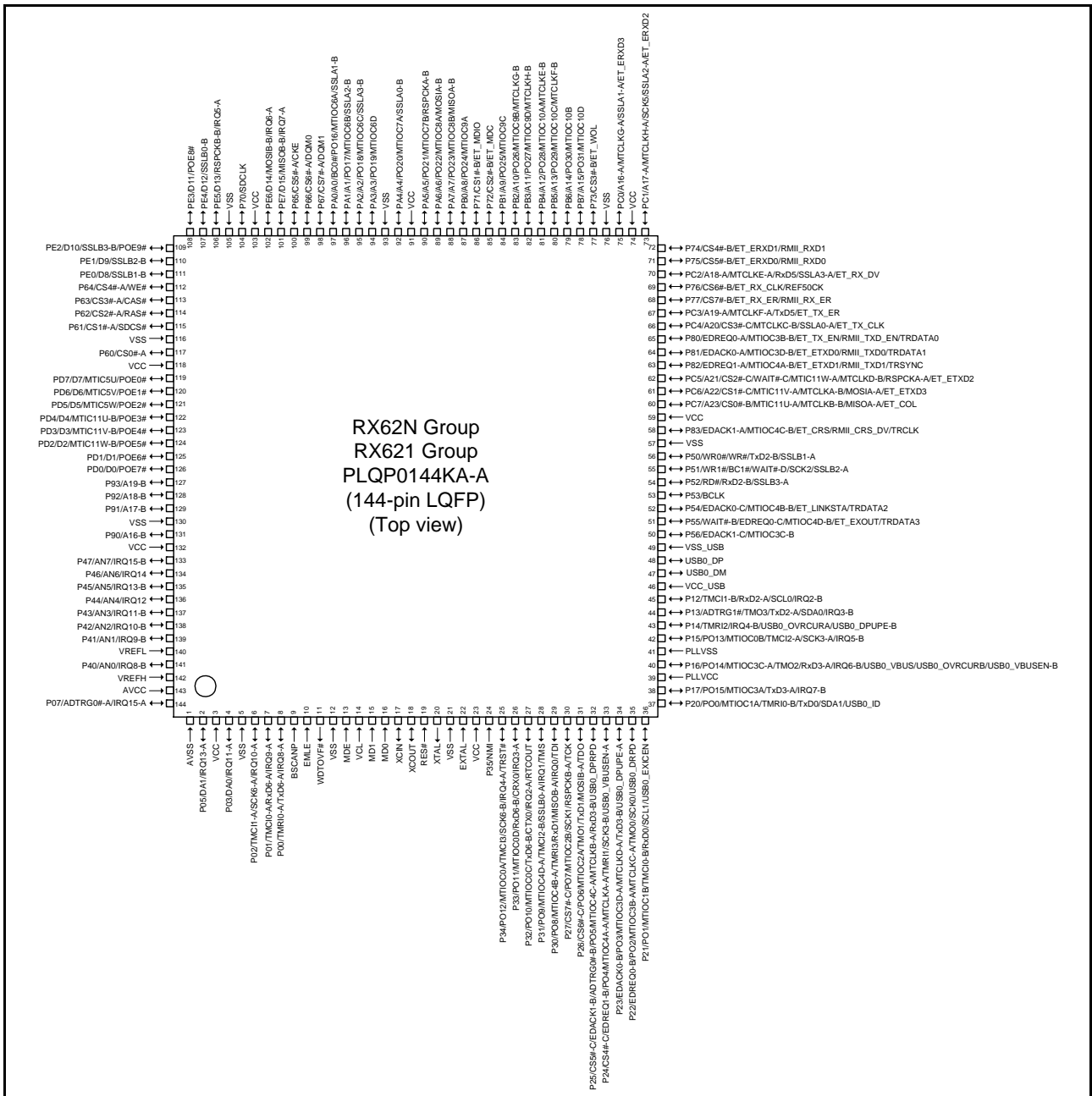


Figure 1.5 Pin Assignment of the 144-Pin LQFP

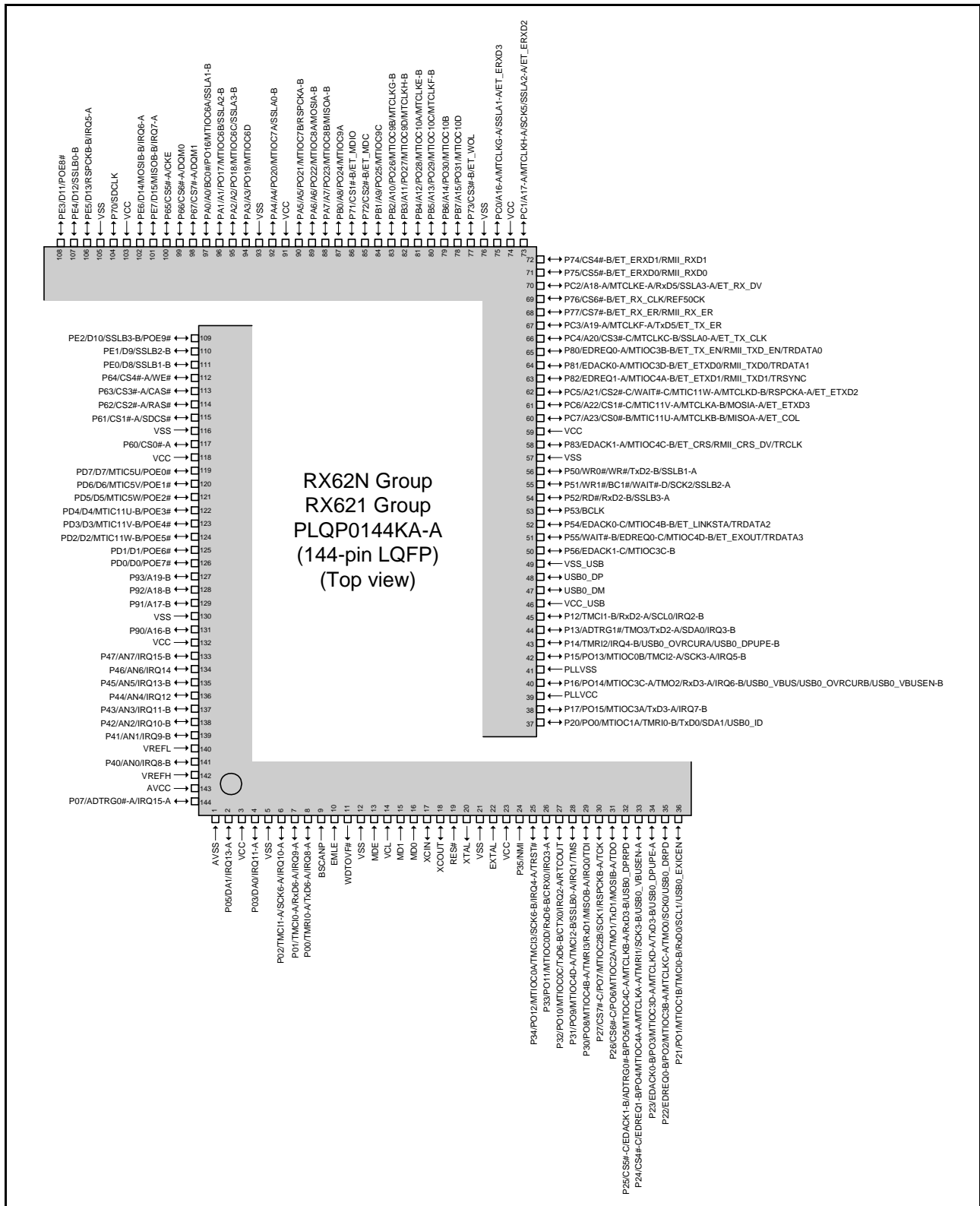


Figure 1.6 Pin Assignment of the 144-Pin LQFP (Assistance Diagram)

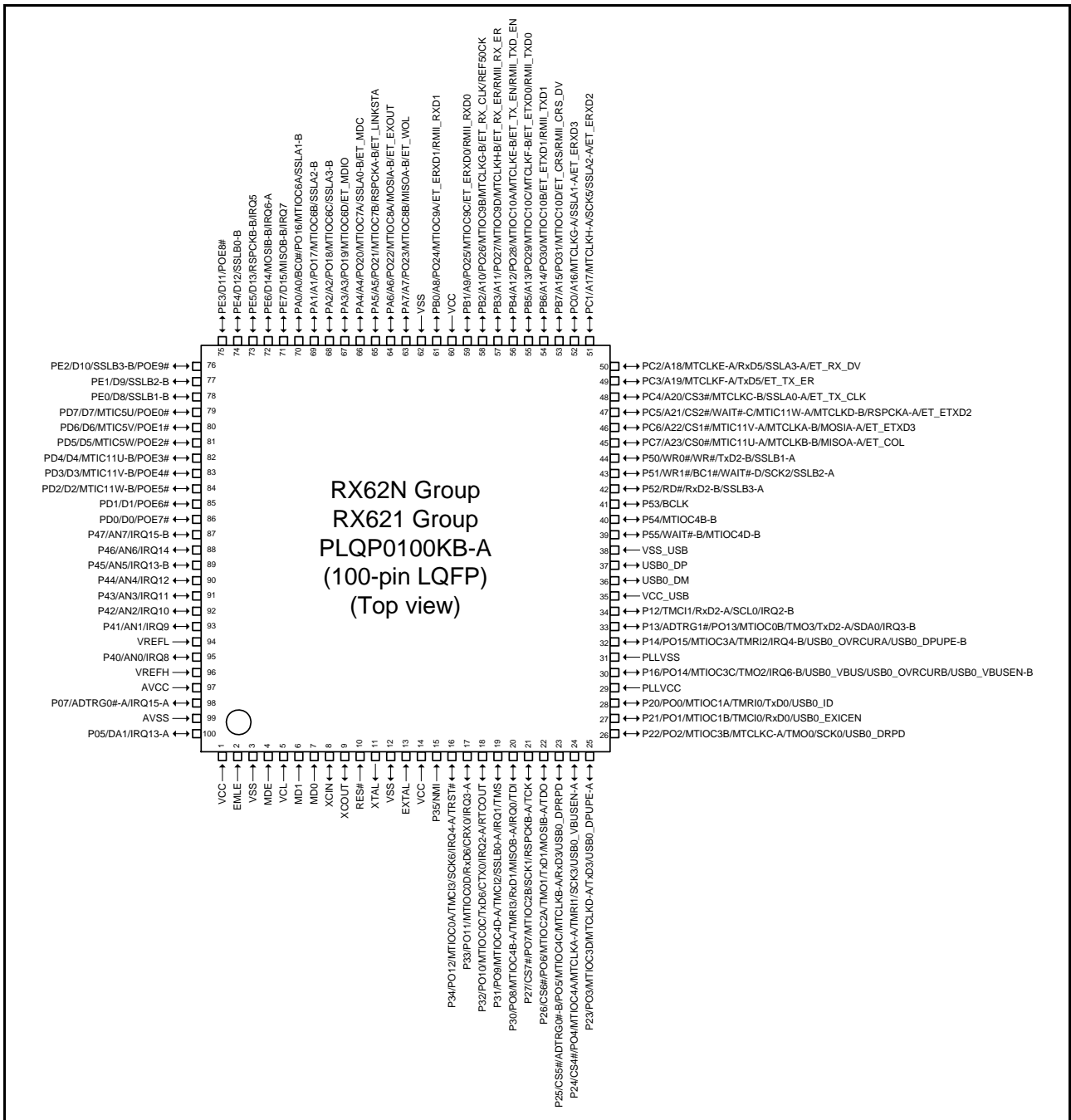


Figure 1.7 Pin Assignment of the 100-Pin LQFP

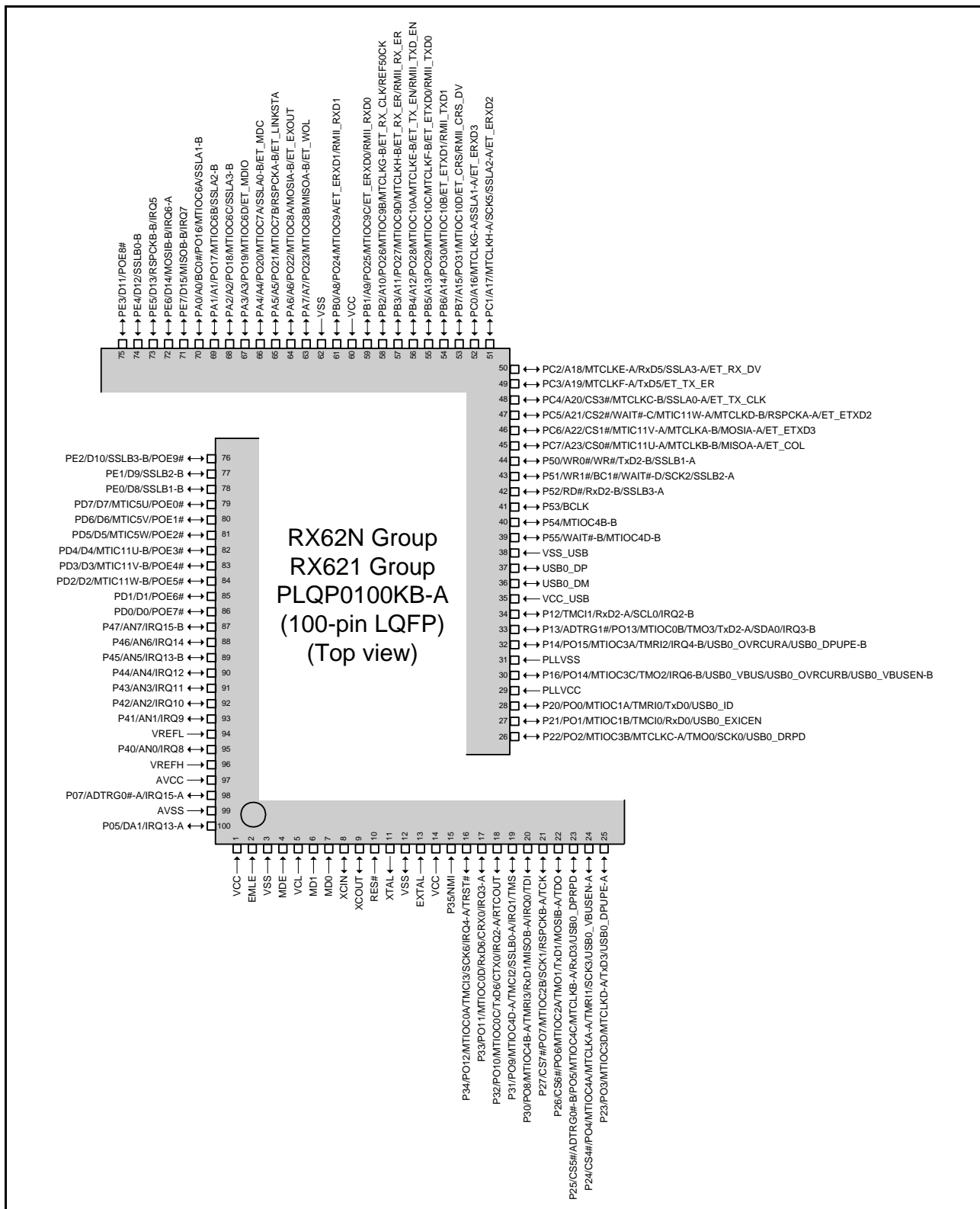


Figure 1.8 Pin Assignment of the 100-Pin LQFP (Assistance Diagram)

	A	B	C	D	E	F	G	H	J	K	
10	PD6	PA1	PA0	PA2	PA4	PA7	PB1	PB4	PC0	PC1	10
9	PD7	PA3	PA5	PA6	PB0	PB2	PB5	PB7	PC3	PC2	9
8	PD5	PD3	BSCANP	VCL	VSS	VCC	PB3	PB6	P51	P50	8
7	PD4	PD2	MD1	RX62N Group RX621 Group PTLG0085JA-A (85-pin TFLGA) (Upper perspective view)				P53	P52	VSS_USB	7
6	PD1	PD0	P45					P13	USB0_DM	USB0_DP	6
5	P47	P46	P44					P14	VCC_USB	P12	5
4	P43	P42	P41					RES#	PLLVCC	P16	PLLVSS
3	VREFL	VREFH	P40	MD0	P34	P32	P27	P26	P24	P20	3
2	AVCC	AVSS	VSS	EMLE	XCOU	EXTAL	P33	P30	P23	P22	2
1	P05	VCC	P03	MDE	XCIN	XTAL	P35	P31	P25	P21	1
	A	B	C	D	E	F	G	H	J	K	

Figure 1.9 Pin Assignment of the 85-Pin TFLGA

Table 1.4 List of Pins and Pin Functions (176-Pin LFBGA) (1 / 6)

Pin No. 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communi- cation (SCI, CAN, RSPI, RIIC)	Others
A1	AVSS							
A2	AVCC							
A3		P42						IRQ10-B/AN2
A4		P45						IRQ13-B/AN5
A5		P46						IRQ14/AN6
A6		P90	D16/A16-B					
A7		PD0	D0			POE7#		
A8		PD2	D2			MTIC11W-B/ POE5#		
A9		PD3	D3			MTIC11V-B/ POE4#		
A10		PD4	D4			MTIC11U-B/ POE3#		
A11		PG0	D24					
A12		PD7	D7			MTIC5U-B/ POE0#		
A13		P61	CS1#-A/ SDCS#					
A14		P63	CS3#-A/ CAS#					
A15		PE1	D9				SSLB2-B	
B1		P02				TMCI1-A	SCK6-A	IRQ10-A
B2	VREFH							
B3	VREFL							
B4		P44						IRQ12/AN4
B5		P47						IRQ15-B/AN7
B6		P91	D17/A17-B					
B7		PD1	D1			POE6#		
B8		P96	D22/A22-B					
B9		P97	D23/A23-B					
B10		PD6	D6			MTIC5V-B/ POE1#		
B11		P60	CS0#-A					
B12		P62	CS2#-A/ RAS#					
B13		P64	CS4#-A/ WE#					
B14		PE2	D10			POE9#	SSLB3-B	
B15	SDCLK	P70						
C1		P00				TMRI0-A	TxD6-A	IRQ8-A
C2		P03						IRQ11-A/DA0
C3		P05						IRQ13-A/DA1
C4		P07						IRQ15-A/ ADTRG0#-A
C5		P40						IRQ8-B/AN0

Table 1.4 List of Pins and Pin Functions (176-Pin LFBGA) (2 / 6)

Pin No. 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communi- cation (SCI, CAN, RSPI, RIIC)	Others
C6	VCC							
C7		P92	D18/A18-B					
C8		P94	D20/A20-B					
C9	VCC							
C10		PD5	D5			MTIC5W-B/ POE2#		
C11	VCC							
C12		PE0	D8				SSLB1-B	
C13		PE3	D11			POE8#		
C14		PE5	D13				RSPCKB-B	IRQ5-A
C15		PE6	D14				MOSIB-B	IRQ6-A
D1	EMLE							
D2		P01				TMCIO-A	RxD6-A	IRQ9-A
D3	VCC							
D4		P41						IRQ9-B/AN1
D5		P43						IRQ11-B/AN3
D6	VSS							
D7		P93	D19/A19-B					
D8		P95	D21/A21-B					
D9	VSS							
D10		PG1	D25					
D11	VSS							
D12	VSS							
D13		PE4	D12				SSLB0-B	
D14		PE7	D15				MISOB-B	IRQ7-A
D15		P65	CS5#-A/ CKE					
E1	XCIN							
E2	CNVSS							
E3	BSCANP							
E4	VSS							
E12		PG2	D26					TRDATA0
E13	VCC							
E14		P66	CS6#-A/ DQM0					
E15		P67	CS7#-A/ DQM1					
F1	XCOUT							
F2						WDTOVF#		
F3	VCL							
F4	VSS							
F12		PG4	D28					TRSYNC
F13		PG3	D27					TRDATA1
F14		PA0	A0/BC0#/ DQM2			MTIOC6A/ PO16	SSLA1-B	

Table 1.4 List of Pins and Pin Functions (176-Pin LFBGA) (3 / 6)

Pin No. 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communi- cation (SCI, CAN, RSPI, RIIC)	Others
F15		PG5	D29					TRCLK
G1	VSS							
G2	MD1							
G3	MD0							
G4	MDE							
G12	VSS							
G13	VCC							
G14		PG6	D30					TRDATA2
G15		PA1	A1/DQM3			MTIOC6B/ PO17	SSLA2-B	
H1	XTAL							
H2		P35						NMI
H3	VCC							
H4	RES#							
H12		PG7	D31					TRDATA3
H13		PA2	A2			MTIOC6C/ PO18	SSLA3-B	
H14		PA4	A4			MTIOC7A/ PO20	SSLA0-B	
H15		PA3	A3			MTIOC6D/ PO19		
J1	EXTAL							
J2		P32				MTIOC0C/ PO10/ RTCOUT	CTX0/ TxD6-B	IRQ2-A
J3		PF3						TMS
J4		P34				MTIOC0A/ TMCI3-B/ PO12	SCK6-B	IRQ4-A
J12	VSS							
J13		PA5	A5			MTIOC7B/ PO21	RSPCKA-B	
J14		PA7	A7			MTIOC8B/ PO23	MISOA-B	
J15		PA6	A6			MTIOC8A/ PO22	MOSIA-B	
K1		P33				MTIOC0D/ PO11	CRX0/ RxD6-B	IRQ3-A
K2		P31			USB1_DPRPD	MTIOC4D-A/ TMCI2-B/ PO9	SSLB0-A	IRQ1-A
K3		PF0					TxD1-B	TDO
K4		PF4						TRST#
K12		PB1	A9			MTIOC9C/ PO25		
K13		P71	CS1#-B	ET_MDIO				
K14		P72	CS2#-B	ET_MDC				

Table 1.4 List of Pins and Pin Functions (176-Pin LFBGA) (4 / 6)

Pin No. 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communi- cation (SCI, CAN, RSPI, RIIC)	Others
K15		PB0	A8			MTIOC9A/ PO24		
L1		PF2					RxD1-B	TDI
L2		P27	CS7#-C		USB1_EXICEN	MTIOC2B/ PO7	RSPCKB-A/ SCK1-A	
L3	VCC							
L4		P30			USB1_DRPD	MTIOC4B-A/ TMR13-B/ PO8	MISOB-A/ RxD1-A	IRQ0-A
L12	VSS							
L13		PB4	A12			MTIOC10A/ MTCLKE-B/ PO28		
L14		PB3	A11			MTIOC9D/ MTCLKH-B/ PO27		
L15	VCC							
M1		PF1					SCK1-B	TCK
M2		P25	CS5#-C/ EDACK1-B		USB0_DPRPD	MTIOC4C-A/ MTCLKB-A/ PO5	RxD3-B	ADTRG0#-B
M3		P22	EDREQ0-B		USB0_DRPD	MTIOC3B-A/ MTCLKC-A/ TMO0/ PO2	SCK0	
M4	VSS							
M5		P11			USB1_VBUSEN -A	MTIC5V-A/ TMC13-A	SCK2-A	IRQ1-B
M6		P55	WAIT#-B/ EDREQ0-C	ET_EXOUT		MTIOC4D-B		
M7		P54	EDACK0-C	ET_LINKSTA		MTIOC4B-B		
M8		P51	WR1#/ BC1#/ WAIT#-D				SSLB2-A/ SCK2-B	
M9	VSS							
M10		PC6	A22-A/ CS1#-C	ET_ETXD3		MTIC11V-A/ MTCLKA-B	MOSIA-A	
M11		P81	EDACK0-A	ET_ETXD0/ RMII_TXD0		MTIOC3D-B		
M12		PC0	A16-A	ET_ERXD3		MTCLKG-A	SSLA1-A	
M13	VCC							
M14		PB6	A14			MTIOC10B/ PO30		
M15		PB2	A10			MTIOC9B/ MTCLKG-B/ PO26		
N1		P26	CS6#-C		USB1_ID	MTIOC2A/ TMO1/ PO6	MOSIB-A/ TxD1-A	

Table 1.4 List of Pins and Pin Functions (176-Pin LFBGA) (5 / 6)

Pin No.	Power Supply		External	ETHERC		Timers	Communi-	
176-Pin LFBGA	Clock System Control	I/O Port	Bus EXDMAC	EDMAC	USB	(MTU, TMR, PPG, POE, WDT)	(SCI, CAN, RSPI, RIIC)	Others
N2		P23	EDACK0-B		USB0_DPUPE- A	MTIOC3D-A/ MTCLKD-A/ PO3	TxD3-B	
N3		P20			USB0_ID	MTIOC1A/ TMRI0-B/ PO0	SDA1/ TxD0	
N4		P17			USB1_VBUS/ USB1_OVRCU RB/ USB1_VBUSEN -B	MTIOC3A/ PO15	TxD3-A	IRQ7-B
N5		P15			USB1_OVRCU RA/ USB1_DPUPE- B	MTIOC0B/ TMCI2-A/ PO13	SCK3-A	IRQ5-B
N6		P57	WAIT#-A/ WR3#/ BC3#/ EDREQ1-C					
N7		P10			USB1_DPUPE- A	MTIC5W-A/ TMRI3-A		IRQ0-B
N8		P52	RD#				SSLB3-A/ RxD2-B	
N9	VCC							
N10		PC5	A21-A/ CS2#-C/ WAIT#-C	ET_ETXD2		MTIC11W-A/ MTCLKD-B	RSPCKA-A	
N11		PC3	A19-A	ET_TX_ER		MTCLKF-A	TxD5	
N12		PC2	A18-A	ET_RX_DV		MTCLKE-A	SSLA3-A/ RxD5	
N13		P74	CS4#-B	ET_ERXD1/ RMII_RXD1				
N14		P73	CS3#-B	ET_WOL				
N15		PB5	A13			MTIOC10C/ MTCLKF-B/ PO29		
P1		P24	CS4#-C/ EDREQ1-B		USB0_VBUSEN -A	MTIOC4A-A/ MTCLKA-A/ TMRI1/ PO4	SCK3-B	
P2	PLLVCC							
P3		P16			USB0_VBUS/ USB0_OVRCU RB/ USB0_VBUSEN -B	MTIOC3C-A/ TMO2/ PO14	RxD3-A	IRQ6-B
P4		P14			USB0_OVRCU RA/ USB0_DPUPE- B	TMRI2		IRQ4-B
P5		P13				TMO3	SDA0/ TxD2-A	IRQ3-B/ ADTRG1#
P6	VCC_USB							

Table 1.4 List of Pins and Pin Functions (176-Pin LFBGA) (6 / 6)

Pin No. 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communi- cation (SCI, CAN, RSPI, RIIC)	Others
P7		P56	WR2#/ BC2#/ EDACK1-C			MTIOC3C-B		
P8	VCC_USB							
P9		P84						
P10		P50	WR0#/ WR#				SSLB1-A/ TxD2-B	
P11		P82	EDREQ1-A	ET_ETXD1/ RMII_TXD1		MTIOC4A-B		
P12		PC4	A20-A/ CS3#-C	ET_TX_CLK		MTCLKC-B	SSLA0-A	
P13		P76	CS6#-B	ET_RX_CLK/ REF50CK				
P14		PC1	A17-A	ET_ERXD2		MTCLKH-A	SSLA2-A/ SCK5	
P15		PB7	A15			MTIOC10D/ PO31		
R1		P21			USB0_EXICEN	MTIOC1B/ TMCI0-B/ PO1	SCL1/ RxD0	
R2	PLLVSS							
R3		P12				MTIC5U-A/ TMCI1-B	SCL0/ RxD2-A	IRQ2-B
R4					USB0_DM			
R5					USB0_DP			
R6	VSS_USB							
R7					USB1_DM			
R8					USB1_DP			
R9		P85						
R10	BCLK	P53						
R11		P83	EDACK1-A	ET_CRS/ RMII_CRS_D V		MTIOC4C-B		
R12		PC7	A23-A/ CS0#-B	ET_COL		MTIC11U-A/ MTCLKB-B	MISOA-A	
R13		P80	EDREQ0-A	ET_TX_EN/ RMII_TXD_E N		MTIOC3B-B		
R14		P77	CS7#-B	ET_RX_ER/ RMII_RX_ER				
R15		P75	CS5#-B	ET_ERXD0/ RMII_RXD0				

Table 1.5 List of Pins and Pin Functions (145-Pin TFLGA) (1 / 5)

Pin No. 145-Pin TFLGA	Power Supply Clock System Control	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communi- cation (SCI, CAN, RSPI, RIIC)	Others
A1	AVSS							
A2	AVCC							
A3	VREFL							
A4		P42						IRQ10-B/AN2
A5		P44						IRQ12/AN4
A6		P47						IRQ15-B/AN7
A7		P91	A17-B					
A8		PD0	D0			POE7#		
A9		PD3	D3			MTIC11V-B/ POE4#		
A10		PD6	D6			MTIC5V/ POE1#		
A11		P60	CS0#-A					
A12		P62	CS2#-A/ RAS#					
A13		P64	CS4#-A/ WE#					
B1		P03						IRQ11-A/DA0
B2		P07						IRQ15-A/ ADTRG0#-A
B3	VREFH							
B4		P40						IRQ8-B/AN0
B5		P45						IRQ13-B/AN5
B6		P90	A16-B					
B7		PD1	D1			POE6#		
B8		PD5	D5			MTIC5W/ POE2#		
B9	VSS							
B10		PE0	D8				SSLB1-B	
B11		PE2	D10			POE9#	SSLB3-B	
B12		PE1	D9				SSLB2-B	
B13		PE4	D12				SSLB0-B	
C1		P01				TMCIO-A	RxD6-A	IRQ9-A
C2		P05						IRQ13-A/DA1
C3	VSS							
C4		P41						IRQ9-B/AN1
C5		P46						IRQ14/AN6
C6		P92	A18-B					
C7		PD2	D2			MTIC11W-B/ POE5#		
C8		PD7	D7			MTIC5U/ POE0#		
C9		P61	CS1#-A/ SDCS#					

Table 1.5 List of Pins and Pin Functions (145-Pin TFLGA) (2 / 5)

Pin No.	Power Supply					Timers	Communi-	
145-Pin TFLGA	Clock System Control	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	(MTU, TMR, PPG, POE, WDT)	cation (SCI, CAN, RSPI, RIIC)	Others
C10		P63	CS3#-A/ CAS#					
C11		PE5	D13				RSPCKB-B	IRQ5-A
C12		PE3	D11			POE8#		
C13	SDCLK	P70						
D1	EMLE							
D2	VCC							
D3		P02				TMCI1-A	SCK6-A	IRQ10-A
D4		P43						IRQ11-B/AN3
D5	VCC							
D6	VSS							
D7		P93	A19-B					
D8		PD4	D4			MTIC11U-B/ POE3#		
D9	VCC							
D10	VSS							
D11	VCC							
D12		PE7	D15				MISOB-B	IRQ7-A
D13		PE6	D14				MOSIB-B	IRQ6-A
E1	VCL							
E2	VSS							
E3		P00				TMRI0-A	TxD6-A	IRQ8-A
E4	BSCANP							
E5	(N.C)							
E10		P65	CS5#-A/ CKE					
E11		P67	CS7#-A/ DQM1					
E12		PA0	A0/BC0#			MTIOC6A/ PO16	SSLA1-B	
E13		P66	CS6#-A/ DQM0					
F1	XCIN							
F2	XCOUT							
F3						WDTOVF#		
F4	MDE							
F10		PA1	A1			MTIOC6B/ PO17	SSLA2-B	
F11		PA3	A3			MTIOC6D/ PO19		
F12	VCC							
F13		PA2	A2			MTIOC6C/ PO18	SSLA3-B	
G1	XTAL							
G2	VSS							

Table 1.5 List of Pins and Pin Functions (145-Pin TFLGA) (3 / 5)

Pin No.	Power Supply Clock	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communi- cation (SCI, CAN, RSPI, RIIC)	Others
G3	MD1							
G4	MD0							
G10	VSS							
G11		PA5	A5			MTIOC7B/ PO21	RSPCKA-B	
G12		PA6	A6			MTIOC8A/ PO22	MOSIA-B	
G13		PA4	A4			MTIOC7A/ PO20	SSLA0-B	
H1	EXTAL							
H2		P34				MTIOC0A/ TMCI3/ PO12	SCK6-B	IRQ4-A/ TRST#
H3	VCC							
H4	RES#							
H10		PB0	A8			MTIOC9A/ PO24		
H11		P71	CS1#-B	ET_MDIO				
H12		PB1	A9			MTIOC9C/ PO25		
H13		PA7	A7			MTIOC8B/ PO23	MISOA-B	
J1		P33				MTIOC0D/ PO11	CRX0/ RxD6-B	IRQ3-A
J2		P27	CS7#-C			MTIOC2B/ PO7	RSPCKB-A/ SCK1	TCK
J3		P35						NMI
J4		P32				MTIOC0C/ PO10/ RTCOUT	CTX0/ TxD6-B	IRQ2-A
J10		PB2	A10			MTIOC9B/ MTCLKG-B/ PO26		
J11		PB4	A12			MTIOC10A/ MTCLKE-B/ PO28		
J12		PB5	A13			MTIOC10C/ MTCLKF-B/ PO29		
J13		P72	CS2#-B	ET_MDC				
K1		P30				MTIOC4B-A/ TMRI3/ PO8	RxD1/ MISOB-A	IRQ0/ TDI
K2		P24	CS4#-C/ EDREQ1-B		USB0_VBUSE N-A	MTIOC4A-A/ MTCLKA-A/ TMRI1/PO4	SCK3-B	
K3		P31				MTIOC4D-A/ TMCI2-B/ PO9	SSLB0-A	IRQ1/ TMS

Table 1.5 List of Pins and Pin Functions (145-Pin TFLGA) (4 / 5)

Pin No.	Power Supply Clock	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communi- cation (SCI, CAN, RSPI, RIIC)	Others
K4		P26	CS6#-C			MTIOC2A/ TMO1/ PO6	MOSIB-A/ TxD1	TDO
K5	BCLK	P53						
K6	VSS							
K7		PC7	A23/ CS0#-B	ET_COL		MTIC11U-A/ MTCLKB-B	MISOA-A	
K8		P82	EDREQ1-A	ET_ETXD1/ RMII_TXD1		MTIOC4A-B		TRSYNC
K9		PC3	A19-A	ET_TX_ER		MTCLKF-A	TxD5	
K10		PB7	A15			MTIOC10D/ PO31		
K11		P73	CS3#-B	ET_WOL				
K12		PC0	A16-A	ET_ERXD3		MTCLKG-A	SSLA1-A	
K13		PB3	A11			MTIOC9D/ MTCLKH-B/ PO27		
L1		P25	CS5#-C/ EDACK1-B		USB0_DPRPD	MTIOC4C-A/ MTCLKB-A/ PO5	RxD3-B	ADTRG0#-B
L2		P22	EDREQ0-B		USB0_DRPD	MTIOC3B-A/ MTCLKC-A/ TMO0/PO2	SCK0	
L3		P17				MTIOC3A/ PO15	TxD3-A	IRQ7-B
L4		P12				TMCI1-B	SCL0/ RxD2-A	IRQ2-B
L5	VCC_USB							
L6		P56	EDACK1-C			MTIOC3C-B		
L7		P52	RD#				SSLB3-A/ RxD2-B	
L8		P83	EDACK1-A	ET_CRG/ RMII_CRG_D V		MTIOC4C-B		TRCLK
L9		P81	EDACK0-A	ET_ETXD0/ RMII_TXD0		MTIOC3D-B		TRDATA1
L10		P77	CS7#-B	ET_RX_ER/ RMII_RX_ER				
L11		P75	CS5#-B	ET_ERXD0/ RMII_RXD0				
L12	VCC							
L13		PB6	A14			MTIOC10B/ PO30		
M1		P23	EDACK0-B		USB0_DPUPE -A	MTIOC3D-A/ MTCLKD-A/ PO3	TxD3-B	
M2		P20			USB0_ID	MTIOC1A/ TMRI0-B/ PO0	SDA1/ TxD0	
M3	PLLVC							

Table 1.5 List of Pins and Pin Functions (145-Pin TFLGA) (5 / 5)

Pin No. 145-Pin TFLGA	Power Supply Clock System Control	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communi- cation (SCI, CAN, RSPI, RIIC)	Others
M4		P15				MTIOC0B/ TMCI2-A/ PO13	SCK3-A	IRQ5-B
M5		P14			USB0_OVRC URA/ USB0_DPUPE -B	TMRI2		IRQ4-B
M6	VSS_USB							
M7		P55	WAIT#-B/ EDREQ0-C	ET_EXOUT		MTIOC4D-B		TRDATA3
M8		P50	WR0#/ WR#				SSLB1-A/ TxD2-B	
M9		PC6	A22/CS1#-C	ET_ETXD3		MTIC11V-A/ MTCLKA-B	MOSIA-A	
M10		P80	EDREQ0-A	ET_TX_EN/ RMII_TXD_E N		MTIOC3B-B		TRDATA0
M11		PC2	A18-A	ET_RX_DV		MTCLKE-A	SSLA3-A/ RxD5	
M12		PC1	A17-A	ET_ERXD2		MTCLKH-A	SSLA2-A/ SCK5	
M13	VSS							
N1		P21			USB0_EXICE N	MTIOC1B/ TMCI0-B/ PO1	SCL1/RxD0	
N2		P16			USB0_VBUS/ USB0_OVRC URB/ USB0_VBUSE N-B	MTIOC3C-A/ TMO2/ PO14	RxD3-A	IRQ6-B
N3	PLLVSS							
N4		P13				TMO3	SDA0/ TxD2-A	IRQ3-B/ ADTRG1#
N5					USB0_DM			
N6					USB0_DP			
N7		P54	EDACK0-C	ET_LINKSTA		MTIOC4B-B		TRDATA2
N8		P51	WR1#/BC1#/ WAIT#-D				SSLB2-A/ SCK2	
N9	VCC							
N10		PC5	A21/CS2#-C/ WAIT#-C	ET_ETXD2		MTIC11W-A/ MTCLKD-B	RSPCKA-A	
N11		PC4	A20/CS3#-C	ET_TX_CLK		MTCLKC-B	SSLA0-A	
N12		P76	CS6#-B	ET_RX_CLK/ REF50CK				
N13		P74	CS4#-B	ET_ERXD1/ RMII_RXD1				

Table 1.6 List of Pins and Pin Functions (144-Pin LQFP) (1 / 5)

Pin No.	Power Supply Clock System Control	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communication (SCI, CAN, RSPI, RIIC)	Others
1	AVSS							
2		P05						IRQ13-A/DA1
3	VCC							
4		P03						IRQ11-A/DA0
5	VSS							
6		P02				TMCI1-A	SCK6-A	IRQ10-A
7		P01				TMCI0-A	RxD6-A	IRQ9-A
8		P00				TMR10-A	TxD6-A	IRQ8-A
9	BSCANP							
10	EMLE							
11						WDTOVF#		
12	VSS							
13	MDE							
14	VCL							
15	MD1							
16	MD0							
17	XCIN							
18	XCOUT							
19	RES#							
20	XTAL							
21	VSS							
22	EXTAL							
23	VCC							
24		P35						NMI
25		P34				MTIOC0A/ TMCI3/ PO12	SCK6-B	IRQ4-A/ TRST#
26		P33				MTIOC0D/ PO11	CRX0/ RxD6-B	IRQ3-A
27		P32				MTIOC0C/ PO10/ RTCOUT	CTX0/ TxD6-B	IRQ2-A
28		P31				MTIOC4D- A/ TMCI2-B/ PO9	SSLB0-A	IRQ1/ TMS
29		P30				MTIOC4B-A/ TMR13/ PO8	RxD1/ MISOB-A	IRQ0/ TDI
30		P27	CS7#-C			MTIOC2B/ PO7	RSPCKB-A/ SCK1	TCK
31		P26	CS6#-C			MTIOC2A/ TMO1/ PO6	MOSIB-A/ TxD1	TDO

Table 1.6 List of Pins and Pin Functions (144-Pin LQFP) (2 / 5)

Pin No.	Power Supply Clock	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communication (SCI, CAN, RSPI, RIIC)	Others
32		P25	CS5#-C/ EDACK1-B		USB0_DPRPD	MTIOC4C-A/ MTCLKB-A/ PO5	RxD3-B	ADTRG0#-B
33		P24	CS4#-C/ EDREQ1-B		USB0_VBUSEN -A	MTIOC4A-A/ MTCLKA-A/ TMR11/PO4	SCK3-B	
34		P23	EDACK0-B		USB0_DPUPE- A	MTIOC3D- A/ MTCLKD-A/ PO3	TxD3-B	
35		P22	EDREQ0-B		USB0_DRPD	MTIOC3B-A/ MTCLKC-A/ TMO0/PO2	SCK0	
36		P21			USB0_EXICEN	MTIOC1B/ TMC10-B/ PO1	SCL1/RxD0	
37		P20			USB0_ID	MTIOC1A/ TMR10-B/ PO0	SDA1/ TxD0	
38		P17				MTIOC3A/ PO15	TxD3-A	IRQ7-B
39	PLLVC							
40		P16			USB0_VBUS/ USB0_OVRCU RB/ USB0_VBUSEN -B	MTIOC3C- A/ TMO2/ PO14	RxD3-A	IRQ6-B
41	PLLVS							
42		P15				MTIOC0B/ TMC12-A/ PO13	SCK3-A	IRQ5-B
43		P14			USB0_OVRCU RA/ USB0_DPUPE- B	TMR12		IRQ4-B
44		P13				TMO3	SDA0/ TxD2-A	IRQ3-B/ ADTRG1#
45		P12				TMC11-B	SCL0/ RxD2-A	IRQ2-B
46	VCC_USB							
47					USB0_DM			
48					USB0_DP			
49	VSS_USB							
50		P56	EDACK1-C			MTIOC3C-B		
51		P55	WAIT#-B/ EDREQ0-C	ET_EXOUT		MTIOC4D-B		TRDATA3
52		P54	EDACK0-C	ET_LINKSTA		MTIOC4B-B		TRDATA2
53	BCLK	P53						
54		P52	RD#				SSLB3-A/ RxD2-B	

Table 1.6 List of Pins and Pin Functions (144-Pin LQFP) (3 / 5)

Pin No.	Power Supply Clock System Control	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communication (SCI, CAN, RSPI, RIIC)	Others
55		P51	WR1#/BC1#/ WAIT#-D				SSLB2-A/ SCK2	
56		P50	WR0#/ WR#				SSLB1-A/ TxD2-B	
57	VSS							
58		P83	EDACK1-A	ET_CRS/ RMII_CRS_DV			MTIOC4C-B	TRCLK
59	VCC							
60		PC7	A23/ CS0#-B	ET_COL			MTIC11U-A/ MTCLKB-B	MISOA-A
61		PC6	A22/ CS1#-C	ET_ETXD3			MTIC11V-A/ MTCLKA-B	MOSIA-A
62		PC5	A21/CS2#-C/ WAIT#-C	ET_ETXD2			MTIC11W-A/ MTCLKD-B	RSPCKA-A
63		P82	EDREQ1-A	ET_ETXD1/ RMII_TXD1			MTIOC4A-B	TRSYNC
64		P81	EDACK0-A	ET_ETXD0/ RMII_TXD0			MTIOC3D-B	TRDATA1
65		P80	EDREQ0-A	ET_TX_EN/ RMII_TXD_EN			MTIOC3B-B	TRDATA0
66		PC4	A20/CS3#-C	ET_TX_CLK			MTCLKC-B	SSLA0-A
67		PC3	A19-A	ET_TX_ER			MTCLKF-A	TxD5
68		P77	CS7#-B	ET_RX_ER/ RMII_RX_ER				
69		P76	CS6#-B	ET_RX_CLK/ REF50CK				
70		PC2	A18-A	ET_RX_DV			MTCLKE-A	SSLA3-A/ RxD5
71		P75	CS5#-B	ET_ERXD0/ RMII_RXD0				
72		P74	CS4#-B	ET_ERXD1/ RMII_RXD1				
73		PC1	A17-A	ET_ERXD2			MTCLKH-A	SSLA2-A/ SCK5
74	VCC							
75		PC0	A16-A	ET_ERXD3			MTCLKG-A	SSLA1-A
76	VSS							
77		P73	CS3#-B	ET_WOL				
78		PB7	A15				MTIOC10D/ PO31	
79		PB6	A14				MTIOC10B/ PO30	
80		PB5	A13				MTIOC10C/ MTCLKF-B/ PO29	
81		PB4	A12				MTIOC10A/ MTCLKE-B/ PO28	

Table 1.6 List of Pins and Pin Functions (144-Pin LQFP) (4 / 5)

Pin No.	Power Supply Clock	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communication (SCI, CAN, RSPI, RIIC)	Others
82		PB3	A11			MTIOC9D/ MTCLKH-B/ PO27		
83		PB2	A10			MTIOC9B/ MTCLKG-B/ PO26		
84		PB1	A9			MTIOC9C/ PO25		
85		P72	CS2#-B	ET_MDC				
86		P71	CS1#-B	ET_MDIO				
87		PB0	A8			MTIOC9A/ PO24		
88		PA7	A7			MTIOC8B/ PO23	MISOA-B	
89		PA6	A6			MTIOC8A/ PO22	MOSIA-B	
90		PA5	A5			MTIOC7B/ PO21	RSPCKA-B	
91	VCC							
92		PA4	A4			MTIOC7A/ PO20	SSLA0-B	
93	VSS							
94		PA3	A3			MTIOC6D/ PO19		
95		PA2	A2			MTIOC6C/ PO18	SSLA3-B	
96		PA1	A1			MTIOC6B/ PO17	SSLA2-B	
97		PA0	A0/BC0#/ DQM1			MTIOC6A/ PO16	SSLA1-B	
98		P67	CS7#-A/ DQM1					
99		P66	CS6#-A/ DQM0					
100		P65	CS5#-A/ CKE					
101		PE7	D15				MISOB-B	IRQ7-A
102		PE6	D14				MOSIB-B	IRQ6-A
103	VCC							
104	SDCLK	P70						
105	VSS							
106		PE5	D13				RSPCKB-B	IRQ5-A
107		PE4	D12				SSLB0-B	
108		PE3	D11			POE8#		
109		PE2	D10			POE9#	SSLB3-B	
110		PE1	D9				SSLB2-B	
111		PE0	D8				SSLB1-B	

Table 1.6 List of Pins and Pin Functions (144-Pin LQFP) (5 / 5)

Pin No.	Power Supply Clock	I/O Port	External Bus EXDMAC	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE, WDT)	Communication (SCI, CAN, RSPI, RIIC)	Others
112		P64	CS4#-A/ WE#					
113		P63	CS3#-A/ CAS#					
114		P62	CS2#-A/ RAS#					
115		P61	CS1#-A/ SDCS#					
116	VSS							
117		P60	CS0#-A					
118	VCC							
119		PD7	D7			MTIC5U/ POE0#		
120		PD6	D6			MTIC5V/ POE1#		
121		PD5	D5			MTIC5W/ POE2#		
122		PD4	D4			MTIC11U-B/ POE3#		
123		PD3	D3			MTIC11V-B/ POE4#		
124		PD2	D2			MTIC11W-B/ POE5#		
125		PD1	D1			POE6#		
126		PD0	D0			POE7#		
127		P93	A19-B					
128		P92	A18-B					
129		P91	A17-B					
130	VSS							
131		P90	A16-B					
132	VCC							
133		P47						IRQ15-B/AN7
134		P46						IRQ14/AN6
135		P45						IRQ13-B/AN5
136		P44						IRQ12/AN4
137		P43						IRQ11-B/AN3
138		P42						IRQ10-B/AN2
139		P41						IRQ9-B/AN1
140	VREFL							
141		P40						IRQ8-B/AN0
142	VREFH							
143	AVCC							
144		P07						IRQ15-A/ ADTRG0#-A

Table 1.7 List of Pins and Pin Functions (100-Pin LQFP) (1 / 4)

Pin No.	Power Supply	I/O	External Bus	ETHERC	USB	Timers	Communi- cation	Others
100-Pin LQFP	Clock System Control	Port		EDMAC		(MTU, TMR, PPG, POE)	(SCI, CAN, RSPI, RIIC)	
1	VCC							
2	EMLE							
3	VSS							
4	MDE							
5	VCL							
6	MD1							
7	MD0							
8	XCIN							
9	XCOUT							
10	RES#							
11	XTAL							
12	VSS							
13	EXTAL							
14	VCC							
15		P35						NMI
16		P34				MTIOC0A/ TMCI3/ PO12	SCK6	IRQ4-A/ TRST#
17		P33				MTIOC0D/ PO11	CRX0/ RxD6	IRQ3-A
18		P32				MTIOC0C/ PO10/ RTCOUT	CTX0/ TxD6	IRQ2-A
19		P31				MTIOC4D- A/ TMCI2/ PO9	SSLB0-A	IRQ1/ TMS
20		P30				MTIOC4B- A/ TMRI3/ PO8	RxD1/ MISOB-A	IRQ0/ TDI
21		P27	CS7#			MTIOC2B/ PO7	RSPCKB- A/ SCK1	TCK
22		P26	CS6#			MTIOC2A/ TMO1/ PO6	MOSIB-A/ TxD1	TDO
23		P25	CS5#		USB0_DPRPD	MTIOC4C/ MTCLKB-A/ PO5	RxD3	ADTRG0#-B
24		P24	CS4#		USB0_VBUSE N-A	MTIOC4A/ MTCLKA-A/ TMRI1/PO4	SCK3	
25		P23			USB0_DPUPE- A	MTIOC3D/ MTCLKD-A/ PO3	TxD3	
26		P22			USB0_DRPD	MTIOC3B/ MTCLKC-A/ TMO0/PO2	SCK0	

Table 1.7 List of Pins and Pin Functions (100-Pin LQFP) (2 / 4)

Pin No.	Power Supply Clock System Control	I/O Port	External Bus	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE)	Communi- cation (SCI, CAN, RSPI, RIIC)	Others
27		P21			USB0_EXICEN	MTIOC1B/ TMC10/ PO1	RxD0	
28		P20			USB0_ID	MTIOC1A/ TMR10/ PO0	TxD0	
29	PLLVC							
30		P16			USB0_VBUS/ USB0_OVRCU RB/ USB0_VBUSE N-B	MTIOC3C/ TMO2/ PO14		IRQ6-B
31	PLLVS							
32		P14			USB0_OVRCU RA/ USB0_DPUPE- B	MTIOC3A/ TMR12/ PO15		IRQ4-B
33		P13				MTIOC0B/ TMO3/ PO13	SDA0/ TxD2-A	IRQ3-B/ ADTRG1#
34		P12				TMC11	SCL0/ RxD2-A	IRQ2-B
35	VCC_USB							
36					USB0_DM			
37					USB0_DP			
38	VSS_USB							
39		P55	WAIT#-B			MTIOC4D-B		
40		P54				MTIOC4B-B		
41	BCLK	P53						
42		P52	RD#				SSLB3-A/ RxD2-B	
43		P51	WR1#/BC1#/ WAIT#-D				SSLB2-A/ SCK2	
44		P50	WR0#/ WR#				SSLB1-A/ TxD2-B	
45		PC7	A23/ CS0#	ET_COL		MTIC11U-A/ MTCLKB-B	MISOA-A	
46		PC6	A22/ CS1#	ET_ETXD3		MTIC11V-A/ MTCLKA-B	MOSIA-A	
47		PC5	A21/CS2#/ WAIT#-C	ET_ETXD2		MTIC11W- A/ MTCLKD-B	RSPCKA-A	
48		PC4	A20/CS3#	ET_TX_CLK		MTCLKC-B	SSLA0-A	
49		PC3	A19	ET_TX_ER		MTCLKF-A	TxD5	
50		PC2	A18	ET_RX_DV		MTCLKE-A	SSLA3-A/ RxD5	
51		PC1	A17	ET_ERXD2		MTCLKH-A	SSLA2-A/ SCK5	
52		PC0	A16	ET_ERXD3		MTCLKG-A	SSLA1-A	

Table 1.7 List of Pins and Pin Functions (100-Pin LQFP) (3 / 4)

Pin No.	Power Supply Clock System Control	I/O Port	External Bus	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE)	Communi- cation (SCI, CAN, RSPI, RIIC)	Others
53		PB7	A15	ET_CRS/ RMII_CRS_D V		MTIOC10D/ PO31		
54		PB6	A14	ET_ETXD1/ RMII_TXD1		MTIOC10B/ PO30		
55		PB5	A13	ET_ETXD0/ RMII_TXD0		MTIOC10C/ MTCLKF-B/ PO29		
56		PB4	A12	ET_TX_EN/ RMII_TXD_E N		MTIOC10A/ MTCLKE-B/ PO28		
57		PB3	A11	ET_RX_ER/ RMII_RX_ER		MTIOC9D/ MTCLKH-B/ PO27		
58		PB2	A10	ET_RX_CLK/ REF50CK		MTIOC9B/ MTCLKG-B/ PO26		
59		PB1	A9	ET_ERXD0/ RMII_RXD0		MTIOC9C/ PO25		
60	VCC							
61		PB0	A8	ET_ERXD1/ RMII_RXD1		MTIOC9A/ PO24		
62	VSS							
63		PA7	A7	ET_WOL		MTIOC8B/ PO23	MISOA-B	
64		PA6	A6	ET_EXOUT		MTIOC8A/ PO22	MOSIA-B	
65		PA5	A5	ET_LINKSTA		MTIOC7B/ PO21	RSPCKA-B	
66		PA4	A4	ET_MDC		MTIOC7A/ PO20	SSLA0-B	
67		PA3	A3	ET_MDIO		MTIOC6D/ PO19		
68		PA2	A2			MTIOC6C/ PO18	SSLA3-B	
69		PA1	A1			MTIOC6B/ PO17	SSLA2-B	
70		PA0	A0/BC0#			MTIOC6A/ PO16	SSLA1-B	
71		PE7	D15				MISOB-B	IRQ7
72		PE6	D14				MOSIB-B	IRQ6-A
73		PE5	D13				RSPCKB-B	IRQ5
74		PE4	D12				SSLB0-B	
75		PE3	D11			POE8#		
76		PE2	D10			POE9#	SSLB3-B	
77		PE1	D9				SSLB2-B	
78		PE0	D8				SSLB1-B	
79		PD7	D7			MTIC5U/ POE0#		

Table 1.7 List of Pins and Pin Functions (100-Pin LQFP) (4 / 4)

Pin No.	Power Supply Clock System Control	I/O Port	External Bus	ETHERC EDMAC	USB	Timers (MTU, TMR, PPG, POE)	Communi- cation (SCI, CAN, RSPI, RIIC)	Others
80		PD6	D6			MTIC5V/ POE1#		
81		PD5	D5			MTIC5W/ POE2#		
82		PD4	D4			MTIC11U-B/ POE3#		
83		PD3	D3			MTIC11V-B/ POE4#		
84		PD2	D2			MTIC11W- B/ POE5#		
85		PD1	D1			POE6#		
86		PD0	D0			POE7#		
87		P47						IRQ15-B/AN7
88		P46						IRQ14/AN6
89		P45						IRQ13-B/AN5
90		P44						IRQ12/AN4
91		P43						IRQ11/AN3
92		P42						IRQ10/AN2
93		P41						IRQ9/AN1
94	VREFL							
95		P40						IRQ8/AN0
96	VREFH							
97	AVCC							
98		P07						IRQ15-A/ ADTRG0#-A
99	AVSS							
100		P05						DA1/IRQ13-A

Table 1.8 List of Pins and Pin Functions (85-Pin TFLGA) (1 / 3)

Pin No.	Power Supply Clock	I/O Port	External Bus	USB	Timers (MTU, TMR, PPG)	Communication (SCI, CAN, RSPI, RIIC)	Others
A1		P05					DA1/ IRQ13-A
A2	AVCC						
A3	VREFL						
A4		P43					IRQ11-B/ AN3
A5		P47					IRQ15/ AN7
A6		PD1	D1				
A7		PD4	D4		MTIC11U		
A8		PD5	D5		MTIC5W		
A9		PD7	D7		MTIC5U		
A10		PD6	D6		MTIC5V		
B1	VCC						
B2	AVSS						
B3	VREFH						
B4		P42					IRQ10/ AN2
B5		P46					IRQ14/ AN6
B6		PD0	D0				
B7		PD2	D2		MTIC11W		
B8		PD3	D3		MTIC11V		
B9		PA3	A3		MTIOC6D/PO19		
B10		PA1	A1		MTIOC6B/PO17	SSLA2	
C1		P03					IRQ11-A/ DA0
C2	VSS						
C3		P40					IRQ8/ AN0
C4		P41					IRQ9/ AN1
C5		P44					IRQ12/ AN4
C6		P45					IRQ13-B/ AN5
C7	MD1						
C8	BSCANP						
C9		PA5	A5		MTIOC7B/PO21	RSPCKA	
C10		PA0	A0		MTIOC6A/PO16	SSLA1	
D1	MDE						
D2	EMLE						
D3	MD0						
D4	RES#						

Table 1.8 List of Pins and Pin Functions (85-Pin TFLGA) (2 / 3)

Pin No.	Power Supply Clock	I/O Port	External Bus	USB	Timers (MTU, TMR, PPG)	Communication (SCI, CAN, RSPI, RIIC)	Others
D8	VCL						
D9		PA6	A6		MTIOC8A/PO22	MOSIA	
D10		PA2	A2		MTIOC6C/PO18	SSLA3	
E1	XCIN						
E2	XCOU						
E3		P34			MTIOC0A/TMC13/PO12	SCK6	IRQ4-A/ TRST#
E8	VSS						
E9		PB0	A8		MTIOC9A/PO24		
E10		PA4	A4		MTIOC7A/PO20	SSLA0	
F1	XTAL						
F2	EXTAL						
F3		P32			MTIOC0C/PO10/RTCOU	TxD6/CTX0	IRQ2-A
F8	VCC						
F9		PB2	A10		MTIOC9B/MTCLKG-B/ PO26		
F10		PA7	A7		MTIOC8B/PO23	MISOA	
G1		P35					NMI
G2		P33			MTIOC0D/PO11	RxD6/CRX0	IRQ3-A
G3		P27	CS7#		MTIOC2B/PO7	SCK1/RSPCKB	TCK
G8		PB3	A11		MTIOC9D/MTCLKH-B/PO27		
G9		PB5	A13		MTIOC10C/MTCLKF-B/PO29		
G10		PB1	A9		MTIOC9C/PO25		
H1		P31			MTIOC4D/TMC12/PO9	SSLB0	IRQ1/ TMS
H2		P30			MTIOC4B/TMR13/PO8	RxD1/MISOB	IRQ0/ TDI
H3		P26	CS6#		MTIOC2A/TMO1/PO6	TxD1/MOSIB	TDO
H4	PLLVC						
H5		P14		USB0_OVRCUR A/USB0_DPUPE- B	MTIOC3A/TMR12/PO15		IRQ4-B
H6		P13			MTIOC0B/TMO3/PO13	TxD2-A/SDA0	IRQ3-B/ ADTRG 1#
H7	BCLK	P53					
H8		PB6	A14		MTIOC10B/PO30		
H9		PB7	A15		MTIOC10D/PO31		
H10		PB4	A12		MTIOC10A/MTCLKE-B/PO28		
J1		P25	CS5#	USB0_DPRPD	MTIOC4C/MTCLKB/PO5	RxD3	ADTRG 0#
J2		P23		USB0_DPUPE-A	MTIOC3D/MTCLKD/PO3	TxD3	
J3		P24	CS4#	USB0_VBUSEN- A	MTIOC4A/MTCLKA/TMR11/ PO4	SCK3	

Table 1.8 List of Pins and Pin Functions (85-Pin TFLGA) (3 / 3)

Pin No.	Power Supply Clock	I/O Port	External Bus	USB	Timers (MTU, TMR, PPG)	Communication (SCI, CAN, RSPI, RIIC)	Others
J4		P16		USB0_VBUS/ USB0_OVRCUR B/ USB0_VBUSEN- B	MTIOC3C/TMO2/PO14		IRQ6
J5	VCC_USB						
J6				USB0_DM			
J7		P52	RD#			RxD2-B/SSLB3	
J8		P51	WAIT#			SCK2/SSLB2	
J9		PC3	A19		MTCLKF-A	TxD5	
J10		PC0	A16		MTCLKG-A		
K1		P21		USB0_EXICEN	MTIOC1B/TMCI0/PO1	RxD0/SCL1	
K2		P22		USB0_DRPD	MTIOC3B/MTCLKC/TMO0/PO2	SCK0	
K3		P20		USB0_ID	MTIOC1A/TMRI0/PO0	TxD0/SDA1	
K4	PLLVS						
K5		P12			TMCI1	RxD2-A/SCL0	IRQ2-B
K6				USB0_DP			
K7	VSS_USB						
K8		P50	WR0#			TxD2-B/SSLB1	
K9		PC2	A18		MTCLKE-A	RxD5	
K10		PC1	A17		MTCLKH-A	SCK5	

1.5 Pin Functions

Table 1.8 lists the pin functions.

Table 1.9 Pin Functions (1 / 7)

Classifications	Pin Name	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply.
	VCL	Input	Connect this pin to VSS via a 0.1- μ F capacitor. The capacitor should be placed close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
	PLLVCC	Input	Power supply pin for the PLL circuit. Connect it to the system power supply.
	PLLVSS	Input	Ground pin for the PLL circuit.
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	BCLK	Output	Outputs the external bus clock for external devices.
	SDCLK	Output	Outputs the clock dedicated for the SDRAM.
	XCOUT	Output	Input/output pins for the subclock generation circuit. Connect a crystal resonator between XCOUT and XCIN.
	XCIN	Input	
Operating mode control	MD0, MD1, MDE	Input	Pins for setting the operating mode. The signal levels on these pins must not be changed during operation.
System control	RES#	Input	Reset signal input pin. This LSI enters the reset state when this signal goes low.
	EMLE	Input	Input pin to enable the connection of the on-chip emulator signal. When the on-chip emulator is used, this pin should be driven high. When not used, it should be driven low.
	BSCANP	Input	Boundary scan pin. Boundary scan is enabled when this pin goes high. When not used, it should be driven low.
CNVSS	CNVSS	Input	Connect this pin to VSS via pull-down resistor.
On-chip emulator	TRST#	Input	On-chip emulator pins or boundary scan pins. When the EMLE pin is driven high, these pins are dedicated for the on-chip emulator.
	TMS	Input	
	TDI	Input	
	TCK	Input	
	TDO	Output	
	TRCLK	Output	This pin outputs the clock for synchronization with the trace data.
	TRSYNC	Output	This pin indicates that output from the TRDATA0 to TRDATA3 pins is valid.
TRDATA0-A/TRDATA3-B	Output	These pins output the trace information.	
Address bus	A0 to A15 A16-A/A16-B to A23-A/A23-B	Output	Output pins for the address.
Data bus	D0 to D31	I/O	Input and output pins for the bidirectional data bus.

Table 1.9 Pin Functions (2 / 7)

Classifications	Pin Name	I/O	Description	
Bus control	RD#	Output	Strobe signal which indicates that reading from the external bus interface space is in progress.	
	WR#	Output	Strobe signal which indicates that writing to the external bus interface space is in progress, in 1-write strobe mode.	
	WR0# to WR3#	Output	Strobe signals which indicate that any group of data bus pins (D7 to D0, D15 to D8, D23 to D16, and D31 to D24) is valid in writing to the external bus interface space, in byte strobe mode.	
	BC0# to BC3#	Output	Strobe signals which indicate that any group of data bus pins (D7 to D0, D15 to D8, D23 to D16, and D31 to D24) is valid in access to the external bus interface space, in 1-write strobe mode.	
	WE#	Output	Output pin for SDRAM write enable signals.	
	CAS#	Output	Output pin for SDRAM column address strobe signals.	
	RAS#	Output	Output pin for SDRAM row address strobe signals.	
	CKE	Output	Output pin for SDRAM clock enable signals.	
	DQM0 to DQM34	Output	Output pins for SDRAM I/O data mask enable signals.	
	SDCS#	Output	Output pin for SDRAM chip select signals.	
	CS0#-A/CS0#-B CS1#-A/CS1#-B/CS1#-C CS2#-A/CS2#-B/CS2#-C CS3#-A/CS3#-B/CS3#-C CS4#-A/CS4#-B/CS4#-C CS5#-A/CS5#-B/CS5#-C CS6#-A/CS6#-B/CS6#-C CS7#-A/CS7#-B/CS7#-C	Output	Select signals for areas 0 to 7.	
	WAIT#-A/WAIT#-B/ WAIT#-C/WAIT#-D	Input	Input pins for wait request signals in access to the external space.	
	EXDMA controller	EDREQ0-A/EDREQ0-B/ EDREQ0-C	Input	Input pins for external DMA transfer requests of channel 0.
		EDREQ1-A/EDREQ1-B/ EDREQ1-C	Input	Input pins for external DMA transfer requests of channel 1.
EDACK0-A/EDACK0-B/ EDACK0-C		Output	Output pins for single address transfer acknowledge signals of channel 0.	
EDACK1-A/EDACK1-B/ EDACK1-C		Output	Output pins for single address transfer acknowledge signals of channel 1.	
Interrupt	NMI	Input	Non-maskable interrupt request signal.	
	IRQ0-A/IRQ0-B IRQ1-A/IRQ1-B IRQ2-A/IRQ2-B IRQ3-A/IRQ3-B IRQ4-A/IRQ4-B IRQ5-A/IRQ5-B IRQ6-A/IRQ6-B IRQ7-A/IRQ7-B IRQ8-A/IRQ8-B IRQ9-A/IRQ9-B IRQ10-A/IRQ10-B IRQ11-A/IRQ11-B IRQ12 IRQ13-A/IRQ13-B IRQ14 IRQ15-A/IRQ15-B	Input	Interrupt request signals.	

Table 1.9 Pin Functions (3 / 7)

Classifications	Pin Name	I/O	Description
Multi-function timer pulse unit	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-A/MTIOC3B-B MTIOC3C-A/MTIOC3C-B MTIOC3D-A/MTIOC3D-B	I/O	The TGRA3 to TGRD3 input capture input/output compare output/PWM output pins.
	MTIOC4A-A/MTIOC4A-B MTIOC4B-A/MTIOC4B-B MTIOC4C-A/MTIOC4C-B MTIOC4D-A/MTIOC4D-B	I/O	The TGRA4 and TGRB4 input capture input/output compare output/PWM output pins.
	MTIC5U-A/MTIC5U-B MTIC5V-A/MTIC5V-B MTIC5W-A/MTIC5W-B	Input	The TGRU5, TGRV5, and TGRW5 input capture input/dead time compensation input pins.
	MTIOC6A MTIOC6B MTIOC6C MTIOC6D	I/O	The TGRA6 to TGRD6 input capture input/output compare output/PWM output pins.
	MTIOC7A MTIOC7B	I/O	The TGRA7 and TGRB7 input capture input/output compare output/PWM output pins.
	MTIOC8A MTIOC8B	I/O	The TGRA8 and TGRB8 input capture input/output compare output/PWM output pins.
	MTIOC9A MTIOC9B MTIOC9C MTIOC9D	I/O	The TGRA9 to TGRD9 input capture input/output compare output/PWM output pins.
	MTIOC10A MTIOC10B MTIOC10C MTIOC10D	I/O	The TGRA10 to TGRB10 input capture input/output compare output/PWM output pins.
	MTIC11U-A/MTIC11U-B MTIC11V-A/MTIC11V-B MTIC11W-A/MTIC11W-B	Input	The TGRU11, TGRV11, and TGRW11 input capture input/dead time compensation input pins.
	MTCLKA-A/MTCLKA-B MTCLKB-A/MTCLKB-B MTCLKC-A/MTCLKC-B MTCLKD-A/MTCLKD-B MTCLKE-A/MTCLKE-B MTCLKF-A/MTCLKF-B MTCLKG-A/MTCLKG-B MTCLKH-A/MTCLKH-B	Input	Input pins for external clock signals.
	Port output enable	POE0# to POE9#	Input
Programmable pulse generator	PO0 to PO31	Output	Output pins for the pulse signals.

Table 1.9 Pin Functions (4 / 7)

Classifications	Pin Name	I/O	Description
8-bit timer	TMO0 to TMO3	Output	Output pins for the compare match signals.
	TMC10-A/TMC10-B TMC11-A/TMC11-B TMC12-A/TMC12-B TMC13-A/TMC13-B	Input	Input pins for the external clock signals that drive for the counters.
	TMR10-A/TMR10-B TMR11 TMR12 TMR13-A/TMR13-B	Input	Input pins for the counter-reset signals.
Watchdog timer	WDTOVF#	Output	Output pin for the counter-overflow signal in watchdog-timer mode.
Serial communications interface	TxD0 TxD1-A/TxD1-B TxD2-A/TxD2-B TxD3-A/TxD3-B TxD5 TxD6-A/TxD6-B	Output	Output pins for data transmission.
	RxD0 RxD1-A/RxD1-B RxD2-A/RxD2-B RxD3-A/RxD3-B RxD5 RxD6-A/RxD6-B	Input	Input pins for data reception.
	SCK0 SCK1-A/SCK1-B SCK2-A/SCK2-B SCK3-A/SCK3-B SCK5 SCK6-A/SCK6-B	I/O	Input/output pins for clock signals.
I ² C bus interface	SCL0, SCL1	I/O	Input/output pins for I ² C bus interface clocks. Bus can be directly driven by the NMOS open drain output.
	SDA0, SDA1	I/O	Input/output pins for I ² C bus interface data. Bus can be directly driven by the NMOS open drain output.

Table 1.9 Pin Functions (5 / 7)

Classifications	Pin Name	I/O	Description
Ethernet controller	REF50CK	Input	50-MHz reference clock. This pin inputs reference signals for transmission/reception timings in RMII mode.
	RMII_CRS_DV	Input	Indicates that there are carrier detection signals and valid receive data on RMII_RXD1 and RMII_RXD0 in RMII mode.
	RMII_TXD0, RMII_TXD1	Output	2-bit transmit data in RMII mode.
	RMII_RXD0, RMII_RXD1	Input	2-bit receive data in RMII mode.
	RMII_TXD_EN	Output	Output pin for data transmit enable signals in RMII mode.
	RMII_RX_ER	Input	Indicates an error has occurred during reception of data in RMII mode.
	ET_CRS	Input	Carrier detection/data reception enable pin.
	ET_RX_DV	Input	Indicates that there are valid receive data on ET_ERXD3 to ET_ERXD0.
	ET_EXOUT	Output	General-purpose external output pin.
	ET_LINKSTA	Input	Inputs link status from the PHY-LSI.
	ET_ETXD0 to ET_ETXD3	Output	4 bits of MII transmit data.
	ET_ERXD0 to ET_ERXD3	Input	4 bits of MII receive data.
	ET_TX_EN	Output	Transmit enable pin. Indicates that transmit data is ready on ET_ETXD3 to ET_ETXD0.
	ET_TX_ER	Output	Transmit error pin. Notifies the PHY_LSI of an error during transmission.
	ET_RX_ER	Input	Receive error pin. Recognizes an error during reception.
	ET_TX_CLK	Input	Transmit clock pin. This pin inputs reference signals for output timings from ET_TX_EN, ET_ETXD3 to ET_ETXD0, and ET_TX_ER.
	ET_RX_CLK	Input	Receive clock pin. This pin inputs reference signals for input timings to ET_RX_DV, ET_ERXD3 to ET_ERXD0, and ET_RX_ER.
	ET_COL	Input	Inputs collision detection signals.
	ET_WOL	Output	Receives Magic Packets™
	ET_MDC	Output	Outputs reference clock signals for information transfer via ET_MDIO.
ET_MDIO	I/O	These pins carry bidirectional signals for the exchange of management information between the RX62N Group and the PHY-LSI.	

Table 1.9 Pin Functions (6 / 7)

Classifications	Pin Name	I/O	Description
USB 2.0 host/function module	VCC_USB	Input	Power-supply pin for the USB. Connect this pin to the system power supply even when the USB is not to be used.
	VSS_USB	Input	Ground pin for the USB. Connect this pin to the system power supply (0 V) even when the USB is not to be used.
	USB0_DP USB1_DP	I/O	Inputs or outputs D+ data for the USB bus.
	USB0_DM USB1_DM	I/O	Inputs or outputs D- data for the USB bus.
	USB0_DPRPD USB1_DPRPD	Output	Enable D+ pull-down.
	USB0_DRPD USB1_DRPD	Output	Enable D- pull-down.
	USB0_EXICEN USB1_EXICEN	Output	Connect these pins to the OTG power supply IC.
	USB0_ID USB1_ID	Input	Connect these pins to the OTG power supply IC.
	USB0_VBUSEN-A/ USB0_VBUSEN-B USB1_VBUSEN-A/ USB1_VBUSEN-B	Output	VBUS power enable pins for the USB.
	USB0_DPUPE-A/ USB0_DPUPE-B USB1_DPUPE-A/ USB1_DPUPE-B	Output	Pull-up pins for the USB.
	USB0_OVRCURA/ USB0_OVRCURB USB1_OVRCURA/ USB1_OVRCURB	Input	Over current pins for the USB.
	USB0_VBUS USB1_VBUS	Input	Input pins for detection of connection and disconnection of the USB cable.
	CAN module	CRX0	Input
CTX0		Output	Output pins for the CAN.
Serial peripheral interfaces	RSPCKA-A/ RSPCKA-B	I/O	Clock input/output pins for the RSPI.
	RSPCKB-A/ RSPCKB-B	I/O	Clock input/output pins for the RSPI
	MOSIA-A/MOSIA-B MOSIB-A/MOSIB-B	I/O	Input or output data output from the master for the RSPI.
	MISOA-A/MISOA-B MISOB-A/MISOB-B	I/O	Input or output data output from the slave for the RSPI.
	SSLA0-A/SSLA0-B	I/O	Select the slave for the RSPI.
	SSLA1-A/SSLA1-B SSLA2-A/SSLA2-B SSLA3-A/SSLA3-B	Output	
	SSLB0-A/SSLB0-B	I/O	
	SSLB1-A/SSLB1-B SSLB2-A/SSLB2-B SSLB3-A/SSLB3-B	Output	
Realtime clock	RTCOUT	Output	Output pin for 1-Hz clock.
A/D converter	AN0 to AN7	Input	Input pins for the analog signals to be processed by the A/D converter.
	ADTRG0#-A/ADTRG0#-B ADTRG1#	Input	Input pins for the external trigger signals that start the A/D conversion.
D/A converter	DA0, DA1	Output	Output pins for the analog signals from the D/A converter.

Table 1.9 Pin Functions (7 / 7)

Classifications	Pin Name	I/O	Description
Analog power supply	AVCC	Input	Analog power supply pin for the A/D and D/A converters. When the A/D and D/A converters are not in use, connect this pin to the system power supply.
	AVSS	Input	Ground pin for the A/D and D/A converters. Connect this pin to the system power supply (0 V).
	VREFH	Input	Reference power supply pin for the A/D and D/A converters. When the A/D and D/A converters are not in use, connect this pin to the system power supply.
	VREFL	Input	Reference ground pin for the A/D and D/A converters. Make sure to connect this pin to the analog reference power supply (0 V). When the A/D and D/A converters are not in use, connect this pin to the system power supply (0 V). For details, see section 35.6.7, Ranges of Settings for Analog Power Supply and Other Pins.
I/O ports	P00 to P03, P05, P07	I/O	6-bit input/output pins.
	P10 to P17	I/O	8-bit input/output pins.
	P20 to P27	I/O	8-bit input/output pins.
	P30 to P34	I/O	5-bit input/output pins.
	P35	Input	1-bit input pin.
	P40 to P47	I/O	8-bit input/output pins.
	P50 to P52, P54 to P57	I/O	7-bit input/output pins.
	P53	Input	1-bit input pin.
	P60 to P67	I/O	8-bit input/output pins.
	P70 to P77	I/O	8-bit input/output pins.
	P80 to P85	I/O	6-bit input/output pins.
	P90 to P97	I/O	8-bit input/output pins.
	PA0 to PA7	I/O	8-bit input/output pins.
	PB0 to PB7	I/O	8-bit input/output pins.
	PC0 to PC7	I/O	8-bit input/output pins.
	PD0 to PD7	I/O	8-bit input/output pins.
	PE0 to PE7	I/O	8-bit input/output pins.
	PF0 to PF4	I/O	5-bit input/output pins.
	PG0 to PG7	I/O	8-bit input/output pins.

2. CPU

The RX62N/RX621 Group is an MCU with the high-speed, high-performance RX CPU as its core.

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

The CPU has 73 basic instructions and 8 floating-point operation instructions, and 9 DSP instructions, for a total of 90 instructions. It has 10 addressing modes and caters to register–register operations, register–memory operations, immediate–register operations, immediate–memory operations, memory–memory transfer, and bitwise operations. High-speed operation was realized by achieving execution in a single cycle not only for register–register operations, but also for other types of multiple instructions. The CPU includes an internal multiplier and an internal divider for high-speed multiplication and division.

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

2.1 Features

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

2.2 Register Set of the CPU

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

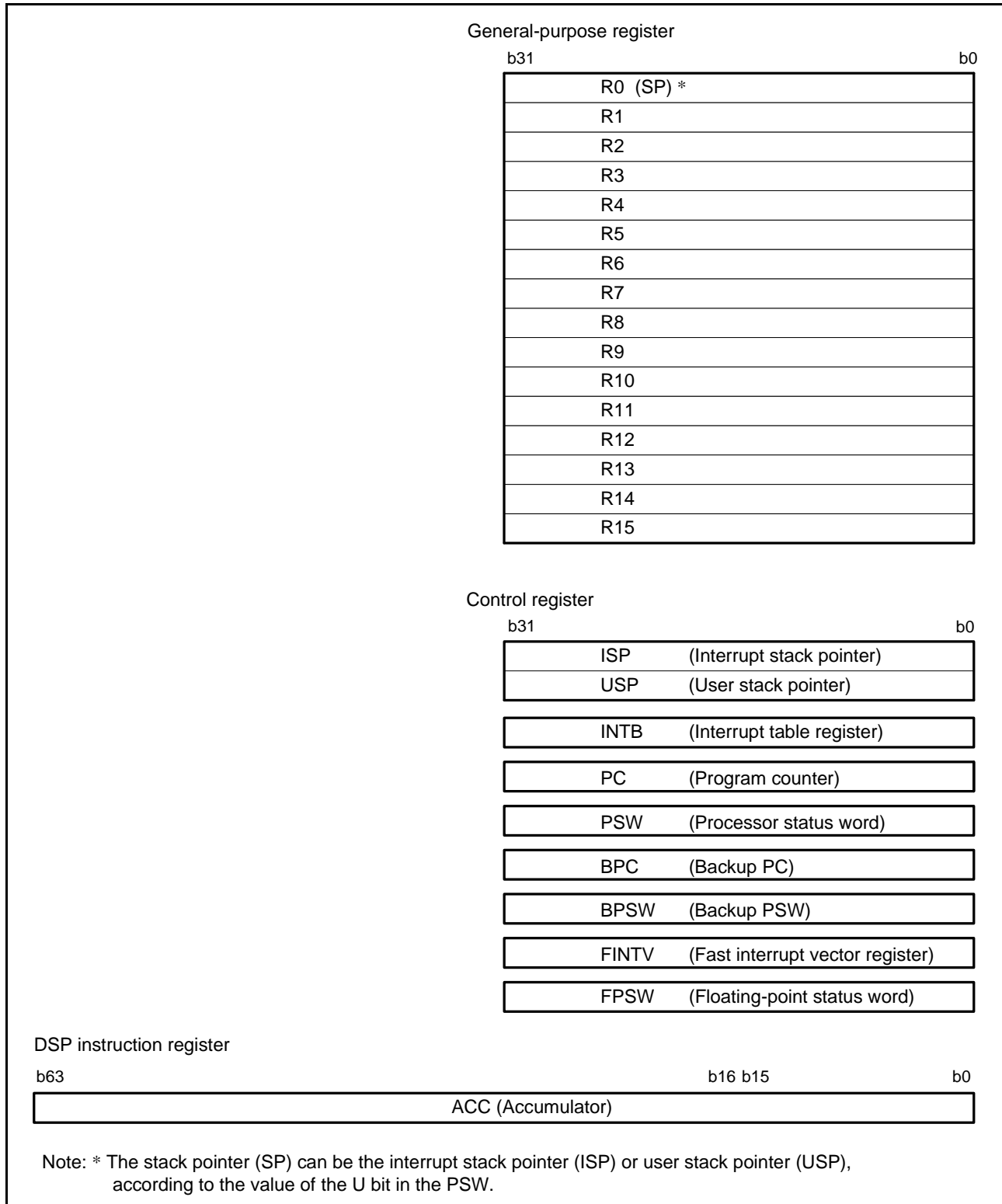


Figure 2.1 Register Set of the CPU

2.2.1 General-Purpose Registers (R0 to R15)

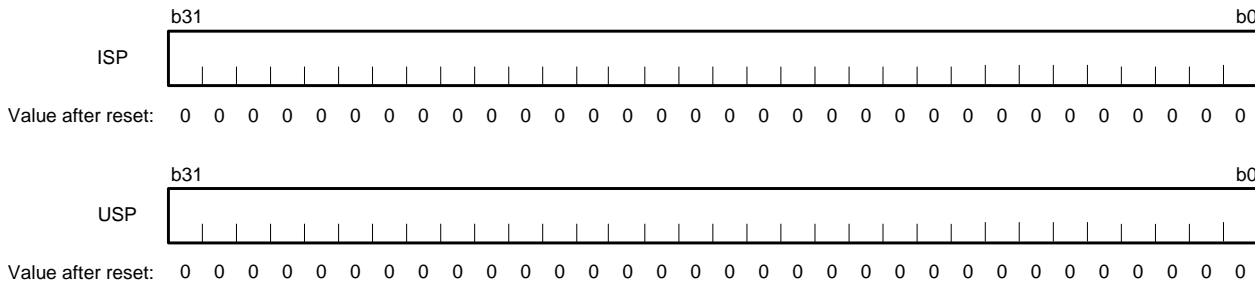
This CPU has sixteen general-purpose registers (R0 to R15). R1 to R15 can be used as data registers or address registers. R0, a general-purpose register, also functions as the stack pointer (SP). The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (USP) by the value of the stack pointer select bit (U) in the processor status word (PSW).

2.2.2 Control Registers

This CPU has the following nine control registers.

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

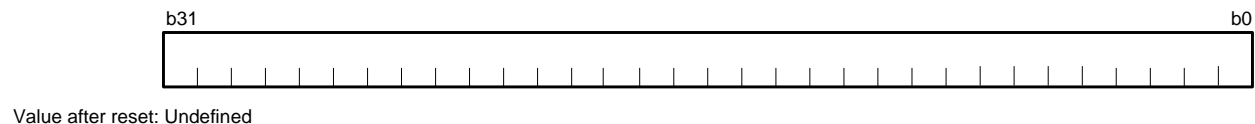
2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP)



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

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

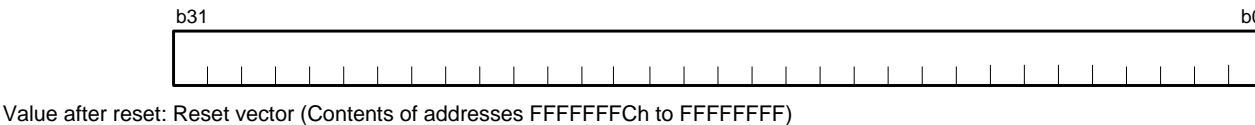
2.2.2.2 Interrupt Table Register (INTB)



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

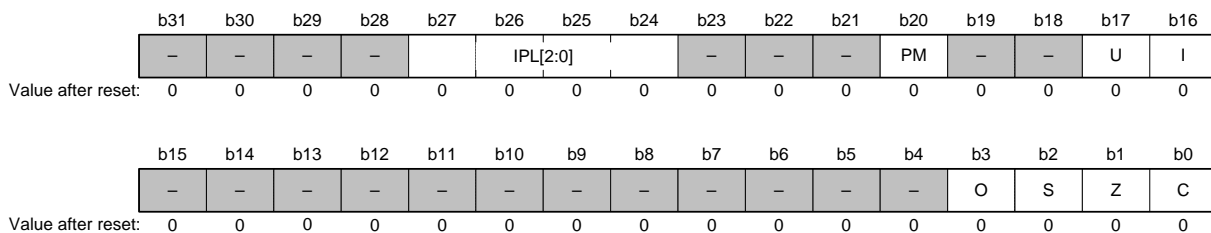
Set INTB to a multiple of four.

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 non-zero. 1: Result is 0.	R/W
b2	S	Sign Flag	0: Result is a positive value or 0. 1: Result is a negative value.	R/W
b3	O	Overflow Flag	0: No overflow has occurred. 1: An overflow has occurred.	R/W
b15 to b4	—	Reserved	These bits are always 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 always 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 always read as 0. The write value should be 0.	R/W
b27 to b24	IPL[3:0]*1	Processor Interrupt Priority Level	b27 b24 0 0 0 0: Priority level 0 (lowest) 0 0 0 1: Priority level 1 0 0 1 0: Priority level 2 0 0 1 1: Priority level 3 0 1 0 0: Priority level 4 0 1 0 1: Priority level 5 0 1 1 0: Priority level 6 0 1 1 1: Priority level 7 1 0 0 0: Priority level 8 1 0 0 1: Priority level 9 1 0 1 0: Priority level 10 1 0 1 1: Priority level 11 1 1 0 0: Priority level 12 1 1 0 1: Priority level 13 1 1 1 0: Priority level 14 1 1 1 1: Priority level 15 (highest)	R/W
b31 to b28	—	Reserved	These bits are always 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 PM bit in PSW saved on the stack to 1 or executing an RTFI instruction after having set the PM bit in BPSW to 1.

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

C Flag (Carry Flag)

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

Z Flag (Zero Flag)

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

S Flag (Sign Flag)

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

O Flag (Overflow Flag)

This flag indicates that an overflow occurred during an operation.

I Bit (Interrupt Enable)

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

U Bit (Stack Pointer Select)

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

PM Bit (Processor Mode Select)

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

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

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

2.2.2.5 Backup PC (BPC)



Value after reset: Undefined

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

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

2.2.2.6 Backup PSW (BPSW)

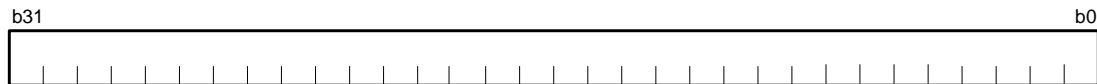


Value after reset: Undefined

The backup PSW (BPSW) is provided to speed up response to interrupts.

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

2.2.2.7 Fast Interrupt Vector Register (FINTV)



Value after reset: Undefined

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

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

2.2.2.8 Floating-Point Status Word (FPSW)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FS	FX	FU	FZ	FO	FV	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	EX	EU	EZ	EO	EV	—	DN	CE	CX	CU	CZ	CO	CV	RM[1:0]	—
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	RM[1:0]	Floating-Point Rounding-Mode Setting	b1 b0 0 0: Rounding to the nearest value 0 1: Rounding to 0 1 0: Rounding to $+\infty$ 1 1: Rounding to $-\infty$	R/W
b2	CV	Invalid Operation Cause Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.	R/(W)*1
b3	CO	Overflow Cause Flag	0: No overflow has occurred. 1: Overflow has occurred.	R/(W)*1
b4	CZ	Division-by-Zero Cause Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.	R/(W)*1
b5	CU	Underflow Cause Flag	0: No underflow has occurred. 1: Underflow has occurred.	R/(W)*1
b6	CX	Inexact Cause Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.	R/(W)*1
b7	CE	Un-Implemented Processing Cause Flag	0: No un-implemented processing has been encountered. 1: Un-implemented process has been encountered.	R/(W)*1
b8	DN	0 Flush Bit of Denormalized Number	0: A denormalized number is handled as a denormalized number. 1: A denormalized number is handled as 0.*2	R/W
b9	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W
b10	EV	Invalid Operation Exception Enable	0: Invalid operation exception is masked. 1: Invalid operation exception is enabled.	R/W
b11	EO	Overflow Exception Enable	0: Overflow exception is masked. 1: Overflow exception is enabled.	R/W
b12	EZ	Division-by-Zero Exception Enable	0: Division-by-zero exception is masked. 1: Division-by-zero exception is enabled.	R/W
b13	EU	Underflow Exception Enable	0: Underflow exception is masked. 1: Underflow exception is enabled.	R/W
b14	EX	Inexact Exception Enable	0: Inexact exception is masked. 1: Inexact exception is enabled.	R/W
b25 to b15	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b26	FV*3	Invalid Operation Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.*8	R/W
b27	FO*4	Overflow Flag	0: No overflow has occurred. 1: Overflow has occurred.*8	R/W
b28	FZ*5	Division-by-Zero Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.*8	R/W
b29	FU*6	Underflow Flag	0: No underflow has occurred. 1: Underflow has occurred.*8	R/W

Bit	Symbol	Bit Name	Description	R/W
b30	FX*7	Inexact Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.*8	R/W
b31	FS	Floating-point error summary flag	This bit reflects the logical OR of the FU, FZ, FO, and FV flags.	R

Note 1. Writing 0 to the bit clears it. Writing 1 to the bit does not affect its value.

Note 2. Positive denormalized numbers are treated as +0, negative denormalized numbers as -0.

Note 3. When the EV bit is set to 0, the FV flag is enabled.

Note 4. When the EO bit is set to 0, the FO flag is enabled.

Note 5. When the EZ bit is set to 0, the FZ flag is enabled.

Note 6. When the EU bit is set to 0, the FU flag is enabled.

Note 7. When the EX bit is set to 0, the FX flag is enabled.

Note 8. Once the bit has been set to 1, this value is retained until it is cleared to 0 by software.

The floating-point status word (FPSW) indicates the results of floating-point operations.

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

RM[1:0] bits (Floating-point rounding-mode setting)

These bits specify the floating-point rounding-mode.

Explanation of Floating-Point Rounding Modes

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

(1) Rounding to the nearest value is specified as the default mode and returns the most accurate value.

(2) Modes such as rounding towards 0, rounding towards $+\infty$, and rounding towards $-\infty$ are used to ensure precision when interval arithmetic is employed.

CV flag (Invalid operation cause flag), CO flag (Overflow cause flag), CZ flag (Division-by-zero cause flag), CU flag (Underflow cause flag), CX flag (Inexact cause flag), and CE flag (Unimplemented processing cause flag)

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

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

DN flag (0 flush bit of denormalized number)

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

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

EV bit (Invalid operation exception enable bit), EO bit (Overflow exception enable bit), EZ bit (Division-by-zero exception enable bit), EU bit (Underflow exception enable bit), and EX bit (Inexact exception enable bit)

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

FV flag (Invalid operation flag), FO flag (Overflow flag), FZ flag (Division-by-zero flag), FU flag (Underflow flag), and FX flag (Inexact flag)

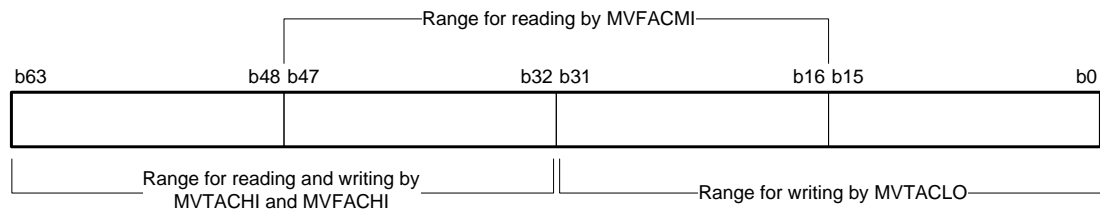
While the exception handling enable bit (Ej) is 0 (exception handling is masked), if any of five floating-point exceptions specified in the IEEE754 standard is generated, the corresponding bit is set to 1.

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

FS flag (Floating-point error summary flag)

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

2.2.3 Accumulator (ACC)



Value after reset: Undefined

The accumulator (ACC) is a 64-bit register used for DSP instructions. The accumulator is also used for the multiply and multiply-and-accumulate instructions; EMUL, EMULU, FMUL, MUL, and RMPA, in which case the prior value in the accumulator is modified by execution of the instruction.

Use the MVTACHI and MVTACLO instructions for writing to the accumulator. The MVTACHI and MVTACLO instructions write data to the higher-order 32 bits (bits 63 to 32) and the lower-order 32 bits (bits 31 to 0), respectively.

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

2.3 Processor Mode

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

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

The initial state after a reset is supervisor mode.

2.3.1 Supervisor Mode

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

2.3.2 User Mode

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

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

2.3.3 Privileged Instruction

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

2.3.4 Switching Between Processor Modes

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

(1) Switching from user mode to supervisor mode

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

2.4 Data Types

The RX CPU can handle four types of data: integer, floating-point, bit, and string.

2.4.1 Integer

An integer can be signed or unsigned. For signed integers, negative values are represented by two's complements.

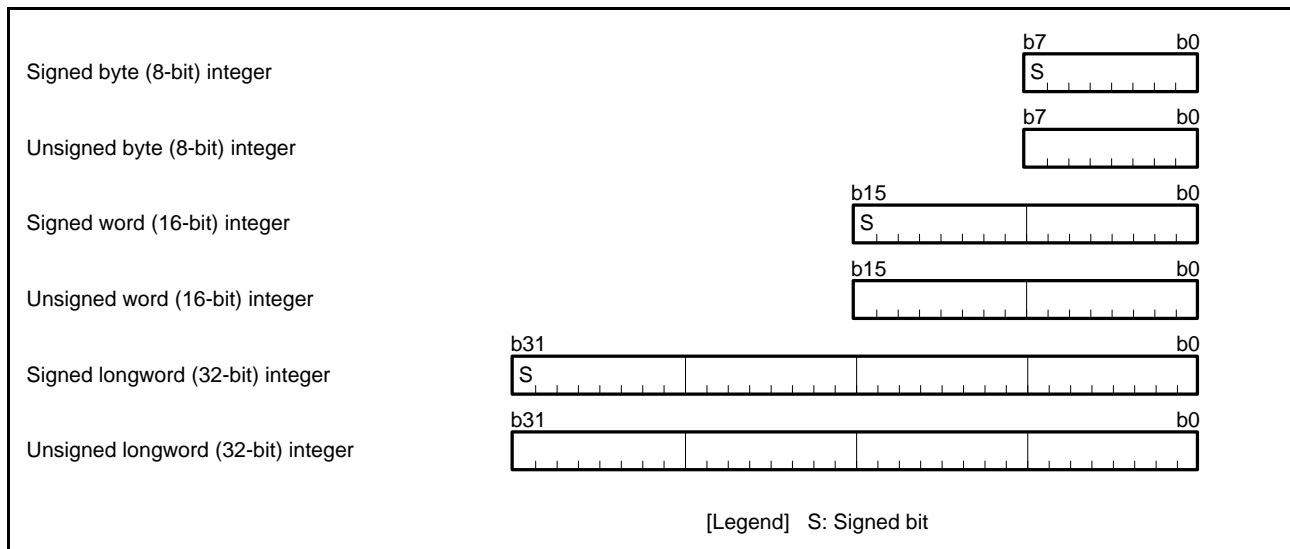


Figure 2.2 Integer

2.4.2 Floating-Point

Floating-point support is for the single-precision floating-point type specified in IEEE754; operands of this type can be used in eight floating-point operation instructions: FADD, FCMP, FDIV, FMUL, FSUB, FTOI, ITOF, and ROUND.

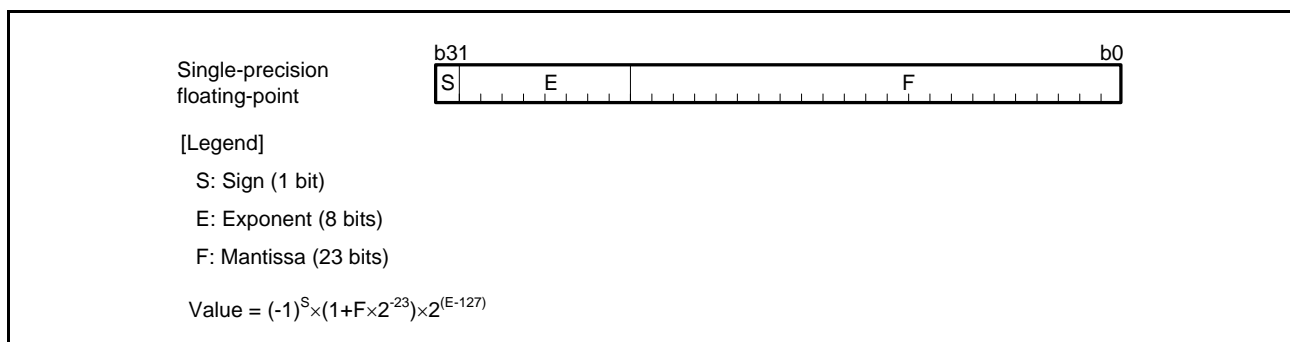


Figure 2.3 Floating-Point

The floating-point format supports the values listed below.

- $0 < E < 255$ (normal numbers)
- $E = 0$ and $F = 0$ (signed zero)
- $E = 0$ and $F > 0$ (denormalized numbers)*

Note: The number is treated as 0 when the DN bit in FPSW is 1. When the DN bit is 0, an un-implemented processing exception is generated.

- $E = 255$ and $F = 0$ (infinity)
- $E = 255$ and $F > 0$ (NaN: Not-a-Number)

2.4.3 Bitwise Operations

Five bit-manipulation instructions are provided for bitwise operations: BCLR, BMCnd, BNOT, BSET, and BTST.

A bit in a register is specified as the destination register and a bit number in the range from 31 to 0.

A bit in memory is specified as the destination address and a bit number from 7 to 0. The addressing modes available to specify addresses are register indirect and register relative.

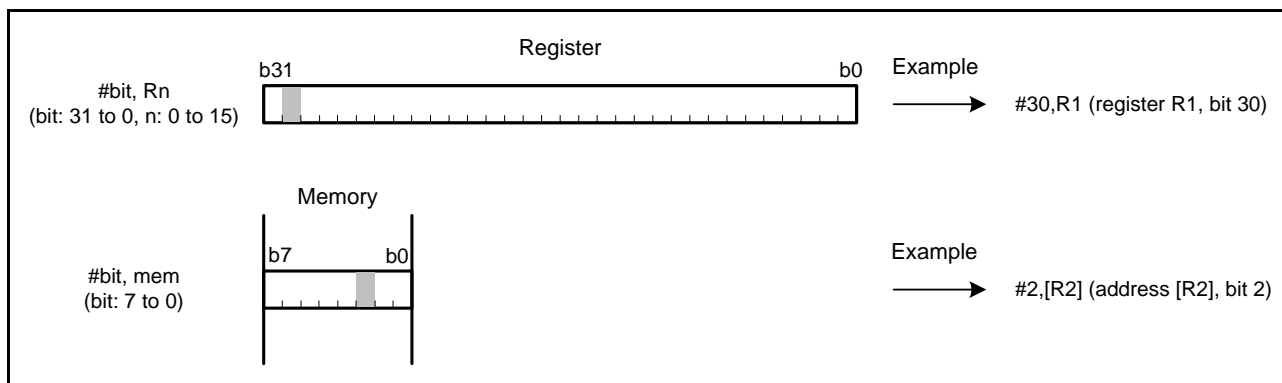


Figure 2.4 Bit

2.4.4 Strings

The string data type consists of an arbitrary number of consecutive byte (8-bit), word (16-bit), or longword (32-bit) units.

Seven string manipulation instructions are provided for use with strings: SCMPU, SMOVB, SMOVE, SMOVU, SSTR, SUNTIL, and SWHILE.

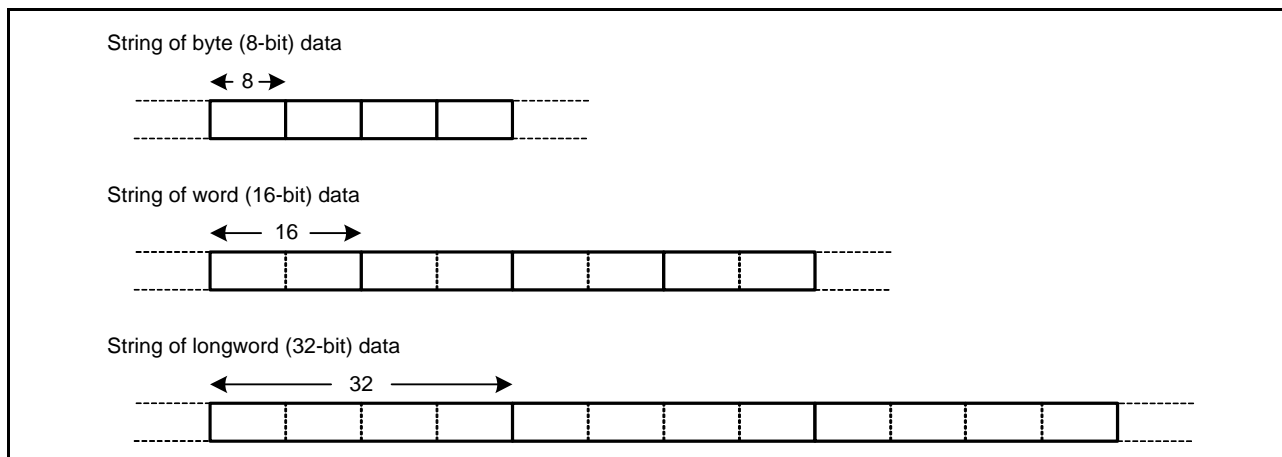


Figure 2.5 String

2.5 Endian

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

2.5.1 Switching the Endian

As arrangements of bytes, the RX62N/RX621 Group supports both big endian, where the higher-order byte (MSB) is at location 0, and little endian, where the lower-order byte (LSB) is at location 0.

For details on the endian setting, see section 3, Operating Modes, and section 12, Buses.

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 register,

LH indicates bits D15 to D8 of the general register,

HL indicates bits D23 to D16 of the general register, and

HH indicates bits D31 to D24 of the general register.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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.

- For I/O registers with a bus width of eight bits, use instructions with size specifier (.size) of .B or size extension specifier (.memex) of .B or .UB.
- For I/O registers with a bus width of 16 bits, use instructions with size specifier (.size) of .W or size extension specifier (.memex) of .W or .UW.
- For I/O registers with a bus width of 32 bits, use instructions with size specifier (.size) of .L or size extension specifier (.memex) of .L.

2.5.4 Data Arrangement

2.5.4.1 Data Arrangement in Registers

Figure 2.6 shows the relation between the sizes of registers and bit numbers.

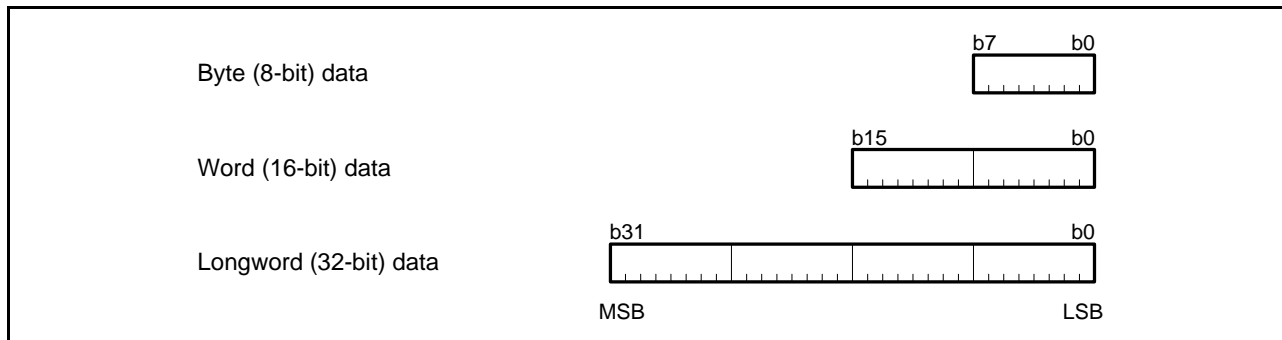


Figure 2.6 Data Arrangement in Registers

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 2.7 shows the arrangement of data in memory.

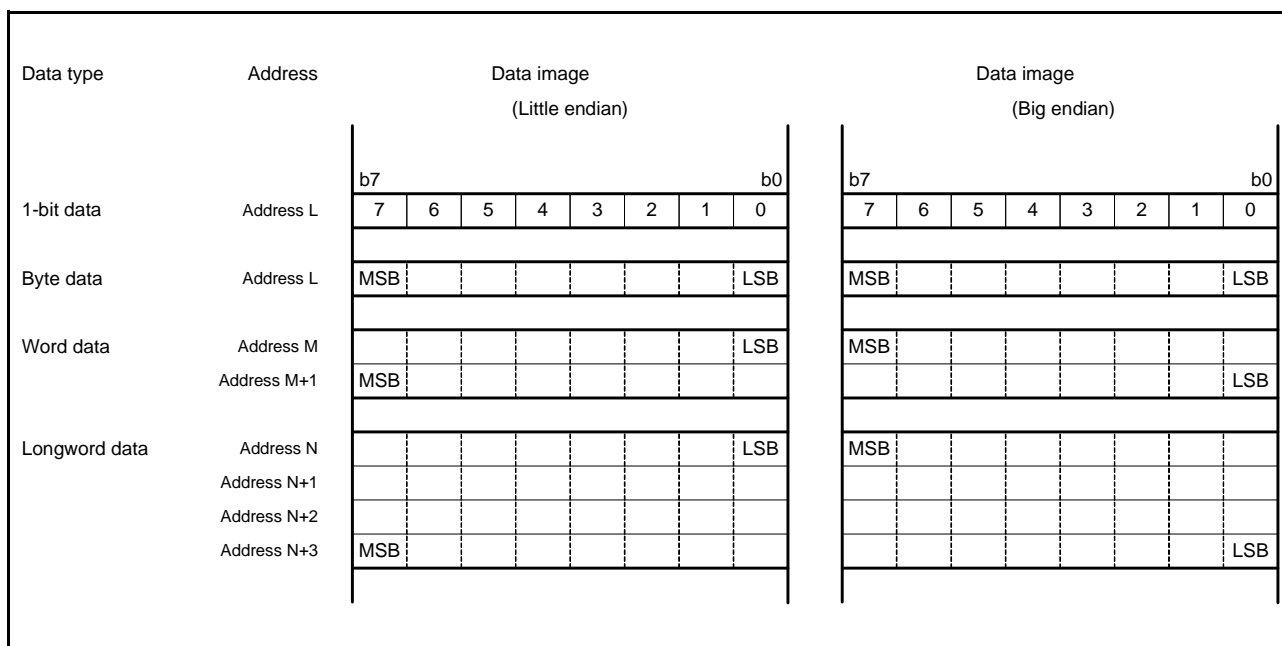


Figure 2.7 Data Arrangement in Memory

2.5.5 Notes on the Allocation of Instruction Codes

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

2.6 Vector Table

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

2.6.1 Fixed Vector Table

The fixed vector table is allocated to a fixed address range. The individual vectors for the privileged instruction exception, access exceptions, undefined instruction exception, floating-point exception, non-maskable interrupt, and reset are allocated to addresses in the range from FFFFFFFF80h to FFFFFFFFCh. Figure 2.8 shows the fixed vector table.

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

Figure 2.8 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 2.9 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 that of the instruction itself (from 0 to 255). The BRK instruction is allocated to the vector with number 0. Furthermore, vector numbers within the set from 0 to 255 may also be allocated to other interrupt sources on a per-product basis. For more on the interrupt vector numbers, see section 11, Interrupt Control Unit (ICUa).

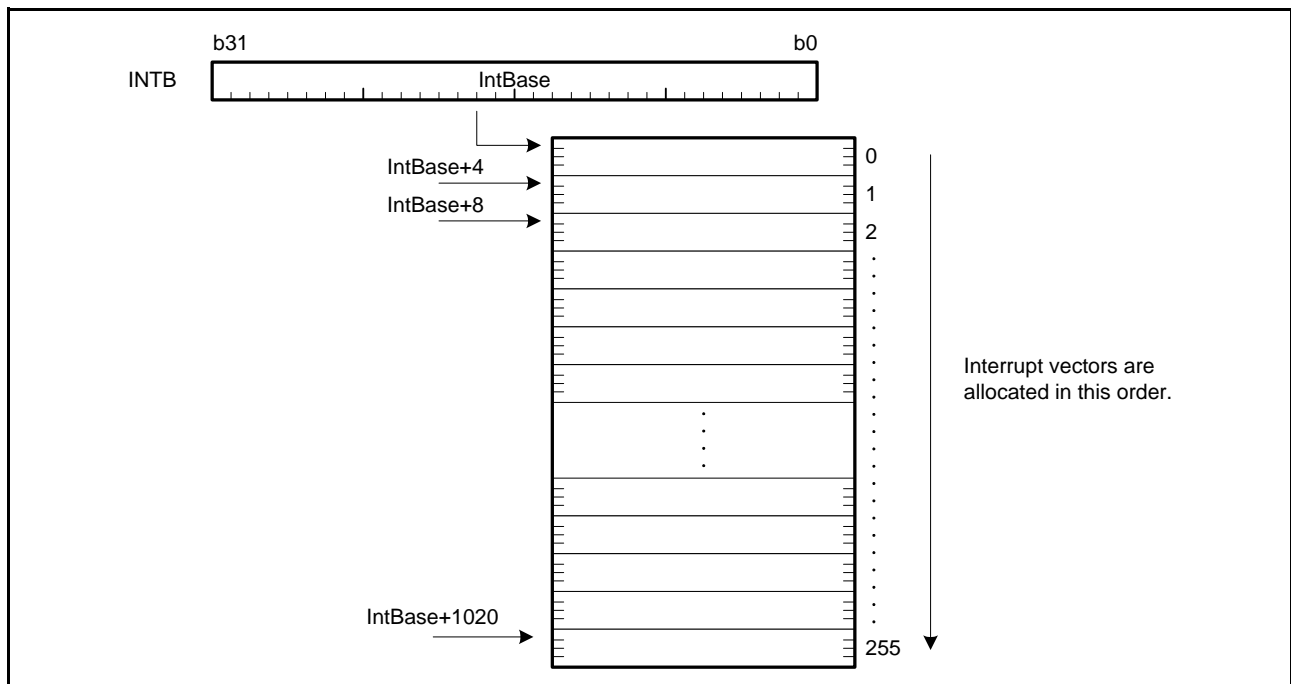


Figure 2.9 Relocatable Vector Table

2.7 Operation of Instructions

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

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

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

2.8 Pipeline

2.8.1 Overview

The RX CPU has 5-stage pipeline structure. The RX CPU instruction is converted into one or more micro-operations, which are then executed in pipeline processing. In the pipeline stage, the IF stage is executed in the unit of instructions, while the D and subsequent stages are executed in the unit of micro-operations.

The operation of pipeline and respective stages is described below.

(1) IF stage (instruction fetch stage)

In the IF stage, the CPU fetches instructions from the memory. As the RX CPU has four 8-byte instruction queues, it fetches instructions until the instruction queue is full, regardless of the completion of decoding in the D (decoding) stage.

(2) D stage (decoding stage)

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

(3) E stage (execution stage)

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

(4) M stage (memory access stage)

Operand memory accesses (OA) 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, OA2) 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 2.10 shows the pipeline configuration and its operation.

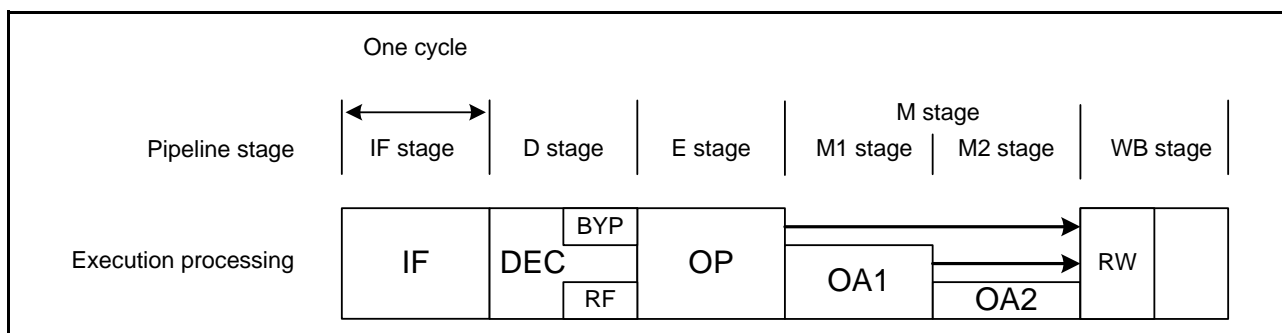


Figure 2.10 Pipeline Configuration and its Operation

2.8.2 Instructions and Pipeline Processing

The operands in the table below indicate the following meaning.

#IMM: Immediate

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

dsp: dsp5, dsp8, dsp16, dsp24

pcdsp: pcdsp3, pcdsp8, pcdsp16, pcdsp24

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 (division)	DIV "#IMM, Rd"/"Rs, Rd"	Figure 2.11	3 to 20*1
	DIVU "#IMM, Rd"/"Rs, Rd"	Figure 2.11	2 to 18*1
Data transfer instructions (register-register, immediate-register)	{MOV, MOVU, REVL, REVW} "#IMM, Rd"/"Rs, Rd" SCCnd "Rd" {STNZ, STZ} "#IMM, Rd"	Figure 2.11	1
Transfer instructions (load operation)	{MOV, MOVU} "[Rs], Rd"/"dsp[Rs], Rd" /"[Rs+], Rd"/"[-Rs], Rd"/"Rs, [Ri, Rb]" POP "Rd"	Figure 2.12	Throughput: 1 Latency: 2*2
Transfer instructions (store operation)	MOV "Rs, [Rd]"/"Rs, dsp[Rd]"/"Rs, [Rd+]" /"Rs, [-Rd]"/"Rs, [Ri, Rb]" PUSH "Rs" PUSHC "CR"	Figure 2.13	1
Bit manipulation instructions (register)	{BCLR, BNOT, BSET, BTST} "#IMM, Rd"/"Rs, Rd" BMCnd "#IMM, Rd"	Figure 2.11	1
Branch instructions	BCnd "pcdsp" {BRA, BSR} "pcdsp"/"Rs" {JMP, JSR} "Rs"	Figure 2.22	Branch taken: 3 Branch not taken: 1
Floating-point operation instructions (register-register, immediate-register)	FCMP "#IMM, Rd"/"Rs, Rd"	Figure 2.11	1
System manipulation instructions	CLRPSW, SETPSW "#IMM" MVTC "#IMM, CR"/"Rs, CR" MVFC "CR, Rd" MVTIPL "#IMM"	—	1
DSP function instructions	{MACHI, MACLO, MULHI, MULLO}"Rs, Rs2" {MVFACHI, MVFACMI}"Rd" {MVTACHI, MVTACLO}"Rs" RACW"#IMM"	Figure 2.11	1

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

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

Figure 2.11 to Figure 2.13 show the operation of instructions that are converted into a basic single micro-operation.

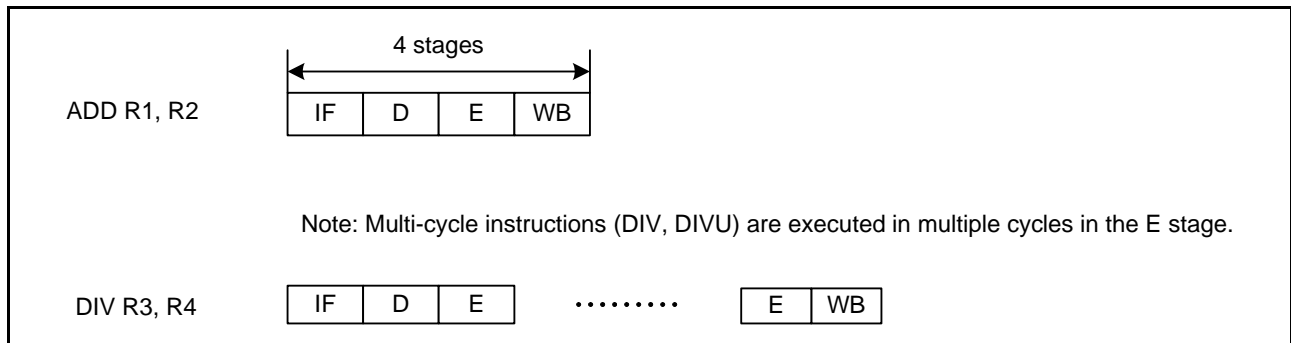


Figure 2.11 Operation for Register-Register, Immediate-Register

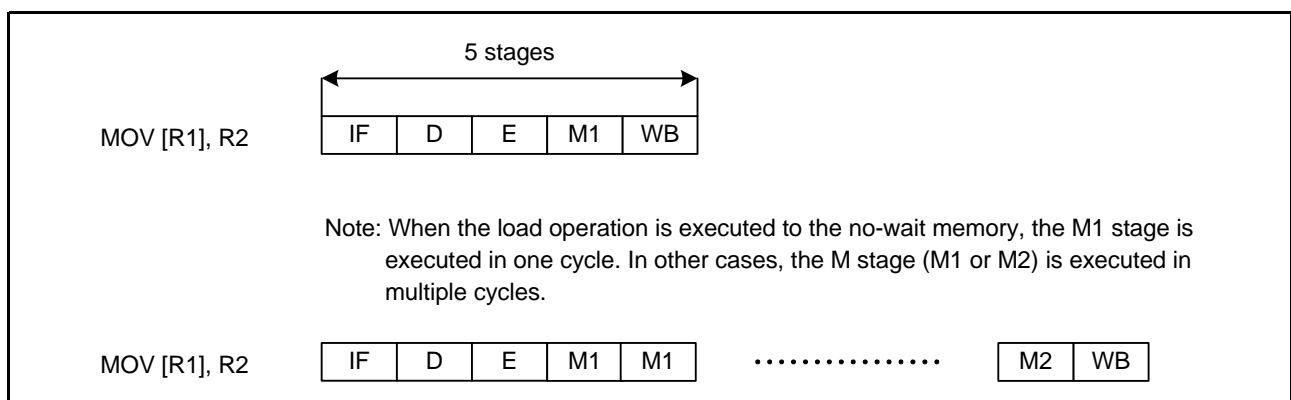


Figure 2.12 Load Operation

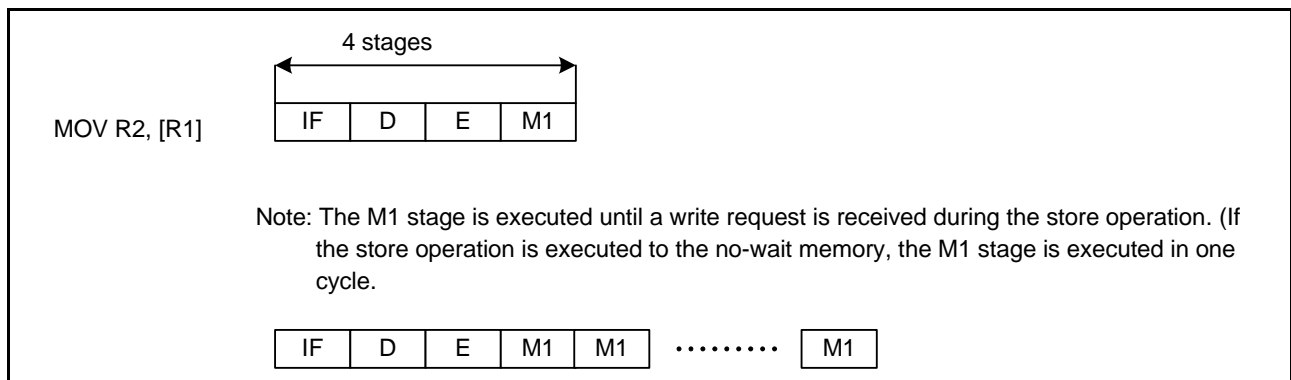


Figure 2.13 Store Operation

2.8.2.2 Instructions Converted into Multiple Micro-Operations and Pipeline Processing

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

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

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
Arithmetic/logic instructions (memory source operand)	• {ADC, ADD, AND, CMP, MAX, MIN, MUL, OR, SBB, SUB, TST, XOR} “[Rs], Rd” / “dsp[Rs], Rd”	Figure 2.14	3
Arithmetic/logic instructions (division)	• DIV “[Rs], Rd/dsp[Rs], Rd”	—	5 to 22
	• DIVU “[Rs], Rd/dsp[Rs], Rd”	—	4 to 20
Arithmetic/logic instructions (multiplier: 32 x 32 → 64 bits) (register-register, register-immediate)	• {EMUL, EMULU} “#IMM, Rd” / “Rs, Rd”	Figure 2.16	2
Arithmetic/logic instructions (multiply-and-accumulate operation)	• RMPA.B	—	$6+7 \times \text{floor}(n/4)+4 \times (n\%4)$ n: Number of processing bytes*1
	• RMPA.W	—	$6+5 \times \text{floor}(n/2)+4 \times (n\%2)$ n: Number of processing words*1
	• RMPA.L	—	$6+4n$ n: Number of processing longwords*1
Arithmetic/logic instructions (64-bit signed saturation for RMPA instruction)	• SATR	—	3
Data transfer instructions (memory-memory transfer)	• MOV “[Rs], [Rd]” / “dsp[Rs], [Rd]” / “[Rs], dsp[Rd]” / “dsp[Rs], [Rd]” • PUSH “[Rs]” / “dsp[Rs]”	Figure 2.15	3
Bit manipulation instructions (memory source operand)	• {BCLR, BNOT, BSET, BTST} “#IMM, [Rd]” / “#IMM, dsp[Rd]” • BMCnd “#IMM, [Rd]” / “#IMM, dsp[Rd]”	Figure 2.15	3
Transfer instructions (load operation)	• POPC “CR”	—	Throughput: 3 Latency: 4*2
Transfer instructions (store operation of multiple registers)	• PUSHM “Rs-Rs2”	—	n n: Number of registers*3
Transfer instructions (store operation of multiple registers)	• POPM “Rs-Rs2”	—	Throughput: n Latency: n + 1 n: Number of registers*2*4
Transfer instructions (register-register)	• XCHG “Rs, Rd”	Figure 2.17	2
Transfer instructions (memory-register)	• XCHG “[Rs], Rd” / “dsp[Rs], Rd”	Figure 2.18	2
Branch instructions	• RTS	—	5
	• RTSD “#IMM”	—	5
	• RTSD “#IMM, Rd-Rd2”	—	Throughput: $n < 5 ? 5 : 1 + n$ Latency: $n < 4 ? 5 : 2 + n$ n: Number of registers*2

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

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
String manipulation instructions*5	• SCMPU	—	$2+4 \times \text{floor}(n/4)+4 \times (n\%4)$ 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
	• SMOV, 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
Floating-point operation instructions (register-register, immediate-register)	• {FADD, FSUB} "#IMM, Rd"/"Rs, Rd"	Figure 2.19	4
	• FMUL "#IMM, Rd"/"Rs, Rd"	—	3
	• FDIV "#IMM, Rd"/"Rs, Rd"	—	16
	• {FTOI, ROUND, ITOF} "Rs, Rd"	—	2
Floating-point operation instructions (memory source operand)	• {FADD, FSUB} "[Rs], Rd"/"dsp[Rs], Rd"	—	6
	• FMUL "[Rs], Rd"/"dsp[Rs], Rd"	—	5
	• FDIV "[Rs], Rd"/"dsp[Rs], Rd"	—	18
	• {FTOI, ROUND, ITOF} "[Rs], Rd" / "dsp[Rs], Rd"	—	4
System manipulation instructions	• RTE	—	6
	• RTFI	—	3

[Legend] ? : Conditional operator

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

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

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

Note 4. The POPM instruction is converted into multiple 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 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 2.14 to Figure 2.19 show the operation of instructions that are converted into basic multiple micro-operations.

[Legend]

mop: Micro-operation, stall: Pipeline stall

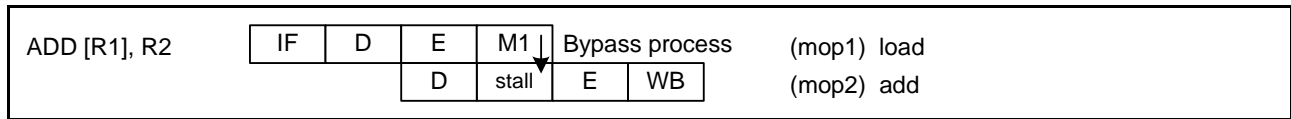


Figure 2.14 Arithmetic/Logic Instruction (Memory Source Operand)

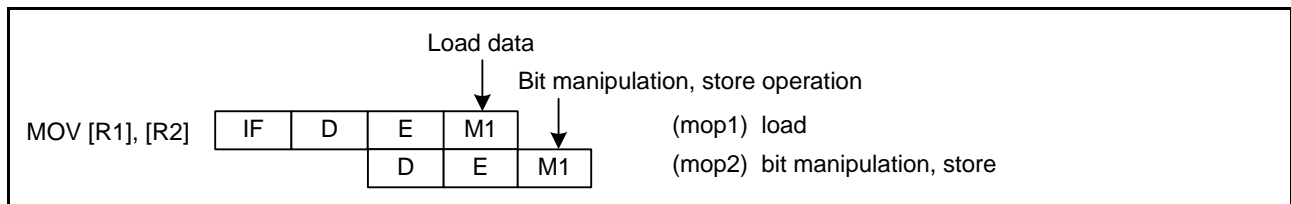


Figure 2.15 OV Instruction (Memory-Memory), Bit Manipulation Instruction (Memory Source Operand)

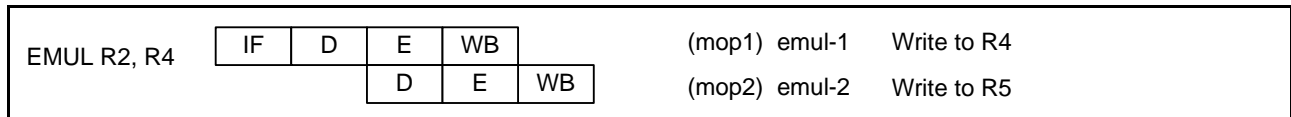


Figure 2.16 EMUL, EMULU Instructions (Register- Register, Register-Immediate)

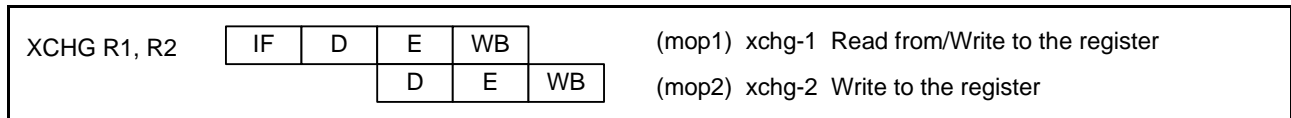


Figure 2.17 XCHG Instruction (Registers)

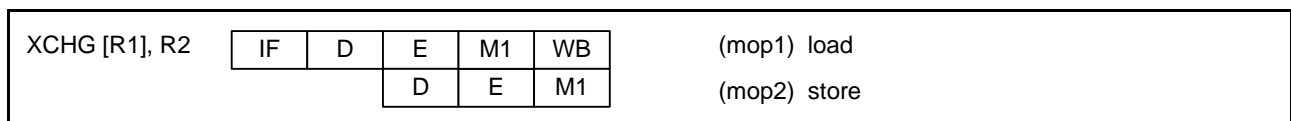


Figure 2.18 XCHG Instruction (Memory Source Operand)

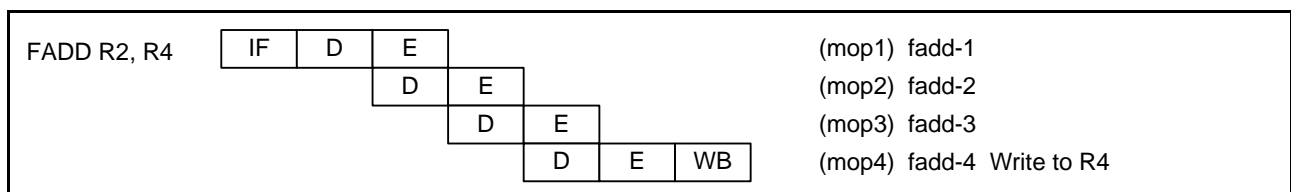


Figure 2.19 Floating-Point Operation Instruction (Register-Register, Immediate-Register)

2.8.2.3 Pipeline Basic Operation

In the ideal pipeline processing, each stage is executed in one cycle, though all instructions may not be pipelined in due to the processing 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.

[Legend]

mop: Micro-operation, stall: Pipeline stall

(1) Pipeline Flow with Stalls

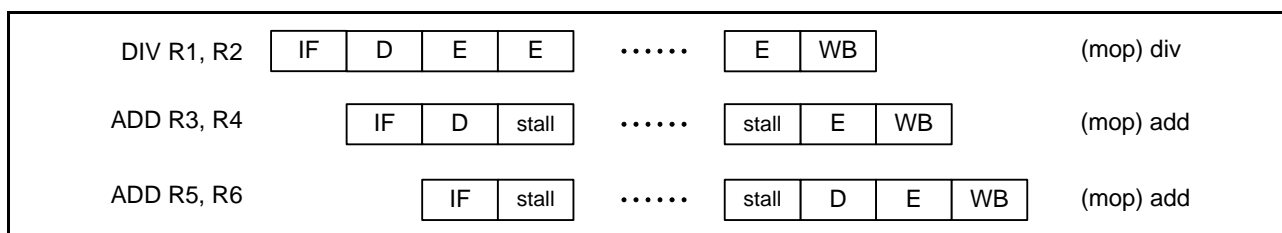


Figure 2.20 When an Instruction which Requires Multiple Cycles is Executed in the E Stage

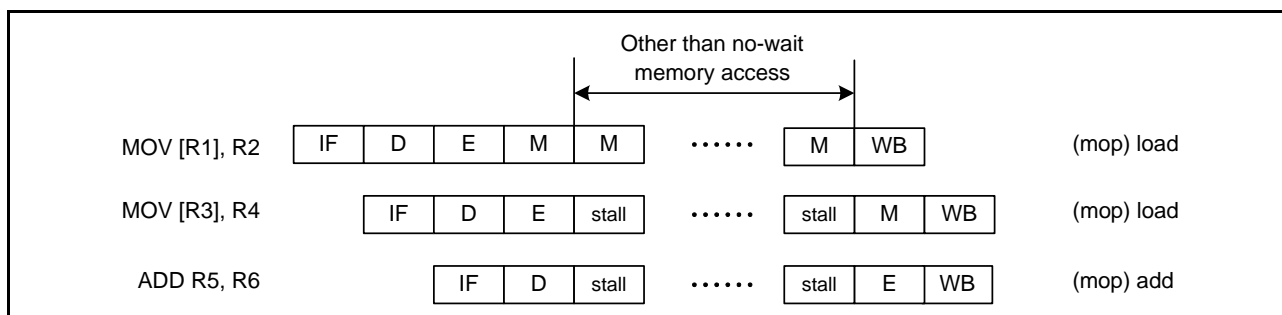


Figure 2.21 When an Instruction which Requires more than One Cycle for its Operand Access is Executed

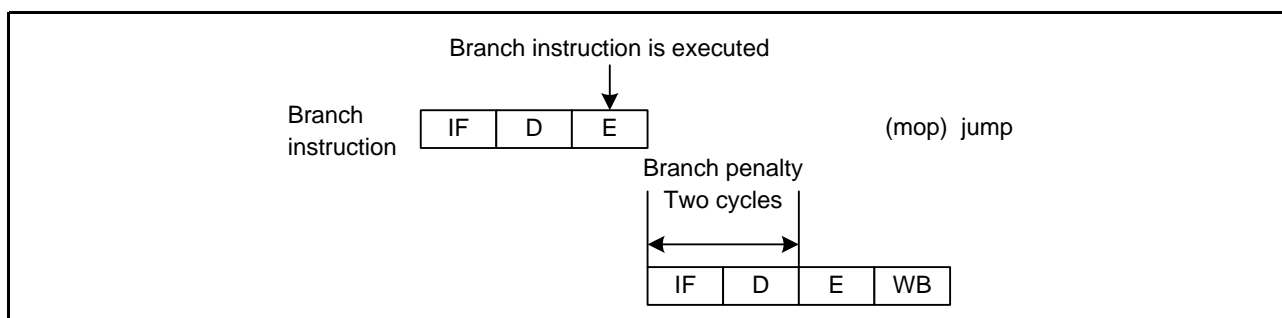


Figure 2.22 When a Branch Instruction is Executed (While the Condition is Satisfied for Unconditional/ Conditional Branch Instruction)

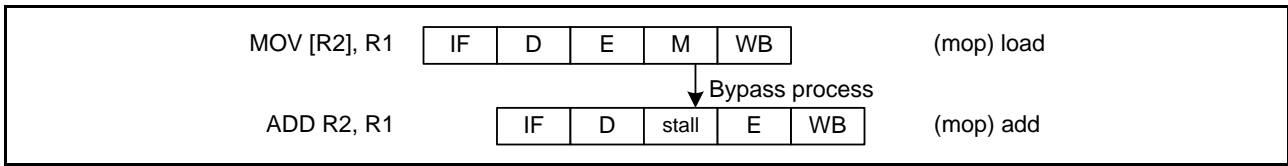


Figure 2.23 When the Subsequent Instruction Uses an Operand Read from the Memory

(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.

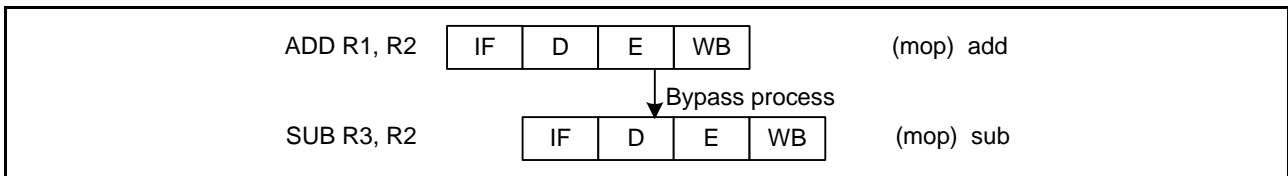


Figure 2.24 Bypass Process

(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.

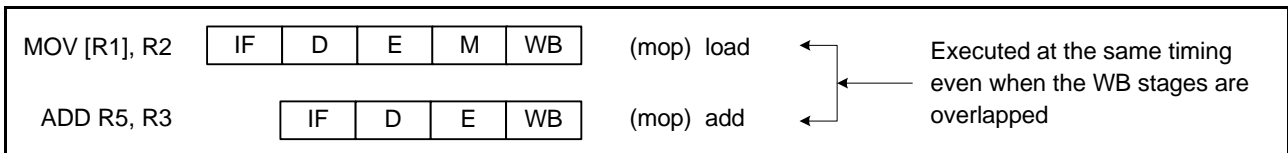


Figure 2.25 When WB Stages for the Memory Load and for the Operation are Overlapped

(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.

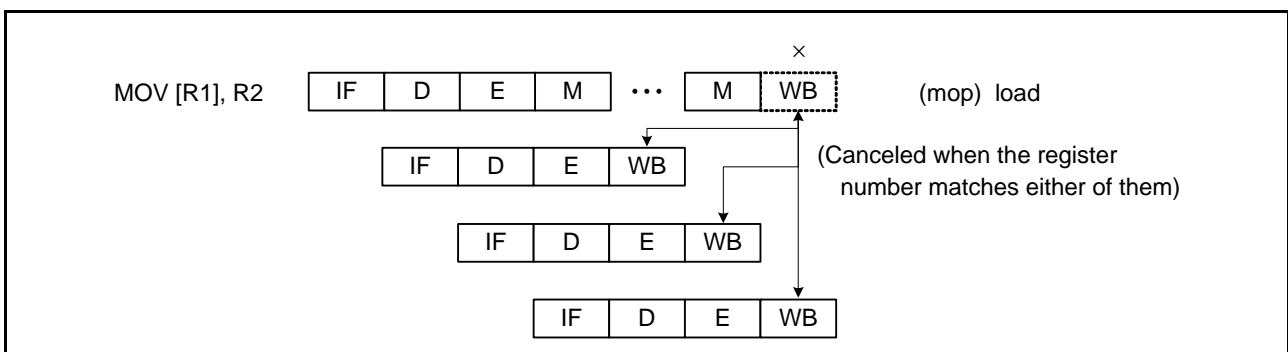


Figure 2.26 When Subsequent Instruction Writes to the Same Register before the End of Memory Load

(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).

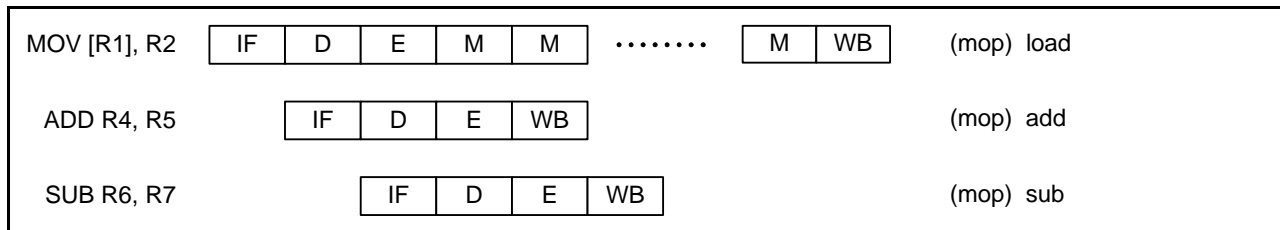


Figure 2.27 When Load Data is not Used by the Subsequent Instruction

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. The number of memory access cycles in the RX62N/RX621 Group is on a per-product basis.

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	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 always processed with no waiting. The on-chip RAM and ROM in products of the RX62N/RX621 Groups always allows such access. Numbers of cycles for response to interrupts can be minimized by placing program code (and vectors) in on-chip ROM and the stack in on-chip RAM. Branching to the start of the exception handling routine

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

The timing of interrupt acceptance depends on the state of the pipelines. For more information on this, see section 10.3.1, Timing of Acceptance and Saved PC Values.

3. Operating Modes

3.1 Operating Mode Types and Selection

The RX62N/RX621 Group has five types of operating modes. Operating modes are specified by the MD1 and MD0 pins and the ROME and EXBE bits in the system control register 0 (SYSCR0).

The endian of the RX62N/RX621 Group can be selected in each operating mode. Endian is specified by the MDE pin. For the endian of the RX62N/RX621 Group, see section 2.5, Endian, and section 12, Buses.

Note: Do not change the state of pins MDE, MD1, and MD0 while the LSI is working. Do not select any combination other than those specified in table 3.1.

Table 3.1 Selection of Operating Modes by the Mode Pins

Mode Pin		SYSCR0 Initial State		Operating Mode	On-Chip ROM*	External Bus
MD1	MD0	ROME	EXBE			
0	1	1	0	Boot mode	Enabled	Disabled
1	0	1	0	USB boot mode	Enabled	Disabled
1	1	1	0	Single-chip mode	Enabled	Disabled

Note: The on-chip ROM is classified into two flash memories: ROM and data flash. For details, see section 38, ROM (Flash Memory for Code Storage), and section 39, Data Flash Memory (Flash Memory for Data Storage).

Table 3.2 Selection of Operating Modes by Register Setting

SYSCR0		Operating Mode	On-Chip ROM*	External Bus
ROME	EXBE			
0	0	Single-chip mode	Disabled	Disabled
1	0		Enabled	Disabled
0	1	On-chip ROM disabled extended mode	Disabled	Enabled
1	1	On-chip ROM enabled extended mode	Enabled	Enabled

Note: The on-chip ROM is classified into two flash memories: ROM and data flash. For details, see section 38, ROM (Flash Memory for Code Storage), and section 39, Data Flash Memory (Flash Memory for Data Storage).

Table 3.3 Selection of Endian

Mode Pin	Endian
MDE	
0	Little endian
1	Big endian

3.2 Register Descriptions

Table 3.4 lists the registers related to operating modes.

Table 3.4 Registers Related to Operating Modes

Register Name	Symbol	Value after Reset	Address	Access Size
Mode monitor register	MDMONR	10000000 x00000xxb	0008 0000h	16
Mode status register	MDSR	00000000 00001001b	0008 0002h	16
System control register 0	SYSCR0	00000000 00000001b	0008 0006h	16
System control register 1	SYSCR1	00000000 00000001b	0008 0008h	16

3.2.1 Mode Monitor Register (MDMONR)

Address: 0008 0000h

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0
	b7	b6	b5	b4	b3	b2	b1	b0
	MDE	—	—	—	—	—	MD1	MD0
Value after reset:	x*	0	0	0	0	0	x*	x*

Note: * Depends on the setting of the mode pins (MDE, MD1, and MD0).

Bit	Symbol	Bit Name	Description	R/W
b0	MD0	MD0 Status Flag	0: The MD0 pin is 0 1: The MD0 pin is 1	R
b1	MD1	MD1 Status Flag	0: The MD1 pin is 0 1: The MD1 pin is 1	R
b6 to b2	—	Reserved	These bits are always read as 0. Writing to these bits has no effect.	R
b7	MDE	MDE Status Flag	0: The MDE pin is 0 (little endian) 1: The MDE pin is 1 (big endian)	R
b14 to b8	—	Reserved	These bits are always read as 0. Writing to these bits has no effect.	R
b15	—	Reserved	This bit is always read as 1. Writing to this bit has no effect.	R

MDMONR indicates the status of the mode pins.

3.2.2 Mode Status Register (MDSR)

Address: 0008 0002h

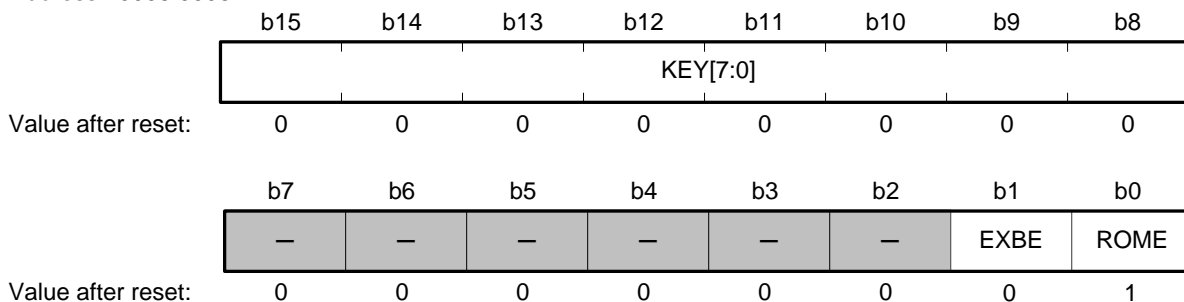
	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0
	b7	b6	b5	b4	b3	b2	b1	b0
	—	UBTS	—	BOTS	BSW[1:0]	EXB	IROM	
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	IROM	On-Chip ROM Startup Status Flag	0: The on-chip ROM is disabled at startup 1: The on-chip ROM is enabled at startup	R
b1	EXB	External Bus Startup Status Flag	0: The external bus is disabled at startup 1: The external bus is enabled at startup	R
b3, b2	BSW[1:0]	External Bus Width Flags	0 0: The 16-bit bus is activated 0 1: Reserved 1 0: The 8-bit bus is activated 1 1: Reserved	R
b4	BOTS	Boot Mode Startup Flag	0: Started with a mode except boot mode 1: Started with boot mode	R
b5	—	Reserved	This bit is always read as 0. Writing to this bit has no effect.	R
b6	UBTS	USB Boot Mode Startup Flag	0: Started with a mode except USB boot mode 1: Started with USB boot mode	R
b15 to b7	—	Reserved	These bits are always read as 0. Writing to these bits has no effect.	R

MDSR indicates the internal status of this LSI at startup.

3.2.3 System Control Register 0 (SYSCR0)

Address: 0008 0006h



Bit	Symbol	Bit Name	Description	R/W
b0	ROME	On-Chip ROM Enable	0: The on-chip ROM is disabled 1: The on-chip ROM is enabled	R/W
b1	EXBE	External Bus Enable	0: The external bus is disabled 1: The external bus is enabled	R/W
b7 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b15 to b8	KEY[7:0]	SYSCR0 Key Code	5Ah: Modifying SYSCR0 is enabled Other codes: Modifying SYSCR0 is disabled These bits are always read as 00h.	R/W

SYSCR0 is used to enable or disable the on-chip ROM and the external bus.

ROME Bit (On-Chip ROM Enable)

The ROME bit enables or disables the on-chip ROM (ROM, data flash).

While this bit is 1, it can be cleared to 0. While this bit is 0, it cannot be set to 1. Once the on-chip ROM is disabled by clearing this bit to 0, the on-chip ROM can no longer be enabled with the ROME bit. A 0 should not be written to this bit during access to the on-chip ROM.

After writing a 0 to this bit to disable the on-chip ROM, always make sure that the ROME bit has been changed to 0 before proceeding to the next processing.

EXBE Bit (External Bus Enable)

The EXBE bit enables* or disables the external bus.

Write 0 to this bit while the external bus cycle is not performed.

Note that the external bus and the internal bus are concurrently activated in some cases. When the EXBE bit is changed, the bus should be accessed after the change in the EXBE bit is completed.

When the EXBE bit is changed, the I/O port setting should also be changed simultaneously. For details, see section 17, I/O Ports.

KEY[7:0] Bits (SYSCR0 Key Code)

The KEY[7:0] bits enable or disable modifying SYSCR0.

When writing a value to the ROME or EXBE bit, write 5Ah to the KEY[7:0] bits simultaneously. If SYSCR0 is modified with a KEY[7:0] value other than 5Ah, the ROME and EXBE values remain unchanged.

3.2.4 System Control Register 1 (SYSCR1)

Address: 0008 0008h

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

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

SYSCR1 is used to enable or disable the on-chip RAM.

RAME Bit (RAM Enable)

The RAME bit enables or disables the on-chip RAM.

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

A 0 should not be written to this bit during access to the on-chip RAM.

When accessing the on-chip RAM immediately after changing the RAME bit from 0 (on-chip RAM disabled) to 1 (on-chip RAM enabled), always make sure that the RAME bit is 1 before the access.

Even when the RAME bit is cleared to 0, the on-chip RAM retains its value. To retain the value in the on-chip RAM, keep the specified RAM standby voltage (VRAM). For details, see section 41, Electrical Characteristics.

3.3 Details of Operating Modes

3.3.1 Single-Chip Mode

In this mode, the on-chip ROM is enabled or disabled, the external bus is disabled (SYSCR0.EXBE bit = 0), and all I/O ports can be used as input/output ports.

The on-chip ROM is enabled when this LSI is started. While the on-chip ROM is enabled (ROME bit = 1 in SYSCR0), it can be disabled by clearing the ROME bit in SYSCR0 to 0. While the on-chip ROM is disabled (ROME bit = 0 in SYSCR0), it cannot be enabled by setting the ROME bit in SYSCR0 to 1.

3.3.2 On-Chip ROM Enabled Extended Mode

In this mode, the on-chip ROM is enabled (ROME bit = 1 in SYSCR0) and the external bus is available as external extended mode (EXBE bit = 1 in SYSCR0). This mode allows some I/O ports to be used as data bus input/output, address bus output, or bus control signal input/output. For details, see section 17, I/O Ports.

The external bus width can be changed by the setting of external bus width selection (BSIZE[1:0] bits in CSnCR (n = 0 to 7)). For details, see section 12, Buses.

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

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

3.3.3 On-Chip ROM Disabled Extended Mode

In this mode, the on-chip ROM is disabled (ROME bit = 0 in SYSCR0) and the external bus is available as external extended mode (EXBE bit = 1 in SYSCR0). This mode allows some I/O ports to be used as data bus input/output, address bus output, or bus control signal input/output. For details, see section 17, I/O Ports.

The external bus width can be changed by the setting of external bus width selection (BSIZE[1:0] bits in CSnCR (n = 0 to 7)). For details, see section 12, Buses.

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

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

3.3.4 Boot Mode

Boot mode is provided for the flash memory. This mode functions in the same manner as single-chip mode except for data write/erase to the flash memory. For details, see section 38, ROM (Flash Memory for Code Storage), and section 39, Data Flash Memory (Flash Memory for Data Storage).

3.3.5 USB Boot Mode

USB boot mode is provided for the flash memory. This mode functions in the same manner as single-chip mode except for data write/erase to the flash memory. For details, see section 38, ROM (Flash Memory for Code Storage), and section 39, Data Flash Memory (Flash Memory for Data Storage).

3.4 Transitions of Operating Modes

3.4.1 Operating Mode Transitions According to Mode Pin Setting

Figure 3.1 shows operating mode transitions according to the setting of pins MD1 and MD0. Operating modes can shift in the direction of arrow.

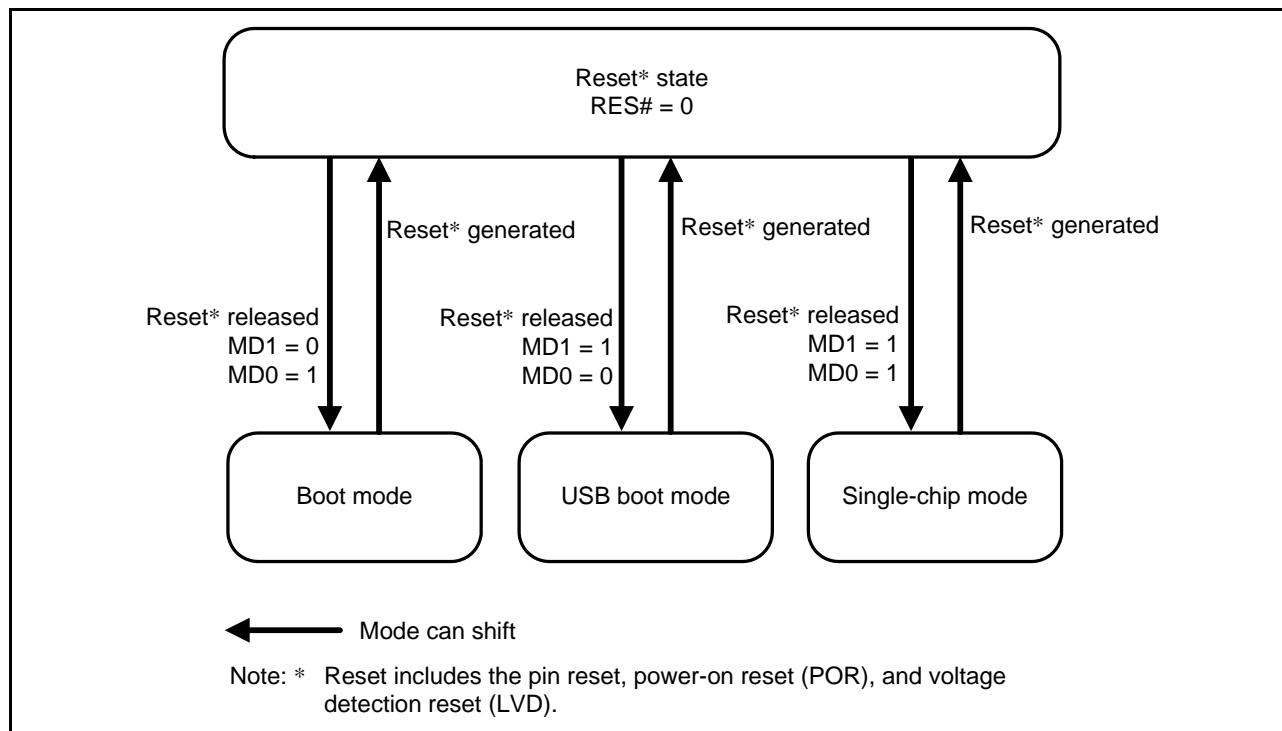


Figure 3.1 Setting of Pins MD1 and MD0 and Operating Modes

3.4.2 Operating Mode Transitions According to Register Setting

Figure 3.2 shows operating mode transitions according to the setting of the ROME and EXBE bits in SYSCR0. Operating modes can shift in the direction of arrow.

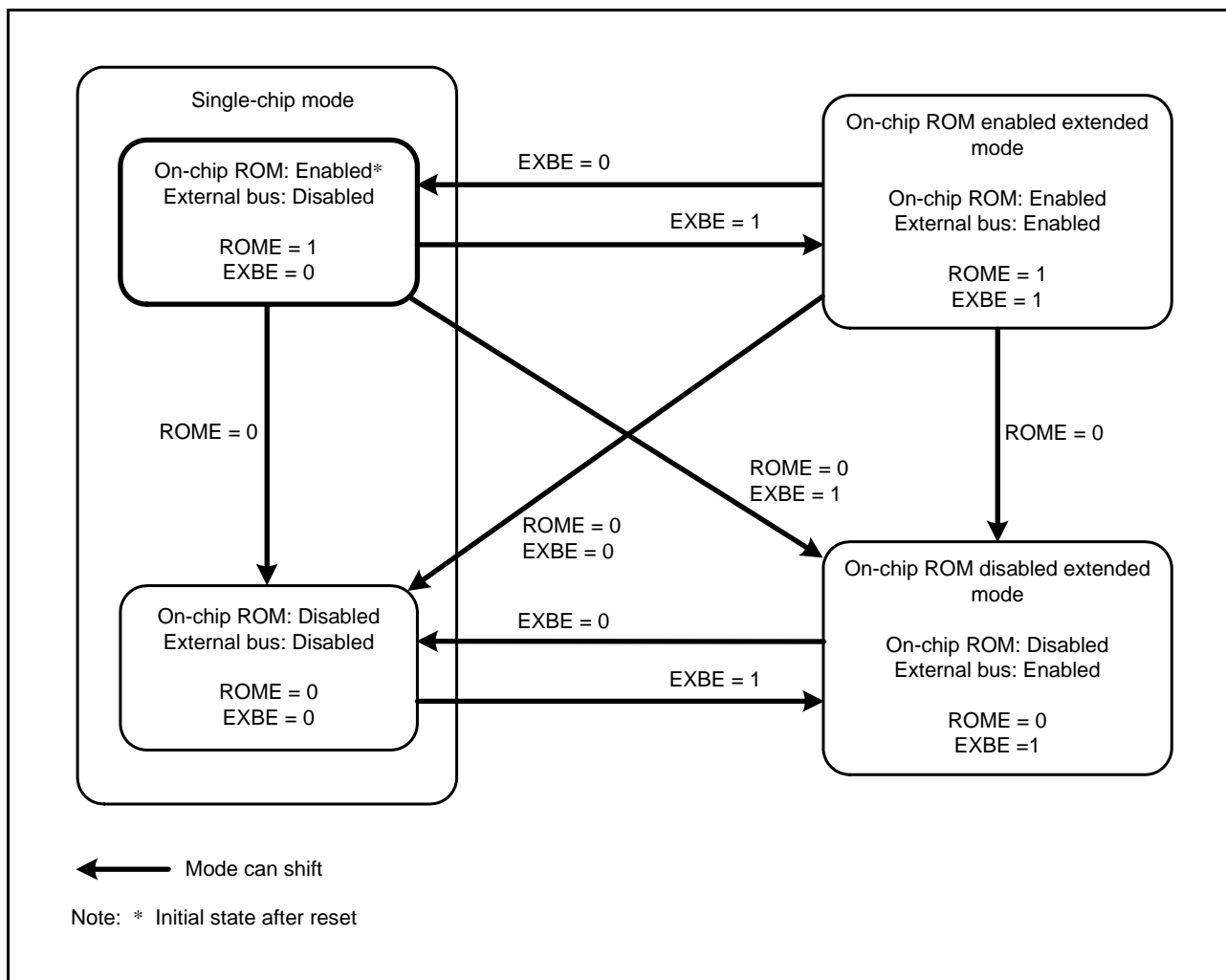


Figure 3.2 Setting of Bits ROME and EXBE and Operating Modes

4. Address Space

4.1 Address Space

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

Figure 4.1 shows the memory maps in the respective operating modes. Accessible areas will differ according to the operating mode and states of control bits.

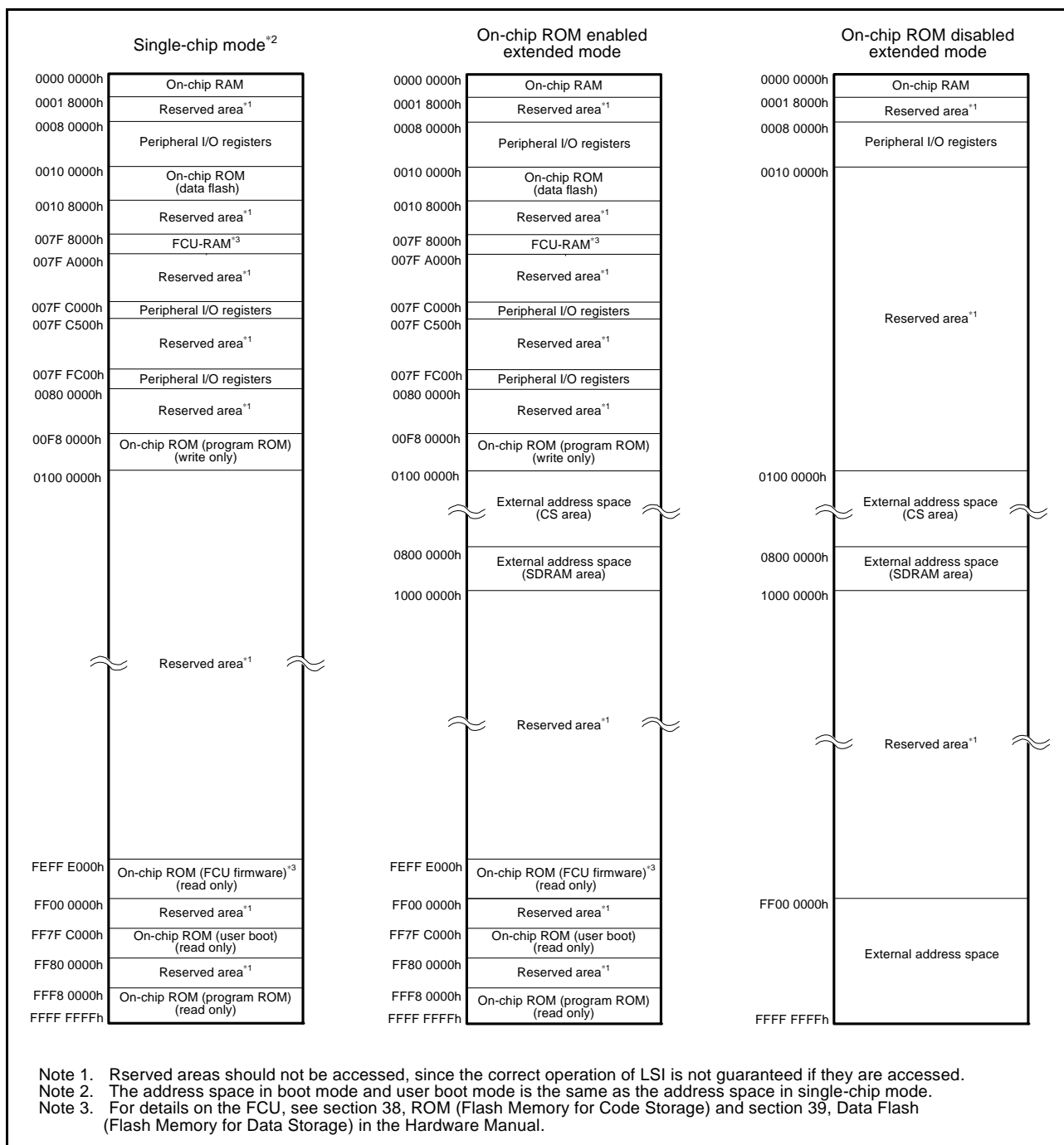


Figure 4.1 Memory Map in Each Operating Mode

4.2 External Address Space

The external address space is classified into CS areas (CS0 to CS7) and SDRAM area (SDCS).

The CS area is divided into up to 8 areas (CS0 to CS7), each corresponding to the CSi# signal output from a CSi# (i = 0 to 7) pin.

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

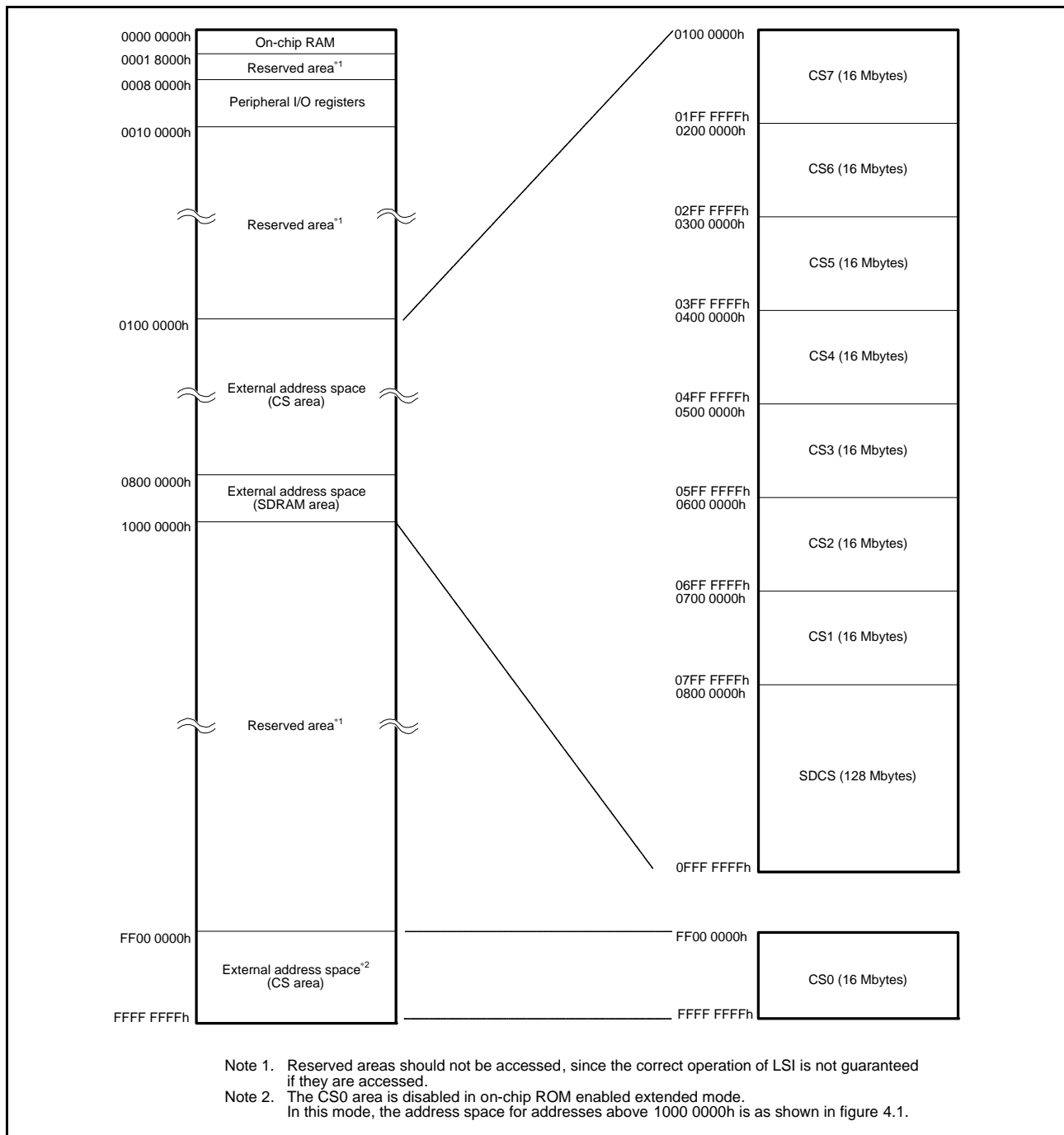


Figure 4.2 Correspondence between External Address Spaces, CS Areas (CS0 to CS7), and SDRAM area (SDCS) (In On-Chip ROM Disabled Extended Mode)

5. I/O Registers

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

1. I/O register addresses (address order)

- Registers are listed from the lower allocation addresses.
- Registers are classified according to functional modules (abbreviations).
- The number of access cycles indicates the number of states based on the specified reference clock.
- Among the internal I/O register area, addresses not listed in the list of registers are reserved. Reserved addresses must not be accessed. Do not access these addresses; otherwise, the operation when accessing these bits and subsequent operations cannot be guaranteed.
- A unit of access is specified for each register. Access other than in the specified unit is prohibited.

2. I/O register bits

- Bit configurations of the registers are listed in the same order as the register addresses.
- Reserved bits are indicated by "—" in the bit name column.
- Space in the bit name field indicates that the entire register is allocated to either the counter or data.
- For the registers of 16 or 32 bits, the MSB is listed first.

3. Notes on writing to I/O registers

When writing to an I/O register, the CPU starts executing the subsequent instruction before completing I/O register write. This may cause the subsequent instruction to be executed before the post-update I/O register value is reflected on the operation.

As described in the following examples, special care is required for the cases in which the subsequent instruction must be executed after the post-update I/O register value is actually reflected.

[Examples of cases requiring special care]

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

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

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

[Instruction examples]

- Byte-size I/O registers


```
MOV.L #SFR_ADDR, R1
MOV.B #SFR_DATA, [R1]
CMP [R1].UB, R1
;; Next process
```
- Word-size I/O registers


```
MOV.L #SFR_ADDR, R1
MOV.W #SFR_DATA, [R1]
CMP [R1].W, R1
;; Next process
```
- Longword-size I/O registers


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

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

4. Number of Access Cycles to I/O Registers

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

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

The number of bus cycles of internal peripheral bus 1 to 6 differs according to the register to be accessed. For the number of access cycles to each I/O register, see [table 5.1, List of I/O Registers](#).

When peripheral functions connected to internal peripheral bus 6 or registers for the external bus control unit (except for bus error related registers) are accessed, the number of divided cycles for clock synchronization is added.

Although the number of divided cycles for clock synchronization differs depending on the number of frequency ratio or bus access timing, the sum of the number of bus cycles for internal main bus 1 and the number of divided cycles for clock synchronization will be one PCLK (or BCLK) at a maximum. Therefore, one PCLK (or BCLK) is added to the number of access cycles shown in [table 5.1](#).

Note: 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 (DMACA or DTC).

5.1 I/O Register Addresses (Address Order)

Table 5.1 List of I/O Registers (Address Order) (1 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 0000h	SYSTEM	Mode monitor register	MDMONR	16	16	3 ICLK
0008 0002h	SYSTEM	Mode status register	MDSR	16	16	3 ICLK
0008 0006h	SYSTEM	System control register 0	SYSCR0	16	16	3 ICLK
0008 0008h	SYSTEM	System control register 1	SYSCR1	16	16	3 ICLK
0008 000Ch	SYSTEM	Standby control register	SBYCR	16	16	3 ICLK
0008 0010h	SYSTEM	Module stop control register A	MSTPCRA	32	32	3 ICLK
0008 0014h	SYSTEM	Module stop control register B	MSTPCRB	32	32	3 ICLK
0008 0018h	SYSTEM	Module stop control register C	MSTPCRC	32	32	3 ICLK
0008 0020h	SYSTEM	System clock control register	SCKCR	32	32	3 ICLK
0008 0030h	SYSTEM	External bus clock control register	BCKCR	8	8	3 ICLK
0008 0040h	SYSTEM	Oscillation stop detection control register	OSTDCR	16	16	3 ICLK
0008 1300h	BSC	Bus error status clear register	BERCLR	8	8	2 ICLK
0008 1304h	BSC	Bus error monitoring enable register	BEREN	8	8	2 ICLK
0008 1308h	BSC	Bus error status register 1	BERSR1	8	8	2 ICLK
0008 130Ah	BSC	Bus error status register 2	BERSR2	16	16	2 ICLK
0008 2000h	DMAC0	DMA source address register	DMSAR	32	32	2 ICLK
0008 2004h	DMAC0	DMA destination address register	DMDAR	32	32	2 ICLK
0008 2008h	DMAC0	DMA transfer count register	DMCRA	32	32	2 ICLK
0008 200Ch	DMAC0	DMA block transfer count register	DMCRB	16	16	2 ICLK
0008 2010h	DMAC0	DMA transfer mode register	DMTMD	16	16	2 ICLK
0008 2013h	DMAC0	DMA interrupt setting register	DMINT	8	8	2 ICLK
0008 2014h	DMAC0	DMA address mode register	DMAMD	16	16	2 ICLK
0008 2018h	DMAC0	DMA offset register	DMOFR	32	32	2 ICLK
0008 201Ch	DMAC0	MA transfer enable register	DMCNT	8	8	2 ICLK
0008 201Dh	DMAC0	DMA software start register	DMREQ	8	8	2 ICLK
0008 201Eh	DMAC0	DMA status register	DMSTS	8	8	2 ICLK
0008 201Fh	DMAC0	DMA activation source flag control register	DMCSL	8	8	2 ICLK
0008 2040h	DMAC1	DMA source address register	DMSAR	32	32	2 ICLK
0008 2044h	DMAC1	DMA destination address register	DMDAR	32	32	2 ICLK
0008 2048h	DMAC1	DMA transfer count register	DMCRA	32	32	2 ICLK
0008 204Ch	DMAC1	DMA block transfer count register	DMCRB	16	16	2 ICLK
0008 2050h	DMAC1	DMA transfer mode register	DMTMD	16	16	2 ICLK
0008 2053h	DMAC1	DMA interrupt setting register	DMINT	8	8	2 ICLK
0008 2054h	DMAC1	DMA address mode register	DMAMD	16	16	2 ICLK
0008 205Ch	DMAC1	MA transfer enable register	DMCNT	8	8	2 ICLK
0008 205Dh	DMAC1	DMA software start register	DMREQ	8	8	2 ICLK
0008 205Eh	DMAC1	DMA status register	DMSTS	8	8	2 ICLK
0008 205Fh	DMAC1	DMA activation source flag control register	DMCSL	8	8	2 ICLK
0008 2080h	DMAC2	DMA source address register	DMSAR	32	32	2 ICLK
0008 2084h	DMAC2	DMA destination address register	DMDAR	32	32	2 ICLK
0008 2088h	DMAC2	DMA transfer count register	DMCRA	32	32	2 ICLK

Table 5.1 List of I/O Registers (Address Order) (2 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 208Ch	DMAC2	DMA block transfer count register	DMCRB	16	16	2 ICLK
0008 2090h	DMAC2	DMA transfer mode register	DMTMD	16	16	2 ICLK
0008 2093h	DMAC2	DMA interrupt setting register	DMINT	8	8	2 ICLK
0008 2094h	DMAC2	DMA address mode register	DMAMD	16	16	2 ICLK
0008 209Ch	DMAC2	DMA transfer enable register	DMCNT	8	8	2 ICLK
0008 209Dh	DMAC2	DMA software start register	DMREQ	8	8	2 ICLK
0008 209Eh	DMAC2	DMA status register	DMSTS	8	8	2 ICLK
0008 209Fh	DMAC2	DMA activation source flag control register	DMCSL	8	8	2 ICLK
0008 20C0h	DMAC3	DMA source address register	DMSAR	32	32	2 ICLK
0008 20C4h	DMAC3	DMA destination address register	DMDAR	32	32	2 ICLK
0008 20C8h	DMAC3	DMA transfer count register	DMCRA	32	32	2 ICLK
0008 20CCh	DMAC3	DMA block transfer count register	DMCRB	16	16	2 ICLK
0008 20D0h	DMAC3	DMA transfer mode register	DMTMD	16	16	2 ICLK
0008 20D3h	DMAC3	DMA interrupt setting register	DMINT	8	8	2 ICLK
0008 20D4h	DMAC3	DMA address mode register	DMAMD	16	16	2 ICLK
0008 20DCh	DMAC3	DMA transfer enable register	DMCNT	8	8	2 ICLK
0008 20DDh	DMAC3	DMA software start register	DMREQ	8	8	2 ICLK
0008 20DEh	DMAC3	DMA status register	DMSTS	8	8	2 ICLK
0008 20DFh	DMAC3	DMA activation source flag control register	DMCSL	8	8	2 ICLK
0008 2200h	DMAC	DMACA start register	DMAST	8	8	2 ICLK
0008 2400h	DTC	DTC control register	DTCCR	8	8	2 ICLK
0008 2404h	DTC	DTC vector base register	DTCVBR	32	32	2 ICLK
0008 2408h	DTC	DTC address mode register	DTCADMOD	8	8	2 ICLK
0008 240Ch	DTC	DTC module start register	DTCST	8	8	2 ICLK
0008 240Eh	DTC	DTC status register	DTCSTS	16	16	2 ICLK
0008 2800h	EXDMAC0	EXDMA source address register	EDMSAR	32	32	1 to 2 BCLK*8
0008 2804h	EXDMAC0	EXDMA destination address register	EDMDAR	32	32	1 to 2 BCLK*8
0008 2808h	EXDMAC0	EXDMA transfer count register	EDMCRA	32	32	1 to 2 BCLK*8
0008 280Ch	EXDMAC0	EXDMA block transfer count register	EDMCRB	16	16	1 to 2 BCLK*8
0008 2810h	EXDMAC0	EXDMA transfer mode register	EDMTMD	16	16	1 to 2 BCLK*8
0008 2812h	EXDMAC0	EXDMA output setting register	EDMOMD	8	8	1 to 2 BCLK*8
0008 2813h	EXDMAC0	EXDMA interrupt setting register	EDMINT	8	8	1 to 2 BCLK*8
0008 2814h	EXDMAC0	EXDMA address mode register	EDMAMD	32	32	1 to 2 BCLK*8
0008 2818h	EXDMAC0	EXDMA output setting register	EDMOFR	32	32	1 to 2 BCLK*8
0008 281Ch	EXDMAC0	EXDMA transfer enable register	EDMCNT	8	8	1 to 2 BCLK*8
0008 281Dh	EXDMAC0	EXDMA software start register	EDMREQ	8	8	1 to 2 BCLK*8
0008 281Eh	EXDMAC0	EXDMA status register	EDMSTS	8	8	1 to 2 BCLK*8
0008 2820h	EXDMAC0	EXDMA external request sense mode register	EDMRMD	8	8	1 to 2 BCLK*8
0008 2821h	EXDMAC0	EXDMA external request flag register	EDMERF	8	8	1 to 2 BCLK*8
0008 2822h	EXDMAC0	EXDMA peripheral request flag register	EDMPRF	8	8	1 to 2 BCLK*8
0008 2840h	EXDMAC1	EXDMA source address register	EDMSAR	32	32	1 to 2 BCLK*8
0008 2844h	EXDMAC1	EXDMA destination address register	EDMDAR	32	32	1 to 2 BCLK*8
0008 2848h	EXDMAC1	EXDMA transfer count register	EDMCRA	32	32	1 to 2 BCLK*8

Table 5.1 List of I/O Registers (Address Order) (3 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 284Ch	EXDMAC1	EXDMA block transfer count register	EDMCRB	16	16	1 to 2 BCLK*8
0008 2850h	EXDMAC1	EXDMA transfer mode register	EDMTMD	16	16	1 to 2 BCLK*8
0008 2852h	EXDMAC1	EXDMA output setting register	EDMOMD	8	8	1 to 2 BCLK*8
0008 2853h	EXDMAC1	EXDMA interrupt setting register	EDMINT	8	8	1 to 2 BCLK*8
0008 2854h	EXDMAC1	EXDMA address mode register	EDMAMD	32	32	1 to 2 BCLK*8
0008 285Ch	EXDMAC1	EXDMA transfer enable register	EDMCNT	8	8	1 to 2 BCLK*8
0008 285Dh	EXDMAC1	EXDMA software start register	EDMREQ	8	8	1 to 2 BCLK*8
0008 285Eh	EXDMAC1	EXDMA status register	EDMSTS	8	8	1 to 2 BCLK*8
0008 2860h	EXDMAC1	EXDMA external request sense mode register	EDMRMD	8	8	1 to 2 BCLK*8
0008 2861h	EXDMAC1	EXDMA external request flag register	EDMERF	8	8	1 to 2 BCLK*8
0008 2862h	EXDMAC1	EXDMA peripheral request flag register	EDMPRF	8	8	1 to 2 BCLK*8
0008 2A00h	EXDMAC	EXDMA module start register	EDMAST	8	8	1 to 2 BCLK*8
0008 2BE0h	EXDMAC	Cluster buffer register 0	CLSBR0	32	32	1 to 2 BCLK*8
0008 2BE4h	EXDMAC	Cluster buffer register 1	CLSBR1	32	32	1 to 2 BCLK*8
0008 2BE8h	EXDMAC	Cluster buffer register 2	CLSBR2	32	32	1 to 2 BCLK*8
0008 2BECh	EXDMAC	Cluster buffer register 3	CLSBR3	32	32	1 to 2 BCLK*8
0008 2BF0h	EXDMAC	Cluster buffer register 4	CLSBR4	32	32	1 to 2 BCLK*8
0008 2BF4h	EXDMAC	Cluster buffer register 5	CLSBR5	32	32	1 to 2 BCLK*8
0008 2BF8h	EXDMAC	Cluster buffer register 6	CLSBR6	32	32	1 to 2 BCLK*8
0008 3002h	BSC	CS0 mode register	CS0MOD	16	16	1 to 2 BCLK*8
0008 3004h	BSC	CS0 wait control register 1	CS0WCR1	32	32	1 to 2 BCLK*8
0008 3008h	BSC	CS0 wait control register 2	CS0WCR2	32	32	1 to 2 BCLK*8
0008 3012h	BSC	CS1 mode register	CS1MOD	16	16	1 to 2 BCLK*8
0008 3014h	BSC	CS1 wait control register 1	CS1WCR1	32	32	1 to 2 BCLK*8
0008 3018h	BSC	CS1 wait control register 2	CS1WCR2	32	32	1 to 2 BCLK*8
0008 3022h	BSC	CS2 mode register	CS2MOD	16	16	1 to 2 BCLK*8
0008 3024h	BSC	CS2 wait control register 1	CS2WCR1	32	32	1 to 2 BCLK*8
0008 3028h	BSC	CS2 wait control register 2	CS2WCR2	32	32	1 to 2 BCLK*8
0008 3032h	BSC	CS3 mode register	CS3MOD	16	16	1 to 2 BCLK*8
0008 3034h	BSC	CS3 wait control register 1	CS3WCR1	32	32	1 to 2 BCLK*8
0008 3038h	BSC	CS3 wait control register 2	CS3WCR2	32	32	1 to 2 BCLK*8
0008 3042h	BSC	CS4 mode register	CS4MOD	16	16	1 to 2 BCLK*8
0008 3044h	BSC	CS4 wait control register 1	CS4WCR1	32	32	1 to 2 BCLK*8
0008 3048h	BSC	CS4 wait control register 2	CS4WCR2	32	32	1 to 2 BCLK*8
0008 3052h	BSC	CS5 mode register	CS5MOD	16	16	1 to 2 BCLK*8
0008 3054h	BSC	CS5 wait control register 1	CS5WCR1	32	32	1 to 2 BCLK*8
0008 3058h	BSC	CS5 wait control register 2	CS5WCR2	32	32	1 to 2 BCLK*8
0008 3062h	BSC	CS6 mode register	CS6MOD	16	16	1 to 2 BCLK*8
0008 3064h	BSC	CS6 wait control register 1	CS6WCR1	32	32	1 to 2 BCLK*8
0008 3068h	BSC	CS6 wait control register 2	CS6WCR2	32	32	1 to 2 BCLK*8
0008 3072h	BSC	CS7 mode register	CS7MOD	16	16	1 to 2 BCLK*8
0008 3074h	BSC	CS7 wait control register 1	CS7WCR1	32	32	1 to 2 BCLK*8
0008 3078h	BSC	CS7 wait control register 2	CS7WCR2	32	32	1 to 2 BCLK*8
0008 3802h	BSC	CS0 control register	CS0CR	16	16	1 to 2 BCLK*8

Table 5.1 List of I/O Registers (Address Order) (4 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 380Ah	BSC	CS0 recovery cycle register	CS0REC	16	16	1 to 2 BCLK*8
0008 3812h	BSC	CS1 control register	CS1CR	16	16	1 to 2 BCLK*8
0008 381Ah	BSC	CS1 recovery cycle register	CS1REC	16	16	1 to 2 BCLK*8
0008 3822h	BSC	CS2 control register	CS2CR	16	16	1 to 2 BCLK*8
0008 382Ah	BSC	CS2 recovery cycle register	CS2REC	16	16	1 to 2 BCLK*8
0008 3832h	BSC	CS3 control register	CS3CR	16	16	1 to 2 BCLK*8
0008 383Ah	BSC	CS3 recovery cycle register	CS3REC	16	16	1 to 2 BCLK*8
0008 3842h	BSC	CS4 control register	CS4CR	16	16	1 to 2 BCLK*8
0008 384Ah	BSC	CS4 recovery cycle register	CS4REC	16	16	1 to 2 BCLK*8
0008 3852h	BSC	CS5 control register	CS5CR	16	16	1 to 2 BCLK*8
0008 385Ah	BSC	CS5 recovery cycle register	CS5REC	16	16	1 to 2 BCLK*8
0008 3862h	BSC	CS6 control register	CS6CR	16	16	1 to 2 BCLK*8
0008 386Ah	BSC	CS6 recovery cycle register	CS6REC	16	16	1 to 2 BCLK*8
0008 3872h	BSC	CS7 control register	CS7CR	16	16	1 to 2 BCLK*8
0008 387Ah	BSC	CS7 recovery cycle register	CS7REC	16	16	1 to 2 BCLK*8
0008 3C00h	BSC	SDC control register	SDCCR	8	8	1 to 2 BCLK*8
0008 3C01h	BSC	SDC mode register	SDCMOD	8	8	1 to 2 BCLK*8
0008 3C02h	BSC	SDRAM access mode register	SDAMOD	8	8	1 to 2 BCLK*8
0008 3C10h	BSC	SDRAM self-refresh control register	SDSELF	8	8	1 to 2 BCLK*8
0008 3C14h	BSC	SDRAM refresh control register	SDRFCR	16	16	1 to 2 BCLK*8
0008 3C16h	BSC	SDRAM auto-refresh control register	SDRFEN	8	8	1 to 2 BCLK*8
0008 3C20h	BSC	SDRAM initialization sequence control register	SDICR	8	8	1 to 2 BCLK*8
0008 3C24h	BSC	SDRAM initialization register	SDIR	16	16	1 to 2 BCLK*8
0008 3C40h	BSC	SDRAM address register	SDADR	8	8	1 to 2 BCLK*8
0008 3C44h	BSC	SDRAM timing register	SDTR	32	32	1 to 2 BCLK*8
0008 3C48h	BSC	SDRAM mode register	SDMOD	16	16	1 to 2 BCLK*8
0008 3C50h	BSC	SDRAM status register	SDSR	8	8	1 to 2 BCLK*8
0008 6400h	MPU	Region 0 start page-number register	RSPAGE0	32	32	1 ICLK
0008 6404h	MPU	Region 0 end page-number register	REPAGE0	32	32	1 ICLK
0008 6408h	MPU	Region 1 start page-number register	RSPAGE1	32	32	1 ICLK
0008 640Ch	MPU	Region 1 end page-number register	REPAGE1	32	32	1 ICLK
0008 6410h	MPU	Region 2 start page-number register	RSPAGE2	32	32	1 ICLK
0008 6414h	MPU	Region 2 end page-number register	REPAGE2	32	32	1 ICLK
0008 6418h	MPU	Region 3 start page-number register	RSPAGE3	32	32	1 ICLK
0008 641Ch	MPU	Region 3 end page-number register	REPAGE3	32	32	1 ICLK
0008 6420h	MPU	Region 4 start page-number register	RSPAGE4	32	32	1 ICLK
0008 6424h	MPU	Region 4 end page-number register	REPAGE4	32	32	1 ICLK
0008 6428h	MPU	Region 5 start page-number register	RSPAGE5	32	32	1 ICLK
0008 642Ch	MPU	Region 5 end page-number register	REPAGE5	32	32	1 ICLK
0008 6430h	MPU	Region 6 start page-number register	RSPAGE6	32	32	1 ICLK
0008 6434h	MPU	Region 6 end page-number register	REPAGE6	32	32	1 ICLK
0008 6438h	MPU	Region 7 start page-number register	RSPAGE7	32	32	1 ICLK
0008 643Ch	MPU	Region 7 end page-number register	REPAGE7	32	32	1 ICLK
0008 6500h	MPU	Memory-protection enable register	MPEN	32	32	1 ICLK

Table 5.1 List of I/O Registers (Address Order) (5 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 6504h	MPU	Background access control register	MPBAC	32	32	1 ICLK
0008 6508h	MPU	Memory-protection error status-clearing register	MPECLR	32	32	1 ICLK
0008 650Ch	MPU	Memory-protection error status register	MPESTS	32	32	1 ICLK
0008 6514h	MPU	Data memory-protection error address register	MPDEA	32	32	1 ICLK
0008 6520h	MPU	Region search address register	MPSA	32	32	1 ICLK
0008 6524h	MPU	Region search operation register	MPOPS	16	16	1 ICLK
0008 6526h	MPU	Region invalidation operation register	MPOPI	16	16	1 ICLK
0008 6528h	MPU	Instruction-hit region register	MHITI	32	32	1 ICLK
0008 652Ch	MPU	Data-hit region register	MHITD	32	32	1 ICLK
0008 7010h	ICU	Interrupt request register 016	IR016	8	8	2 ICLK
0008 7015h	ICU	Interrupt request register 021	IR021	8	8	2 ICLK
0008 7017h	ICU	Interrupt request register 023	IR023	8	8	2 ICLK
0008 701Bh	ICU	Interrupt request register 027	IR027	8	8	2 ICLK
0008 701Ch	ICU	Interrupt request register 028	IR028	8	8	2 ICLK
0008 701Dh	ICU	Interrupt request register 029	IR029	8	8	2 ICLK
0008 701Eh	ICU	Interrupt request register 030	IR030	8	8	2 ICLK
0008 701Fh	ICU	Interrupt request register 031	IR031	8	8	2 ICLK
0008 7020h	ICU	Interrupt request register 032	IR032	8	8	2 ICLK
0008 7024h	ICU	Interrupt request register 036	IR036	8	8	2 ICLK
0008 7025h	ICU	Interrupt request register 037	IR037	8	8	2 ICLK
0008 7026h	ICU	Interrupt request register 038	IR038	8	8	2 ICLK
0008 7028h	ICU	Interrupt request register 040	IR040	8	8	2 ICLK
0008 7029h	ICU	Interrupt request register 041	IR041	8	8	2 ICLK
0008 702Ah	ICU	Interrupt request register 042	IR042	8	8	2 ICLK
0008 702Ch	ICU	Interrupt request register 044	IR044	8	8	2 ICLK
0008 702Dh	ICU	Interrupt request register 045	IR045	8	8	2 ICLK
0008 702Eh	ICU	Interrupt request register 046	IR046	8	8	2 ICLK
0008 702Fh	ICU	Interrupt request register 047	IR047	8	8	2 ICLK
0008 7030h	ICU	Interrupt request register 048	IR048	8	8	2 ICLK
0008 7031h	ICU	Interrupt request register 049	IR049	8	8	2 ICLK
0008 7032h	ICU	Interrupt request register 050	IR050	8	8	2 ICLK
0008 7033h	ICU	Interrupt request register 051	IR051	8	8	2 ICLK
0008 7038h	ICU	Interrupt request register 056	IR056	8	8	2 ICLK
0008 7039h	ICU	Interrupt request register 057	IR057	8	8	2 ICLK
0008 703Ah	ICU	Interrupt request register 058	IR058	8	8	2 ICLK
0008 703Bh	ICU	Interrupt request register 059	IR059	8	8	2 ICLK
0008 703Ch	ICU	Interrupt request register 060	IR060	8	8	2 ICLK
0008 703Eh	ICU	Interrupt request register 062	IR062	8	8	2 ICLK
0008 703Fh	ICU	Interrupt request register 063	IR063	8	8	2 ICLK
0008 7040h	ICU	Interrupt request register 064	IR064	8	8	2 ICLK
0008 7041h	ICU	Interrupt request register 065	IR065	8	8	2 ICLK
0008 7042h	ICU	Interrupt request register 066	IR066	8	8	2 ICLK
0008 7043h	ICU	Interrupt request register 067	IR067	8	8	2 ICLK

Table 5.1 List of I/O Registers (Address Order) (6 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 7044h	ICU	Interrupt request register 068	IR068	8	8	2 ICLK
0008 7045h	ICU	Interrupt request register 069	IR069	8	8	2 ICLK
0008 7046h	ICU	Interrupt request register 070	IR070	8	8	2 ICLK
0008 7047h	ICU	Interrupt request register 071	IR071	8	8	2 ICLK
0008 7048h	ICU	Interrupt request register 072	IR072	8	8	2 ICLK
0008 7049h	ICU	Interrupt request register 073	IR073	8	8	2 ICLK
0008 704Ah	ICU	Interrupt request register 074	IR074	8	8	2 ICLK
0008 704Bh	ICU	Interrupt request register 075	IR075	8	8	2 ICLK
0008 704Ch	ICU	Interrupt request register 076	IR076	8	8	2 ICLK
0008 704Dh	ICU	Interrupt request register 077	IR077	8	8	2 ICLK
0008 704Eh	ICU	Interrupt request register 078	IR078	8	8	2 ICLK
0008 704Fh	ICU	Interrupt request register 079	IR079	8	8	2 ICLK
0008 705Ah	ICU	Interrupt request register 090	IR090	8	8	2 ICLK
0008 705Bh	ICU	Interrupt request register 091	IR091	8	8	2 ICLK
0008 705Ch	ICU	Interrupt request register 092	IR092	8	8	2 ICLK
0008 7060h	ICU	Interrupt request register 096	IR096	8	8	2 ICLK
0008 7062h	ICU	Interrupt request register 098	IR098	8	8	2 ICLK
0008 7063h	ICU	Interrupt request register 099	IR099	8	8	2 ICLK
0008 7066h	ICU	Interrupt request register 102	IR102	8	8	2 ICLK
0008 7072h	ICU	Interrupt request register 114	IR114	8	8	2 ICLK
0008 7073h	ICU	Interrupt request register 115	IR115	8	8	2 ICLK
0008 7074h	ICU	Interrupt request register 116	IR116	8	8	2 ICLK
0008 7075h	ICU	Interrupt request register 117	IR117	8	8	2 ICLK
0008 7076h	ICU	Interrupt request register 118	IR118	8	8	2 ICLK
0008 7077h	ICU	Interrupt request register 119	IR119	8	8	2 ICLK
0008 7078h	ICU	Interrupt request register 120	IR120	8	8	2 ICLK
0008 7079h	ICU	Interrupt request register 121	IR121	8	8	2 ICLK
0008 707Ah	ICU	Interrupt request register 122	IR122	8	8	2 ICLK
0008 707Bh	ICU	Interrupt request register 123	IR123	8	8	2 ICLK
0008 707Ch	ICU	Interrupt request register 124	IR124	8	8	2 ICLK
0008 707Dh	ICU	Interrupt request register 125	IR125	8	8	2 ICLK
0008 707Eh	ICU	Interrupt request register 126	IR126	8	8	2 ICLK
0008 707Fh	ICU	Interrupt request register 127	IR127	8	8	2 ICLK
0008 7080h	ICU	Interrupt request register 128	IR128	8	8	2 ICLK
0008 7081h	ICU	Interrupt request register 129	IR129	8	8	2 ICLK
0008 7082h	ICU	Interrupt request register 130	IR130	8	8	2 ICLK
0008 7083h	ICU	Interrupt request register 131	IR131	8	8	2 ICLK
0008 7084h	ICU	Interrupt request register 132	IR132	8	8	2 ICLK
0008 7085h	ICU	Interrupt request register 133	IR133	8	8	2 ICLK
0008 7086h	ICU	Interrupt request register 134	IR134	8	8	2 ICLK
0008 7087h	ICU	Interrupt request register 135	IR135	8	8	2 ICLK
0008 7088h	ICU	Interrupt request register 136	IR136	8	8	2 ICLK
0008 7089h	ICU	Interrupt request register 137	IR137	8	8	2 ICLK
0008 708Ah	ICU	Interrupt request register 138	IR138	8	8	2 ICLK
0008 708Bh	ICU	Interrupt request register 139	IR139	8	8	2 ICLK

Table 5.1 List of I/O Registers (Address Order) (7 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 708Ch	ICU	Interrupt request register 140	IR140	8	8	2 ICLK
0008 708Dh	ICU	Interrupt request register 141	IR141	8	8	2 ICLK
0008 708Eh	ICU	Interrupt request register 142	IR142	8	8	2 ICLK
0008 708Fh	ICU	Interrupt request register 143	IR143	8	8	2 ICLK
0008 7090h	ICU	Interrupt request register 144	IR144	8	8	2 ICLK
0008 7091h	ICU	Interrupt request register 145	IR145	8	8	2 ICLK
0008 7092h	ICU	Interrupt request register 146	IR146	8	8	2 ICLK
0008 7093h	ICU	Interrupt request register 147	IR147	8	8	2 ICLK
0008 7094h	ICU	Interrupt request register 148	IR148	8	8	2 ICLK
0008 7095h	ICU	Interrupt request register 149	IR149	8	8	2 ICLK
0008 7096h	ICU	Interrupt request register 150	IR150	8	8	2 ICLK
0008 7097h	ICU	Interrupt request register 151	IR151	8	8	2 ICLK
0008 7098h	ICU	Interrupt request register 152	IR152	8	8	2 ICLK
0008 7099h	ICU	Interrupt request register 153	IR153	8	8	2 ICLK
0008 709Ah	ICU	Interrupt request register 154	IR154	8	8	2 ICLK
0008 709Bh	ICU	Interrupt request register 155	IR155	8	8	2 ICLK
0008 709Ch	ICU	Interrupt request register 156	IR156	8	8	2 ICLK
0008 709Dh	ICU	Interrupt request register 157	IR157	8	8	2 ICLK
0008 709Eh	ICU	Interrupt request register 158	IR158	8	8	2 ICLK
0008 709Fh	ICU	Interrupt request register 159	IR159	8	8	2 ICLK
0008 70A0h	ICU	Interrupt request register 160	IR160	8	8	2 ICLK
0008 70A1h	ICU	Interrupt request register 161	IR161	8	8	2 ICLK
0008 70A2h	ICU	Interrupt request register 162	IR162	8	8	2 ICLK
0008 70A3h	ICU	Interrupt request register 163	IR163	8	8	2 ICLK
0008 70A4h	ICU	Interrupt request register 164	IR164	8	8	2 ICLK
0008 70A5h	ICU	Interrupt request register 165	IR165	8	8	2 ICLK
0008 70A6h	ICU	Interrupt request register 166	IR166	8	8	2 ICLK
0008 70A7h	ICU	Interrupt request register 167	IR167	8	8	2 ICLK
0008 70A8h	ICU	Interrupt request register 168	IR168	8	8	2 ICLK
0008 70A9h	ICU	Interrupt request register 169	IR169	8	8	2 ICLK
0008 70AAh	ICU	Interrupt request register 170	IR170	8	8	2 ICLK
0008 70ABh	ICU	Interrupt request register 171	IR171	8	8	2 ICLK
0008 70ACh	ICU	Interrupt request register 172	IR172	8	8	2 ICLK
0008 70ADh	ICU	Interrupt request register 173	IR173	8	8	2 ICLK
0008 70AEh	ICU	Interrupt request register 174	IR174	8	8	2 ICLK
0008 70AFh	ICU	Interrupt request register 175	IR175	8	8	2 ICLK
0008 70B0h	ICU	Interrupt request register 176	IR176	8	8	2 ICLK
0008 70B1h	ICU	Interrupt request register 177	IR177	8	8	2 ICLK
0008 70B2h	ICU	Interrupt request register 178	IR178	8	8	2 ICLK
0008 70B3h	ICU	Interrupt request register 179	IR179	8	8	2 ICLK
0008 70B4h	ICU	Interrupt request register 180	IR180	8	8	2 ICLK
0008 70B5h	ICU	Interrupt request register 181	IR181	8	8	2 ICLK
0008 70B6h	ICU	Interrupt request register 182	IR182	8	8	2 ICLK
0008 70B7h	ICU	Interrupt request register 183	IR183	8	8	2 ICLK
0008 70B8h	ICU	Interrupt request register 184	IR184	8	8	2 ICLK

Table 5.1 List of I/O Registers (Address Order) (8 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 70B9h	ICU	Interrupt request register 185	IR185	8	8	2 ICLK
0008 70C6h	ICU	Interrupt request register 198	IR198	8	8	2 ICLK
0008 70C7h	ICU	Interrupt request register 199	IR199	8	8	2 ICLK
0008 70C8h	ICU	Interrupt request register 200	IR200	8	8	2 ICLK
0008 70C9h	ICU	Interrupt request register 201	IR201	8	8	2 ICLK
0008 70CAh	ICU	Interrupt request register 202	IR202	8	8	2 ICLK
0008 70CBh	ICU	Interrupt request register 203	IR203	8	8	2 ICLK
0008 70D6h	ICU	Interrupt request register 214	IR214	8	8	2 ICLK
0008 70D7h	ICU	Interrupt request register 215	IR215	8	8	2 ICLK
0008 70D8h	ICU	Interrupt request register 216	IR216	8	8	2 ICLK
0008 70D9h	ICU	Interrupt request register 217	IR217	8	8	2 ICLK
0008 70DAh	ICU	Interrupt request register 218	IR218	8	8	2 ICLK
0008 70DBh	ICU	Interrupt request register 219	IR219	8	8	2 ICLK
0008 70DCh	ICU	Interrupt request register 220	IR220	8	8	2 ICLK
0008 70DDh	ICU	Interrupt request register 221	IR221	8	8	2 ICLK
0008 70DEh	ICU	Interrupt request register 222	IR222	8	8	2 ICLK
0008 70DFh	ICU	Interrupt request register 223	IR223	8	8	2 ICLK
0008 70E0h	ICU	Interrupt request register 224	IR224	8	8	2 ICLK
0008 70E1h	ICU	Interrupt request register 225	IR225	8	8	2 ICLK
0008 70E2h	ICU	Interrupt request register 226	IR226	8	8	2 ICLK
0008 70E3h	ICU	Interrupt request register 227	IR227	8	8	2 ICLK
0008 70E4h	ICU	Interrupt request register 228	IR228	8	8	2 ICLK
0008 70E5h	ICU	Interrupt request register 229	IR229	8	8	2 ICLK
0008 70EAh	ICU	Interrupt request register 234	IR234	8	8	2 ICLK
0008 70EBh	ICU	Interrupt request register 235	IR235	8	8	2 ICLK
0008 70ECh	ICU	Interrupt request register 236	IR236	8	8	2 ICLK
0008 70EDh	ICU	Interrupt request register 237	IR237	8	8	2 ICLK
0008 70EEh	ICU	Interrupt request register 238	IR238	8	8	2 ICLK
0008 70EFh	ICU	Interrupt request register 239	IR239	8	8	2 ICLK
0008 70F0h	ICU	Interrupt request register 240	IR240	8	8	2 ICLK
0008 70F1h	ICU	Interrupt request register 241	IR241	8	8	2 ICLK
0008 70F6h	ICU	Interrupt request register 246	IR246	8	8	2 ICLK
0008 70F7h	ICU	Interrupt request register 247	IR247	8	8	2 ICLK
0008 70F8h	ICU	Interrupt request register 248	IR248	8	8	2 ICLK
0008 70F9h	ICU	Interrupt request register 249	IR249	8	8	2 ICLK
0008 70FAh	ICU	Interrupt request register 250	IR250	8	8	2 ICLK
0008 70FBh	ICU	Interrupt request register 251	IR251	8	8	2 ICLK
0008 70FCh	ICU	Interrupt request register 252	IR252	8	8	2 ICLK
0008 70FDh	ICU	Interrupt request register 253	IR253	8	8	2 ICLK
0008 711Bh	ICU	DTC activation enable register 027	DTCER027	8	8	2 ICLK
0008 711Ch	ICU	DTC activation enable register 028	DTCER028	8	8	2 ICLK
0008 711Dh	ICU	DTC activation enable register 029	DTCER029	8	8	2 ICLK
0008 711Eh	ICU	DTC activation enable register 030	DTCER030	8	8	2 ICLK
0008 711Fh	ICU	DTC activation enable register 031	DTCER031	8	8	2 ICLK
0008 7124h	ICU	DTC activation enable register 036	DTCER036	8	8	2 ICLK

Table 5.1 List of I/O Registers (Address Order) (9 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 7125h	ICU	DTC activation enable register 037	DTCER037	8	8	2 ICLK
0008 7128h	ICU	DTC activation enable register 040	DTCER040	8	8	2 ICLK
0008 7129h	ICU	DTC activation enable register 041	DTCER041	8	8	2 ICLK
0008 712Dh	ICU	DTC activation enable register 045	DTCER045	8	8	2 ICLK
0008 712Eh	ICU	DTC activation enable register 046	DTCER046	8	8	2 ICLK
0008 7131h	ICU	DTC activation enable register 049	DTCER049	8	8	2 ICLK
0008 7132h	ICU	DTC activation enable register 050	DTCER050	8	8	2 ICLK
0008 7140h	ICU	DTC activation enable register 064	DTCER064	8	8	2 ICLK
0008 7141h	ICU	DTC activation enable register 065	DTCER065	8	8	2 ICLK
0008 7142h	ICU	DTC activation enable register 066	DTCER066	8	8	2 ICLK
0008 7143h	ICU	DTC activation enable register 067	DTCER067	8	8	2 ICLK
0008 7144h	ICU	DTC activation enable register 068	DTCER068	8	8	2 ICLK
0008 7145h	ICU	DTC activation enable register 069	DTCER069	8	8	2 ICLK
0008 7146h	ICU	DTC activation enable register 070	DTCER070	8	8	2 ICLK
0008 7147h	ICU	DTC activation enable register 071	DTCER071	8	8	2 ICLK
0008 7148h	ICU	DTC activation enable register 072	DTCER072	8	8	2 ICLK
0008 7149h	ICU	DTC activation enable register 073	DTCER073	8	8	2 ICLK
0008 714Ah	ICU	DTC activation enable register 074	DTCER074	8	8	2 ICLK
0008 714Bh	ICU	DTC activation enable register 075	DTCER075	8	8	2 ICLK
0008 714Ch	ICU	DTC activation enable register 076	DTCER076	8	8	2 ICLK
0008 714Dh	ICU	DTC activation enable register 077	DTCER077	8	8	2 ICLK
0008 714Eh	ICU	DTC activation enable register 078	DTCER078	8	8	2 ICLK
0008 714Fh	ICU	DTC activation enable register 079	DTCER079	8	8	2 ICLK
0008 7162h	ICU	DTC activation enable register 098	DTCER098	8	8	2 ICLK
0008 7163h	ICU	DTC activation enable register 099	DTCER099	8	8	2 ICLK
0008 7166h	ICU	DTC activation enable register 102	DTCER102	8	8	2 ICLK
0008 7172h	ICU	DTC activation enable register 114	DTCER114	8	8	2 ICLK
0008 7173h	ICU	DTC activation enable register 115	DTCER115	8	8	2 ICLK
0008 7174h	ICU	DTC activation enable register 116	DTCER116	8	8	2 ICLK
0008 7175h	ICU	DTC activation enable register 117	DTCER117	8	8	2 ICLK
0008 7179h	ICU	DTC activation enable register 121	DTCER121	8	8	2 ICLK
0008 717Ah	ICU	DTC activation enable register 122	DTCER122	8	8	2 ICLK
0008 717Dh	ICU	DTC activation enable register 125	DTCER125	8	8	2 ICLK
0008 717Eh	ICU	DTC activation enable register 126	DTCER126	8	8	2 ICLK
0008 7181h	ICU	DTC activation enable register 129	DTCER129	8	8	2 ICLK
0008 7182h	ICU	DTC activation enable register 130	DTCER130	8	8	2 ICLK
0008 7183h	ICU	DTC activation enable register 131	DTCER131	8	8	2 ICLK
0008 7184h	ICU	DTC activation enable register 132	DTCER132	8	8	2 ICLK
0008 7186h	ICU	DTC activation enable register 134	DTCER134	8	8	2 ICLK
0008 7187h	ICU	DTC activation enable register 135	DTCER135	8	8	2 ICLK
0008 7188h	ICU	DTC activation enable register 136	DTCER136	8	8	2 ICLK
0008 7189h	ICU	DTC activation enable register 137	DTCER137	8	8	2 ICLK
0008 718Ah	ICU	DTC activation enable register 138	DTCER138	8	8	2 ICLK
0008 718Bh	ICU	DTC activation enable register 139	DTCER139	8	8	2 ICLK
0008 718Ch	ICU	DTC activation enable register 140	DTCER140	8	8	2 ICLK

Table 5.1 List of I/O Registers (Address Order) (10 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 718Dh	ICU	DTC activation enable register 141	DTCER141	8	8	2 ICLK
0008 718Eh	ICU	DTC activation enable register 142	DTCER142	8	8	2 ICLK
0008 718Fh	ICU	DTC activation enable register 143	DTCER143	8	8	2 ICLK
0008 7190h	ICU	DTC activation enable register 144	DTCER144	8	8	2 ICLK
0008 7191h	ICU	DTC activation enable register 145	DTCER145	8	8	2 ICLK
0008 7195h	ICU	DTC activation enable register 149	DTCER149	8	8	2 ICLK
0008 7196h	ICU	DTC activation enable register 150	DTCER150	8	8	2 ICLK
0008 7199h	ICU	DTC activation enable register 153	DTCER153	8	8	2 ICLK
0008 719Ah	ICU	DTC activation enable register 154	DTCER154	8	8	2 ICLK
0008 719Dh	ICU	DTC activation enable register 157	DTCER157	8	8	2 ICLK
0008 719Eh	ICU	DTC activation enable register 158	DTCER158	8	8	2 ICLK
0008 719Fh	ICU	DTC activation enable register 159	DTCER159	8	8	2 ICLK
0008 71A0h	ICU	DTC activation enable register 160	DTCER160	8	8	2 ICLK
0008 71A2h	ICU	DTC activation enable register 162	DTCER162	8	8	2 ICLK
0008 71A3h	ICU	DTC activation enable register 163	DTCER163	8	8	2 ICLK
0008 71A4h	ICU	DTC activation enable register 164	DTCER164	8	8	2 ICLK
0008 71A5h	ICU	DTC activation enable register 165	DTCER165	8	8	2 ICLK
0008 71A6h	ICU	DTC activation enable register 166	DTCER166	8	8	2 ICLK
0008 71A7h	ICU	DTC activation enable register 167	DTCER167	8	8	2 ICLK
0008 71A8h	ICU	DTC activation enable register 168	DTCER168	8	8	2 ICLK
0008 71A9h	ICU	DTC activation enable register 169	DTCER169	8	8	2 ICLK
0008 71AEh	ICU	DTC activation enable register 174	DTCER174	8	8	2 ICLK
0008 71AFh	ICU	DTC activation enable register 175	DTCER175	8	8	2 ICLK
0008 71B1h	ICU	DTC activation enable register 177	DTCER177	8	8	2 ICLK
0008 71B2h	ICU	DTC activation enable register 178	DTCER178	8	8	2 ICLK
0008 71B4h	ICU	DTC activation enable register 180	DTCER180	8	8	2 ICLK
0008 71B5h	ICU	DTC activation enable register 181	DTCER181	8	8	2 ICLK
0008 71B7h	ICU	DTC activation enable register 183	DTCER183	8	8	2 ICLK
0008 71B8h	ICU	DTC activation enable register 184	DTCER184	8	8	2 ICLK
0008 71C6h	ICU	DTC activation enable register 198	DTCER198	8	8	2 ICLK
0008 71C7h	ICU	DTC activation enable register 199	DTCER199	8	8	2 ICLK
0008 71C8h	ICU	DTC activation enable register 200	DTCER200	8	8	2 ICLK
0008 71C9h	ICU	DTC activation enable register 201	DTCER201	8	8	2 ICLK
0008 71CAh	ICU	DTC activation enable register 202	DTCER202	8	8	2 ICLK
0008 71CBh	ICU	DTC activation enable register 203	DTCER203	8	8	2 ICLK
0008 71D7h	ICU	DTC activation enable register 215	DTCER215	8	8	2 ICLK
0008 71D8h	ICU	DTC activation enable register 216	DTCER216	8	8	2 ICLK
0008 71DBh	ICU	DTC activation enable register 219	DTCER219	8	8	2 ICLK
0008 71DCh	ICU	DTC activation enable register 220	DTCER220	8	8	2 ICLK
0008 71DFh	ICU	DTC activation enable register 223	DTCER223	8	8	2 ICLK
0008 71E0h	ICU	DTC activation enable register 224	DTCER224	8	8	2 ICLK
0008 71E3h	ICU	DTC activation enable register 227	DTCER227	8	8	2 ICLK
0008 71E4h	ICU	DTC activation enable register 228	DTCER228	8	8	2 ICLK
0008 71EBh	ICU	DTC activation enable register 235	DTCER235	8	8	2 ICLK
0008 71ECh	ICU	DTC activation enable register 236	DTCER236	8	8	2 ICLK

Table 5.1 List of I/O Registers (Address Order) (11 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 71EFh	ICU	DTC activation enable register 239	DT CER239	8	8	2 ICLK
0008 71F0h	ICU	DTC activation enable register 240	DT CER240	8	8	2 ICLK
0008 71F7h	ICU	DTC activation enable register 247	DT CER247	8	8	2 ICLK
0008 71F8h	ICU	DTC activation enable register 248	DT CER248	8	8	2 ICLK
0008 71FBh	ICU	DTC activation enable register 251	DT CER251	8	8	2 ICLK
0008 71FCh	ICU	DTC activation enable register 252	DT CER252	8	8	2 ICLK
0008 7202h	ICU	Interrupt request enable register 02	IER02	8	8	2 ICLK
0008 7203h	ICU	Interrupt request enable register 03	IER03	8	8	2 ICLK
0008 7204h	ICU	Interrupt request enable register 04	IER04	8	8	2 ICLK
0008 7205h	ICU	Interrupt request enable register 05	IER05	8	8	2 ICLK
0008 7206h	ICU	Interrupt request enable register 06	IER06	8	8	2 ICLK
0008 7207h	ICU	Interrupt request enable register 07	IER07	8	8	2 ICLK
0008 7208h	ICU	Interrupt request enable register 08	IER08	8	8	2 ICLK
0008 7209h	ICU	Interrupt request enable register 09	IER09	8	8	2 ICLK
0008 720Bh	ICU	Interrupt request enable register 0B	IER0B	8	8	2 ICLK
0008 720Ch	ICU	Interrupt request enable register 0C	IER0C	8	8	2 ICLK
0008 720Eh	ICU	Interrupt request enable register 0E	IER0E	8	8	2 ICLK
0008 720Fh	ICU	Interrupt request enable register 0F	IER0F	8	8	2 ICLK
0008 7210h	ICU	Interrupt request enable register 10	IER10	8	8	2 ICLK
0008 7211h	ICU	Interrupt request enable register 11	IER11	8	8	2 ICLK
0008 7212h	ICU	Interrupt request enable register 12	IER12	8	8	2 ICLK
0008 7213h	ICU	Interrupt request enable register 13	IER13	8	8	2 ICLK
0008 7214h	ICU	Interrupt request enable register 14	IER14	8	8	2 ICLK
0008 7215h	ICU	Interrupt request enable register 15	IER15	8	8	2 ICLK
0008 7216h	ICU	Interrupt request enable register 16	IER16	8	8	2 ICLK
0008 7217h	ICU	Interrupt request enable register 17	IER17	8	8	2 ICLK
0008 7218h	ICU	Interrupt request enable register 18	IER18	8	8	2 ICLK
0008 7219h	ICU	Interrupt request enable register 19	IER19	8	8	2 ICLK
0008 721Ah	ICU	Interrupt request enable register 1A	IER1A	8	8	2 ICLK
0008 721Bh	ICU	Interrupt request enable register 1B	IER1B	8	8	2 ICLK
0008 721Ch	ICU	Interrupt request enable register 1C	IER1C	8	8	2 ICLK
0008 721Dh	ICU	Interrupt request enable register 1D	IER1D	8	8	2 ICLK
0008 721Eh	ICU	Interrupt request enable register 1E	IER1E	8	8	2 ICLK
0008 721Fh	ICU	Interrupt request enable register 1F	IER1F	8	8	2 ICLK
0008 72E0h	ICU	Software interrupt activation register	SWINTR	8	8	2 ICLK
0008 72F0h	ICU	Fast interrupt set register	FIR	16	16	2 ICLK
0008 7300h	ICU	Interrupt source priority register 00	IPR00	8	8	2 ICLK
0008 7301h	ICU	Interrupt source priority register 01	IPR01	8	8	2 ICLK
0008 7302h	ICU	Interrupt source priority register 02	IPR02	8	8	2 ICLK
0008 7303h	ICU	Interrupt source priority register 03	IPR03	8	8	2 ICLK
0008 7304h	ICU	Interrupt source priority register 04	IPR04	8	8	2 ICLK
0008 7305h	ICU	Interrupt source priority register 05	IPR05	8	8	2 ICLK
0008 7306h	ICU	Interrupt source priority register 06	IPR06	8	8	2 ICLK
0008 7307h	ICU	Interrupt source priority register 07	IPR07	8	8	2 ICLK
0008 7308h	ICU	Interrupt source priority register 08	IPR08	8	8	2 ICLK

Table 5.1 List of I/O Registers (Address Order) (12 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 730Ch	ICU	Interrupt source priority register 0C	IPR0C	8	8	2 ICLK
0008 730Dh	ICU	Interrupt source priority register 0D	IPR0D	8	8	2 ICLK
0008 730Eh	ICU	Interrupt source priority register 0E	IPR0E	8	8	2 ICLK
0008 7310h	ICU	Interrupt source priority register 10	IPR10	8	8	2 ICLK
0008 7311h	ICU	Interrupt source priority register 11	IPR11	8	8	2 ICLK
0008 7312h	ICU	Interrupt source priority register 12	IPR12	8	8	2 ICLK
0008 7314h	ICU	Interrupt source priority register 14	IPR14	8	8	2 ICLK
0008 7315h	ICU	Interrupt source priority register 15	IPR15	8	8	2 ICLK
0008 7318h	ICU	Interrupt source priority register 18	IPR18	8	8	2 ICLK
0008 731Eh	ICU	Interrupt source priority register 1E	IPR1E	8	8	2 ICLK
0008 731Fh	ICU	Interrupt source priority register 1F	IPR1F	8	8	2 ICLK
0008 7320h	ICU	Interrupt source priority register 20	IPR20	8	8	2 ICLK
0008 7321h	ICU	Interrupt source priority register 21	IPR21	8	8	2 ICLK
0008 7322h	ICU	Interrupt source priority register 22	IPR22	8	8	2 ICLK
0008 7323h	ICU	Interrupt source priority register 23	IPR23	8	8	2 ICLK
0008 7324h	ICU	Interrupt source priority register 24	IPR24	8	8	2 ICLK
0008 7325h	ICU	Interrupt source priority register 25	IPR25	8	8	2 ICLK
0008 7326h	ICU	Interrupt source priority register 26	IPR26	8	8	2 ICLK
0008 7327h	ICU	Interrupt source priority register 27	IPR27	8	8	2 ICLK
0008 7328h	ICU	Interrupt source priority register 28	IPR28	8	8	2 ICLK
0008 7329h	ICU	Interrupt source priority register 29	IPR29	8	8	2 ICLK
0008 732Ah	ICU	Interrupt source priority register 2A	IPR2A	8	8	2 ICLK
0008 732Bh	ICU	Interrupt source priority register 2B	IPR2B	8	8	2 ICLK
0008 732Ch	ICU	Interrupt source priority register 2C	IPR2C	8	8	2 ICLK
0008 732Dh	ICU	Interrupt source priority register 2D	IPR2D	8	8	2 ICLK
0008 732Eh	ICU	Interrupt source priority register 2E	IPR2E	8	8	2 ICLK
0008 732Fh	ICU	Interrupt source priority register 2F	IPR2F	8	8	2 ICLK
0008 733Ah	ICU	Interrupt source priority register 3A	IPR3A	8	8	2 ICLK
0008 733Bh	ICU	Interrupt source priority register 3B	IPR3B	8	8	2 ICLK
0008 733Ch	ICU	Interrupt source priority register 3C	IPR3C	8	8	2 ICLK
0008 7340h	ICU	Interrupt source priority register 40	IPR40	8	8	2 ICLK
0008 7344h	ICU	Interrupt source priority register 44	IPR44	8	8	2 ICLK
0008 7345h	ICU	Interrupt source priority register 45	IPR45	8	8	2 ICLK
0008 7348h	ICU	Interrupt source priority register 48	IPR48	8	8	2 ICLK
0008 7351h	ICU	Interrupt source priority register 51	IPR51	8	8	2 ICLK
0008 7352h	ICU	Interrupt source priority register 52	IPR52	8	8	2 ICLK
0008 7353h	ICU	Interrupt source priority register 53	IPR53	8	8	2 ICLK
0008 7354h	ICU	Interrupt source priority register 54	IPR54	8	8	2 ICLK
0008 7355h	ICU	Interrupt source priority register 55	IPR55	8	8	2 ICLK
0008 7356h	ICU	Interrupt source priority register 56	IPR56	8	8	2 ICLK
0008 7357h	ICU	Interrupt source priority register 57	IPR57	8	8	2 ICLK
0008 7358h	ICU	Interrupt source priority register 58	IPR58	8	8	2 ICLK
0008 7359h	ICU	Interrupt source priority register 59	IPR59	8	8	2 ICLK
0008 735Ah	ICU	Interrupt source priority register 5A	IPR5A	8	8	2 ICLK
0008 735Bh	ICU	Interrupt source priority register 5B	IPR5B	8	8	2 ICLK

Table 5.1 List of I/O Registers (Address Order) (13 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 735Ch	ICU	Interrupt source priority register 5C	IPR5C	8	8	2 ICLK
0008 735Dh	ICU	Interrupt source priority register 5D	IPR5D	8	8	2 ICLK
0008 735Eh	ICU	Interrupt source priority register 5E	IPR5E	8	8	2 ICLK
0008 735Fh	ICU	Interrupt source priority register 5F	IPR5F	8	8	2 ICLK
0008 7360h	ICU	Interrupt source priority register 60	IPR60	8	8	2 ICLK
0008 7361h	ICU	Interrupt source priority register 61	IPR61	8	8	2 ICLK
0008 7362h	ICU	Interrupt source priority register 62	IPR62	8	8	2 ICLK
0008 7363h	ICU	Interrupt source priority register 63	IPR63	8	8	2 ICLK
0008 7364h	ICU	Interrupt source priority register 64	IPR64	8	8	2 ICLK
0008 7365h	ICU	Interrupt source priority register 65	IPR65	8	8	2 ICLK
0008 7366h	ICU	Interrupt source priority register 66	IPR66	8	8	2 ICLK
0008 7367h	ICU	Interrupt source priority register 67	IPR67	8	8	2 ICLK
0008 7368h	ICU	Interrupt source priority register 68	IPR68	8	8	2 ICLK
0008 7369h	ICU	Interrupt source priority register 69	IPR69	8	8	2 ICLK
0008 736Ah	ICU	Interrupt source priority register 6A	IPR6A	8	8	2 ICLK
0008 736Bh	ICU	Interrupt source priority register 6B	IPR6B	8	8	2 ICLK
0008 7370h	ICU	Interrupt source priority register 70	IPR70	8	8	2 ICLK
0008 7371h	ICU	Interrupt source priority register 71	IPR71	8	8	2 ICLK
0008 7372h	ICU	Interrupt source priority register 72	IPR72	8	8	2 ICLK
0008 7373h	ICU	Interrupt source priority register 73	IPR73	8	8	2 ICLK
0008 7374h	ICU	Interrupt source priority register 74	IPR74	8	8	2 ICLK
0008 7375h	ICU	Interrupt source priority register 75	IPR75	8	8	2 ICLK
0008 7380h	ICU	Interrupt source priority register 80	IPR80	8	8	2 ICLK
0008 7381h	ICU	Interrupt source priority register 81	IPR81	8	8	2 ICLK
0008 7382h	ICU	Interrupt source priority register 82	IPR82	8	8	2 ICLK
0008 7383h	ICU	Interrupt source priority register 83	IPR83	8	8	2 ICLK
0008 7385h	ICU	Interrupt source priority register 85	IPR85	8	8	2 ICLK
0008 7386h	ICU	Interrupt source priority register 86	IPR86	8	8	2 ICLK
0008 7388h	ICU	Interrupt source priority register 88	IPR88	8	8	2 ICLK
0008 7389h	ICU	Interrupt source priority register 89	IPR89	8	8	2 ICLK
0008 738Ah	ICU	Interrupt source priority register 8A	IPR8A	8	8	2 ICLK
0008 738Bh	ICU	Interrupt source priority register 8B	IPR8B	8	8	2 ICLK
0008 738Ch	ICU	Interrupt source priority register 8C	IPR8C	8	8	2 ICLK
0008 738Dh	ICU	Interrupt source priority register 8D	IPR8D	8	8	2 ICLK
0008 738Eh	ICU	Interrupt source priority register 8E	IPR8E	8	8	2 ICLK
0008 738Fh	ICU	Interrupt source priority register 8F	IPR8F	8	8	2 ICLK
0008 7400h	ICU	DMACA activation source select register 0	DMRSR0	8	8	2 ICLK
0008 7404h	ICU	DMACA activation source select register 1	DMRSR1	8	8	2 ICLK
0008 7408h	ICU	DMACA activation source select register 2	DMRSR2	8	8	2 ICLK
0008 740Ch	ICU	DMACA activation source select register 3	DMRSR3	8	8	2 ICLK
0008 7500h	ICU	IRQ control register 0	IRQCR0	8	8	2 ICLK
0008 7501h	ICU	IRQ control register 1	IRQCR1	8	8	2 ICLK
0008 7502h	ICU	IRQ control register 2	IRQCR2	8	8	2 ICLK
0008 7503h	ICU	IRQ control register 3	IRQCR3	8	8	2 ICLK
0008 7504h	ICU	IRQ control register 4	IRQCR4	8	8	2 ICLK

Table 5.1 List of I/O Registers (Address Order) (14 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 7505h	ICU	IRQ control register 5	IRQCR5	8	8	2 ICLK
0008 7506h	ICU	IRQ control register 6	IRQCR6	8	8	2 ICLK
0008 7507h	ICU	IRQ control register 7	IRQCR7	8	8	2 ICLK
0008 7508h	ICU	IRQ control register 8	IRQCR8	8	8	2 ICLK
0008 7509h	ICU	IRQ control register 9	IRQCR9	8	8	2 ICLK
0008 750Ah	ICU	IRQ control register 10	IRQCR10	8	8	2 ICLK
0008 750Bh	ICU	IRQ control register 11	IRQCR11	8	8	2 ICLK
0008 750Ch	ICU	IRQ control register 12	IRQCR12	8	8	2 ICLK
0008 750Dh	ICU	IRQ control register 13	IRQCR13	8	8	2 ICLK
0008 750Eh	ICU	IRQ control register 14	IRQCR14	8	8	2 ICLK
0008 750Fh	ICU	IRQ control register 15	IRQCR15	8	8	2 ICLK
0008 7580h	ICU	Non-maskable interrupt status register	NMISR	8	8	2 ICLK
0008 7581h	ICU	Non-maskable interrupt enable register	NMIER	8	8	2 ICLK
0008 7582h	ICU	Non-maskable interrupt clear register	NMICLR	8	8	2 ICLK
0008 7583h	ICU	NMI pin interrupt control register	NMICR	8	8	2 ICLK
0008 8000h	CMT	Compare match timer start register 0	CMSTR0	16	16	2 to 3 PCLK*8
0008 8002h	CMT0	Compare match timer control register	CMCR	16	16	2 to 3 PCLK*8
0008 8004h	CMT0	Compare match timer counter	CMCNT	16	16	2 to 3 PCLK*8
0008 8006h	CMT0	Compare match timer constant register	CMCOR	16	16	2 to 3 PCLK*8
0008 8008h	CMT1	Compare match timer control register	CMCR	16	16	2 to 3 PCLK*8
0008 800Ah	CMT1	Compare match timer counter	CMCNT	16	16	2 to 3 PCLK*8
0008 800Ch	CMT1	Compare match timer constant register	CMCOR	16	16	2 to 3 PCLK*8
0008 8010h	CMT	Compare match timer start register 1	CMSTR1	16	16	2 to 3 PCLK*8
0008 8012h	CMT2	Compare match timer control register	CMCR	16	16	2 to 3 PCLK*8
0008 8014h	CMT2	Compare match timer counter	CMCNT	16	16	2 to 3 PCLK*8
0008 8016h	CMT2	Compare match timer constant register	CMCOR	16	16	2 to 3 PCLK*8
0008 8018h	CMT3	Compare match timer control register	CMCR	16	16	2 to 3 PCLK*8
0008 801Ah	CMT3	Compare match timer counter	CMCNT	16	16	2 to 3 PCLK*8
0008 801Ch	CMT3	Compare match timer constant register	CMCOR	16	16	2 to 3 PCLK*8
0008 8028h	WDT	Timer control/status register	READ.TCSR	8	8	2 to 3 PCLK*8
0008 8028h	WDT	Write window A register	WRITE.WINA	16	16	2 to 3 PCLK*8
0008 8029h	WDT	Timer counter	READ.TCNT	8	8	2 to 3 PCLK*8
0008 802Ah	WDT	Write window B register	WRITE.WINB	16	16	2 to 3 PCLK*8
0008 802Bh	WDT	Reset control/status register	READ.RSTCSR	8	8	2 to 3 PCLK*8
0008 8030h	IWDT	IWDT refresh register	IWDTRR	8	8	2 to 3 PCLK*8
0008 8032h	IWDT	IWDT control register	IWDTCR	16	16	2 to 3 PCLK*8
0008 8034h	IWDT	IWDT status register	IWDTSR	16	16	2 to 3 PCLK*8
0008 8040h	AD0	A/D data register A	ADDRA	16	16	2 to 3 PCLK*8
0008 8042h	AD0	A/D data register B	ADDRB	16	16	2 to 3 PCLK*8
0008 8044h	AD0	A/D data register C	ADDRC	16	16	2 to 3 PCLK*8
0008 8046h	AD0	A/D data register D	ADDRD	16	16	2 to 3 PCLK*8
0008 8050h	AD0	A/D control/status register	ADCSR	8	8	2 to 3 PCLK*8
0008 8051h	AD0	A/D control register	ADCR	8	8	2 to 3 PCLK*8
0008 8052h	AD0	ADDRn format select register	ADDPR	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (15 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 8053h	AD0	A/D sampling state register	ADSSTR	8	8	2 to 3 PCLK*8
0008 805Fh	AD0	A/D self-diagnostic register	ADDIAGR	8	8	2 to 3 PCLK*8
0008 8060h	AD1	A/D data register A	ADDRA	16	16	2 to 3 PCLK*8
0008 8062h	AD1	A/D data register B	ADDRB	16	16	2 to 3 PCLK*8
0008 8064h	AD1	A/D data register C	ADDRC	16	16	2 to 3 PCLK*8
0008 8066h	AD1	A/D data register D	ADDRD	16	16	2 to 3 PCLK*8
0008 8070h	AD1	A/D control/status register	ADCSR	8	8	2 to 3 PCLK*8
0008 8071h	AD1	A/D control register	ADCR	8	8	2 to 3 PCLK*8
0008 8072h	AD1	ADDRn format select register	ADDPR	8	8	2 to 3 PCLK*8
0008 8073h	AD1	A/D sampling state register	ADSSTR	8	8	2 to 3 PCLK*8
0008 807Fh	AD1	A/D self-diagnostic register	ADDIAGR	8	8	2 to 3 PCLK*8
0008 80C0h	DA	D/A data register 0	DADR0	16	16	2 to 3 PCLK*8
0008 80C2h	DA	D/A data register 1	DADR1	16	16	2 to 3 PCLK*8
0008 80C4h	DA	D/A control register	DACR	8	8	2 to 3 PCLK*8
0008 80C5h	DA	DADRm format select register	DADPR	8	8	2 to 3 PCLK*8
0008 81E6h	PPG0	PPG output control register	PCR	8	8	2 to 3 PCLK*8
0008 81E7h	PPG0	PPG output mode register	PMR	8	8	2 to 3 PCLK*8
0008 81E8h	PPG0	Next data enable register H	NDERH	8	8	2 to 3 PCLK*8
0008 81E9h	PPG0	Next data enable register L	NDERL	8	8	2 to 3 PCLK*8
0008 81EAh	PPG0	Output data register H	PODRH	8	8	2 to 3 PCLK*8
0008 81EBh	PPG0	Output data register L	PODRL	8	8	2 to 3 PCLK*8
0008 81ECh*1	PPG0	Next data register H	NDRH	8	8	2 to 3 PCLK*8
0008 81EDh*2	PPG0	Next data register L	NDRL	8	8	2 to 3 PCLK*8
0008 81EEh*1	PPG0	Next data register H2	NDRH2	8	8	2 to 3 PCLK*8
0008 81EFh*2	PPG0	Next data register L2	NDRL2	8	8	2 to 3 PCLK*8
0008 81F0h	PPG1	PPG trigger select register	PTRSLR	8	8	2 to 3 PCLK*8
0008 81F6h	PPG1	PPG output control register	PCR	8	8	2 to 3 PCLK*8
0008 81F7h	PPG1	PPG output mode register	PMR	8	8	2 to 3 PCLK*8
0008 81F8h	PPG1	Next data enable register H	NDERH	8	8	2 to 3 PCLK*8
0008 81F9h	PPG1	Next data enable register L	NDERL	8	8	2 to 3 PCLK*8
0008 81FAh	PPG1	Output data register H	PODRH	8	8	2 to 3 PCLK*8
0008 81FBh	PPG1	Output data register L	PODRL	8	8	2 to 3 PCLK*8
0008 81FCh*3	PPG1	Next data register H	NDRH	8	8	2 to 3 PCLK*8
0008 81FDh*4	PPG1	Next data register L	NDRL	8	8	2 to 3 PCLK*8
0008 81FEh*3	PPG1	Next data register H2	NDRH2	8	8	2 to 3 PCLK*8
0008 81FFh*4	PPG1	Next data register L2	NDRL2	8	8	2 to 3 PCLK*8
0008 8200h	TMR0	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8201h	TMR1	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8202h	TMR0	Timer control/status register	TCSR	8	8	2 to 3 PCLK*8
0008 8203h	TMR1	Timer control/status register	TCSR	8	8	2 to 3 PCLK*8
0008 8204h	TMR0	Time constant register A	TCORA	8	8	2 to 3 PCLK*8
0008 8205h	TMR1	Time constant register A	TCORA	8	8	2 to 3 PCLK*8
0008 8206h	TMR0	Time constant register B	TCORB	8	8	2 to 3 PCLK*8
0008 8207h	TMR1	Time constant register B	TCORB	8	8	2 to 3 PCLK*8
0008 8208h	TMR0	Timer counter	TCNT	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (16 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 8209h	TMR1	Timer counter	TCNT	8	8	2 to 3 PCLK*8
0008 820Ah	TMR0	Timer counter control register	TCCR	8	8	2 to 3 PCLK*8
0008 820Bh	TMR1	Timer counter control register	TCCR	8	8	2 to 3 PCLK*8
0008 8204h	TMR01	Time constant register A	TCORA	16	16	2 to 3 PCLK*8
0008 8206h	TMR01	Time constant register B	TCORB	16	16	2 to 3 PCLK*8
0008 8208h	TMR01	Timer counter	TCNT	16	16	2 to 3 PCLK*8
0008 820Ah	TMR01	Timer counter control register	TCCR	16	16	2 to 3 PCLK*8
0008 8210h	TMR2	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8211h	TMR3	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8212h	TMR2	Timer control/status register	TCSR	8	8	2 to 3 PCLK*8
0008 8213h	TMR3	Timer control/status register	TCSR	8	8	2 to 3 PCLK*8
0008 8214h	TMR2	Time constant register A	TCORA	8	8	2 to 3 PCLK*8
0008 8215h	TMR3	Time constant register A	TCORA	8	8	2 to 3 PCLK*8
0008 8216h	TMR2	Time constant register B	TCORB	8	8	2 to 3 PCLK*8
0008 8217h	TMR3	Time constant register B	TCORB	8	8	2 to 3 PCLK*8
0008 8218h	TMR2	Timer counter	TCNT	8	8	2 to 3 PCLK*8
0008 8219h	TMR3	Timer counter	TCNT	8	8	2 to 3 PCLK*8
0008 821Ah	TMR2	Timer counter control register	TCCR	8	8	2 to 3 PCLK*8
0008 821Bh	TMR3	Timer counter control register	TCCR	8	8	2 to 3 PCLK*8
0008 8214h	TMR23	Time constant register A	TCORA	16	16	2 to 3 PCLK*8
0008 8216h	TMR23	Time constant register B	TCORB	16	16	2 to 3 PCLK*8
0008 8218h	TMR23	Timer counter	TCNT	16	16	2 to 3 PCLK*8
0008 821Ah	TMR23	Timer counter control register	TCCR	16	16	2 to 3 PCLK*8
0008 8240h	SCI0	Serial mode register	SMR	8	8	2 to 3 PCLK*8
0008 8241h	SCI0	Bit rate register	BRR	8	8	2 to 3 PCLK*8
0008 8242h	SCI0	Serial control register	SCR	8	8	2 to 3 PCLK*8
0008 8243h	SCI0	Transmit data register	TDR	8	8	2 to 3 PCLK*8
0008 8244h	SCI0	Serial status register	SSR	8	8	2 to 3 PCLK*8
0008 8245h	SCI0	Receive data register	RDR	8	8	2 to 3 PCLK*8
0008 8246h	SCI0	Smart card mode register	SCMR	8	8	2 to 3 PCLK*8
0008 8247h	SCI0	Serial extended mode register	SEMR	8	8	2 to 3 PCLK*8
0008 8240h	SMCI0	Serial mode register	SMR	8	8	2 to 3 PCLK*8
0008 8241h	SMCI0	Bit rate register	BRR	8	8	2 to 3 PCLK*8
0008 8242h	SMCI0	Serial control register	SCR	8	8	2 to 3 PCLK*8
0008 8243h	SMCI0	Transmit data register	TDR	8	8	2 to 3 PCLK*8
0008 8244h	SMCI0	Serial status register	SSR	8	8	2 to 3 PCLK*8
0008 8245h	SMCI0	Receive data register	RDR	8	8	2 to 3 PCLK*8
0008 8246h	SMCI0	Smart card mode register	SCMR	8	8	2 to 3 PCLK*8
0008 8248h	SCI1	Serial mode register	SMR	8	8	2 to 3 PCLK*8
0008 8249h	SCI1	Bit rate register	BRR	8	8	2 to 3 PCLK*8
0008 824Ah	SCI1	Serial control register	SCR	8	8	2 to 3 PCLK*8
0008 824Bh	SCI1	Transmit data register	TDR	8	8	2 to 3 PCLK*8
0008 824Ch	SCI1	Serial status register	SSR	8	8	2 to 3 PCLK*8
0008 824Dh	SCI1	Receive data register	RDR	8	8	2 to 3 PCLK*8
0008 824Eh	SCI1	Smart card mode register	SCMR	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (17 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 824Fh	SCI1	Serial extended mode register	SEMR	8	8	2 to 3 PCLK*8
0008 8248h	SMCI1	Serial mode register	SMR	8	8	2 to 3 PCLK*8
0008 8249h	SMCI1	Bit rate register	BRR	8	8	2 to 3 PCLK*8
0008 824Ah	SMCI1	Serial control register	SCR	8	8	2 to 3 PCLK*8
0008 824Bh	SMCI1	Transmit data register	TDR	8	8	2 to 3 PCLK*8
0008 824Ch	SMCI1	Serial status register	SSR	8	8	2 to 3 PCLK*8
0008 824Dh	SMCI1	Receive data register	RDR	8	8	2 to 3 PCLK*8
0008 824Eh	SMCI1	Smart card mode register	SCMR	8	8	2 to 3 PCLK*8
0008 8250h	SCI2	Serial mode register	SMR	8	8	2 to 3 PCLK*8
0008 8251h	SCI2	Bit rate register	BRR	8	8	2 to 3 PCLK*8
0008 8252h	SCI2	Serial control register	SCR	8	8	2 to 3 PCLK*8
0008 8253h	SCI2	Transmit data register	TDR	8	8	2 to 3 PCLK*8
0008 8254h	SCI2	Serial status register	SSR	8	8	2 to 3 PCLK*8
0008 8255h	SCI2	Receive data register	RDR	8	8	2 to 3 PCLK*8
0008 8256h	SCI2	Smart card mode register	SCMR	8	8	2 to 3 PCLK*8
0008 8257h	SCI2	Serial extended mode register	SEMR	8	8	2 to 3 PCLK*8
0008 8250h	SMCI2	Serial mode register	SMR	8	8	2 to 3 PCLK*8
0008 8251h	SMCI2	Bit rate register	BRR	8	8	2 to 3 PCLK*8
0008 8252h	SMCI2	Serial control register	SCR	8	8	2 to 3 PCLK*8
0008 8253h	SMCI2	Transmit data register	TDR	8	8	2 to 3 PCLK*8
0008 8254h	SMCI2	Serial status register	SSR	8	8	2 to 3 PCLK*8
0008 8255h	SMCI2	Receive data register	RDR	8	8	2 to 3 PCLK*8
0008 8256h	SMCI2	Smart card mode register	SCMR	8	8	2 to 3 PCLK*8
0008 8258h	SCI3	Serial mode register	SMR	8	8	2 to 3 PCLK*8
0008 8259h	SCI3	Bit rate register	BRR	8	8	2 to 3 PCLK*8
0008 825Ah	SCI3	Serial control register	SCR	8	8	2 to 3 PCLK*8
0008 825Bh	SCI3	Transmit data register	TDR	8	8	2 to 3 PCLK*8
0008 825Ch	SCI3	Serial status register	SSR	8	8	2 to 3 PCLK*8
0008 825Dh	SCI3	Receive data register	RDR	8	8	2 to 3 PCLK*8
0008 825Eh	SCI3	Smart card mode register	SCMR	8	8	2 to 3 PCLK*8
0008 825Fh	SCI3	Serial extended mode register	SEMR	8	8	2 to 3 PCLK*8
0008 8258h	SMCI3	Serial mode register	SMR	8	8	2 to 3 PCLK*8
0008 8259h	SMCI3	Bit rate register	BRR	8	8	2 to 3 PCLK*8
0008 825Ah	SMCI3	Serial control register	SCR	8	8	2 to 3 PCLK*8
0008 825Bh	SMCI3	Transmit data register	TDR	8	8	2 to 3 PCLK*8
0008 825Ch	SMCI3	Serial status register	SSR	8	8	2 to 3 PCLK*8
0008 825Dh	SMCI3	SMCI3 Receive data register	RDR	8	8	2 to 3 PCLK*8
0008 825Eh	SMCI3	SMCI3 Smart card mode register	SCMR	8	8	2 to 3 PCLK*8
0008 8268h	SCI5	Serial mode register	SMR	8	8	2 to 3 PCLK*8
0008 8269h	SCI5	Bit rate register	BRR	8	8	2 to 3 PCLK*8
0008 826Ah	SCI5	Serial control register	SCR	8	8	2 to 3 PCLK*8
0008 826Bh	SCI5	Transmit data register	TDR	8	8	2 to 3 PCLK*8
0008 826Ch	SCI5	Serial status register	SSR	8	8	2 to 3 PCLK*8
0008 826Dh	SCI5	Receive data register	RDR	8	8	2 to 3 PCLK*8
0008 826Eh	SCI5	Smart card mode register	SCMR	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (18 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 826Fh	SCI5	Serial extended mode register	SEMR	8	8	2 to 3 PCLK*8
0008 8268h	SMCI5	Serial mode register	SMR	8	8	2 to 3 PCLK*8
0008 8269h	SMCI5	Bit rate register	BRR	8	8	2 to 3 PCLK*8
0008 826Ah	SMCI5	Serial control register	SCR	8	8	2 to 3 PCLK*8
0008 826Bh	SMCI5	Transmit data register	TDR	8	8	2 to 3 PCLK*8
0008 826Ch	SMCI5	Serial status register	SSR	8	8	2 to 3 PCLK*8
0008 826Dh	SMCI5	Receive data register	RDR	8	8	2 to 3 PCLK*8
0008 826Eh	SMCI5	Smart card mode register	SCMR	8	8	2 to 3 PCLK*8
0008 8270h	SCI6	Serial mode register	SMR	8	8	2 to 3 PCLK*8
0008 8271h	SCI6	Bit rate register	BRR	8	8	2 to 3 PCLK*8
0008 8272h	SCI6	Serial control register	SCR	8	8	2 to 3 PCLK*8
0008 8273h	SCI6	Transmit data register	TDR	8	8	2 to 3 PCLK*8
0008 8274h	SCI6	Serial status register	SSR	8	8	2 to 3 PCLK*8
0008 8275h	SCI6	Receive data register	RDR	8	8	2 to 3 PCLK*8
0008 8276h	SCI6	Smart card mode register	SCMR	8	8	2 to 3 PCLK*8
0008 8277h	SCI6	Serial extended mode register	SEMR	8	8	2 to 3 PCLK*8
0008 8270h	SMCI6	Serial mode register	SMR	8	8	2 to 3 PCLK*8
0008 8271h	SMCI6	Bit rate register	BRR	8	8	2 to 3 PCLK*8
0008 8272h	SMCI6	Serial control register	SCR	8	8	2 to 3 PCLK*8
0008 8273h	SMCI6	Transmit data register	TDR	8	8	2 to 3 PCLK*8
0008 8274h	SMCI6	Serial status register	SSR	8	8	2 to 3 PCLK*8
0008 8275h	SMCI6	Receive data register	RDR	8	8	2 to 3 PCLK*8
0008 8276h	SMCI6	Smart card mode register	SCMR	8	8	2 to 3 PCLK*8
0008 8280h	CRC	CRC control register	CRCCR	8	8	2 to 3 PCLK*8
0008 8281h	CRC	CRC data input register	CRCDIR	8	8	2 to 3 PCLK*8
0008 8282h	CRC	CRC data output register	CRCDOR	16	16	2 to 3 PCLK*8
0008 8300h	RIIC0	I ² C bus control register 1	ICCR1	8	8	2 to 3 PCLK*8
0008 8301h	RIIC0	I ² C bus control register 2	ICCR2	8	8	2 to 3 PCLK*8
0008 8302h	RIIC0	I ² C bus mode register 1	ICMR1	8	8	2 to 3 PCLK*8
0008 8303h	RIIC0	I ² C bus mode register 2	ICMR2	8	8	2 to 3 PCLK*8
0008 8304h	RIIC0	I ² C bus mode register 3	ICMR3	8	8	2 to 3 PCLK*8
0008 8305h	RIIC0	I ² C bus function enable register	ICFER	8	8	2 to 3 PCLK*8
0008 8306h	RIIC0	I ² C bus status enable register	ICSER	8	8	2 to 3 PCLK*8
0008 8307h	RIIC0	I ² C bus interrupt enable register	ICIER	8	8	2 to 3 PCLK*8
0008 8308h	RIIC0	I ² C bus status register 1	ICSR1	8	8	2 to 3 PCLK*8
0008 8309h	RIIC0	I ² C bus status register 2	ICSR2	8	8	2 to 3 PCLK*8
0008 830Ah	RIIC0	Slave address register L0	SARL0	8	8	2 to 3 PCLK*8
0008 830Bh	RIIC0	Slave address register U0	SARU0	8	8	2 to 3 PCLK*8
0008 830Ch	RIIC0	Slave address register L1	SARL1	8	8	2 to 3 PCLK*8
0008 830Dh	RIIC0	Slave address register U1	SARU1	8	8	2 to 3 PCLK*8
0008 830Eh	RIIC0	Slave address register L2	SARL2	8	8	2 to 3 PCLK*8
0008 830Fh	RIIC0	Slave address register U2	SARU2	8	8	2 to 3 PCLK*8
0008 8310h	RIIC0	I ² C bus bit rate low-level register	ICBRL	8	8	2 to 3 PCLK*8
0008 8311h	RIIC0	I ² C bus bit rate high-level register	ICBRH	8	8	2 to 3 PCLK*8
0008 8312h	RIIC0	I ² C bus transmit data register	ICDRT	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (19 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 8313h	RIIC0	I ² C bus receive data register	ICDRR	8	8	2 to 3 PCLK*8
0008 8320h	RIIC1	I ² C bus control register 1	ICCR1	8	8	2 to 3 PCLK*8
0008 8321h	RIIC1	I ² C bus control register 2	ICCR2	8	8	2 to 3 PCLK*8
0008 8322h	RIIC1	I ² C bus mode register 1	ICMR1	8	8	2 to 3 PCLK*8
0008 8323h	RIIC1	I ² C bus mode register 2	ICMR2	8	8	2 to 3 PCLK*8
0008 8324h	RIIC1	I ² C bus mode register 3	ICMR3	8	8	2 to 3 PCLK*8
0008 8325h	RIIC1	I ² C bus function enable register	ICFER	8	8	2 to 3 PCLK*8
0008 8326h	RIIC1	I ² C bus status enable register	ICSER	8	8	2 to 3 PCLK*8
0008 8327h	RIIC1	I ² C bus interrupt enable register	ICIER	8	8	2 to 3 PCLK*8
0008 8328h	RIIC1	I ² C bus status register 1	ICSR1	8	8	2 to 3 PCLK*8
0008 8329h	RIIC1	I ² C bus status register 2	ICSR2	8	8	2 to 3 PCLK*8
0008 832Ah	RIIC1	Slave address register L0	SARL0	8	8	2 to 3 PCLK*8
0008 832Bh	RIIC1	Slave address register U0	SARU0	8	8	2 to 3 PCLK*8
0008 832Ch	RIIC1	Slave address register L1	SARL1	8	8	2 to 3 PCLK*8
0008 832Dh	RIIC1	Slave address register U1	SARU1	8	8	2 to 3 PCLK*8
0008 832Eh	RIIC1	Slave address register L2	SARL2	8	8	2 to 3 PCLK*8
0008 832Fh	RIIC1	Slave address register U2	SARU2	8	8	2 to 3 PCLK*8
0008 8330h	RIIC1	I ² C bus bit rate low-level register	ICBRL	8	8	2 to 3 PCLK*8
0008 8331h	RIIC1	I ² C bus bit rate high-level register	ICBRH	8	8	2 to 3 PCLK*8
0008 8332h	RIIC1	I ² C bus transmit data register	ICDRT	8	8	2 to 3 PCLK*8
0008 8333h	RIIC1	I ² C bus receive data register	ICDRR	8	8	2 to 3 PCLK*8
0008 8380h	RSPI0	RSPI control register	SPCR	8	8	2 to 3 PCLK*8
0008 8381h	RSPI0	RSPI slave select polarity register	SSLP	8	8	2 to 3 PCLK*8
0008 8382h	RSPI0	RSPI pin control register	SPPCR	8	8	2 to 3 PCLK*8
0008 8383h	RSPI0	RSPI status register	SPSR	8	8	2 to 3 PCLK*8
0008 8384h	RSPI0	RSPI data register	SPDR	32	16, 32	2 to 3 PCLK*8
0008 8388h	RSPI0	RSPI sequence control register	SPSCR	8	8	2 to 3 PCLK*8
0008 8389h	RSPI0	RSPI sequence status register	SPSSR	8	8	2 to 3 PCLK*8
0008 838Ah	RSPI0	RSPI bit rate register	SPBR	8	8	2 to 3 PCLK*8
0008 838Bh	RSPI0	RSPI data control register	SPDCR	8	8	2 to 3 PCLK*8
0008 838Ch	RSPI0	RSPI clock delay register	SPCKD	8	8	2 to 3 PCLK*8
0008 838Dh	RSPI0	RSPI slave select negation delay register	SSLND	8	8	2 to 3 PCLK*8
0008 838Eh	RSPI0	RSPI next-access delay register	SPND	8	8	2 to 3 PCLK*8
0008 838Fh	RSPI0	RSPI control register 2	SPCR2	8	8	2 to 3 PCLK*8
0008 8390h	RSPI0	RSPI command register 0	SPCMD0	16	16	2 to 3 PCLK*8
0008 8392h	RSPI0	RSPI command register 1	SPCMD1	16	16	2 to 3 PCLK*8
0008 8394h	RSPI0	RSPI command register 2	SPCMD2	16	16	2 to 3 PCLK*8
0008 8396h	RSPI0	RSPI command register 3	SPCMD3	16	16	2 to 3 PCLK*8
0008 8398h	RSPI0	RSPI command register 4	SPCMD4	16	16	2 to 3 PCLK*8
0008 839Ah	RSPI0	RSPI command register 5	SPCMD5	16	16	2 to 3 PCLK*8
0008 839Ch	RSPI0	RSPI command register 6	SPCMD6	16	16	2 to 3 PCLK*8
0008 839Eh	RSPI0	RSPI command register 7	SPCMD7	16	16	2 to 3 PCLK*8
0008 83A0h	RSPI1	RSPI control register	SPCR	8	8	2 to 3 PCLK*8
0008 83A1h	RSPI1	RSPI slave select polarity register	SSLP	8	8	2 to 3 PCLK*8
0008 83A2h	RSPI1	RSPI pin control register	SPPCR	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (20 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 83A3h	RSPI1	RSPI status register	SPSR	8	8	2 to 3 PCLK*8
0008 83A4h	RSPI1	RSPI data register	SPDR	32	16, 32	2 to 3 PCLK*8
0008 83A8h	RSPI1	RSPI sequence control register	SPSCR	8	8	2 to 3 PCLK*8
0008 83A9h	RSPI1	RSPI sequence status register	SPSSR	8	8	2 to 3 PCLK*8
0008 83AAh	RSPI1	RSPI bit rate register	SPBR	8	8	2 to 3 PCLK*8
0008 83ABh	RSPI1	RSPI data control register	SPDCR	8	8	2 to 3 PCLK*8
0008 83ACh	RSPI1	RSPI clock delay register	SPCKD	8	8	2 to 3 PCLK*8
0008 83ADh	RSPI1	RSPI slave select negation delay register	SSLND	8	8	2 to 3 PCLK*8
0008 83AEh	RSPI1	RSPI next-access delay register	SPND	8	8	2 to 3 PCLK*8
0008 83AFh	RSPI1	RSPI control register 2	SPCR2	8	8	2 to 3 PCLK*8
0008 83B0h	RSPI1	RSPI command register 0	SPCMD0	16	16	2 to 3 PCLK*8
0008 83B2h	RSPI1	RSPI command register 1	SPCMD1	16	16	2 to 3 PCLK*8
0008 83B4h	RSPI1	RSPI command register 2	SPCMD2	16	16	2 to 3 PCLK*8
0008 83B6h	RSPI1	RSPI command register 3	SPCMD3	16	16	2 to 3 PCLK*8
0008 83B8h	RSPI1	RSPI command register 4	SPCMD4	16	16	2 to 3 PCLK*8
0008 83BAh	RSPI1	RSPI command register 5	SPCMD5	16	16	2 to 3 PCLK*8
0008 83BCh	RSPI1	RSPI command register 6	SPCMD6	16	16	2 to 3 PCLK*8
0008 83BEh	RSPI1	RSPI command register 7	SPCMD7	16	16	2 to 3 PCLK*8
0008 8600h	MTU3	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8601h	MTU4	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8602h	MTU3	Timer mode register	TMDR	8	8	2 to 3 PCLK*8
0008 8603h	MTU4	Timer mode register	TMDR	8	8	2 to 3 PCLK*8
0008 8604h	MTU3	Timer I/O control register H	TIORH	8	8	2 to 3 PCLK*8
0008 8605h	MTU3	Timer I/O control register L	TIORL	8	8	2 to 3 PCLK*8
0008 8606h	MTU4	Timer I/O control register H	TIORH	8	8	2 to 3 PCLK*8
0008 8607h	MTU4	Timer I/O control register L	TIORL	8	8	2 to 3 PCLK*8
0008 8608h	MTU3	Timer interrupt enable register	TIER	8	8	2 to 3 PCLK*8
0008 8609h	MTU4	Timer interrupt enable register	TIER	8	8	2 to 3 PCLK*8
0008 860Ah	MTUA	Timer output master enable register	TOER	8	8	2 to 3 PCLK*8
0008 860Dh	MTUA	Timer gate control register	TGCR	8	8	2 to 3 PCLK*8
0008 860Eh	MTUA	Timer output control register 1	TOCR1	8	8	2 to 3 PCLK*8
0008 860Fh	MTUA	Timer output control register 2	TOCR2	8	8	2 to 3 PCLK*8
0008 8610h	MTU3	Timer counter	TCNT	16	16	2 to 3 PCLK*8
0008 8612h	MTU4	Timer counter	TCNT	16	16	2 to 3 PCLK*8
0008 8614h	MTUA	Timer cycle data register	TCDR	16	16	2 to 3 PCLK*8
0008 8616h	MTUA	Timer dead time data register	TDDR	16	16	2 to 3 PCLK*8
0008 8618h	MTU3	Timer general register A	TGRA	16	16	2 to 3 PCLK*8
0008 861Ah	MTU3	Timer general register B	TGRB	16	16	2 to 3 PCLK*8
0008 861Ch	MTU4	Timer general register A	TGRA	16	16	2 to 3 PCLK*8
0008 861Eh	MTU4	Timer general register B	TGRB	16	16	2 to 3 PCLK*8
0008 8620h	MTUA	Timer subcounter	TCNTS	16	16	2 to 3 PCLK*8
0008 8622h	MTUA	Timer cycle buffer register	TCBR	16	16	2 to 3 PCLK*8
0008 8624h	MTU3	Timer general register C	TGRC	16	16	2 to 3 PCLK*8
0008 8626h	MTU3	Timer general register D	TGRD	16	16	2 to 3 PCLK*8
0008 8628h	MTU4	Timer general register C	TGRC	16	16	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (21 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 862Ah	MTU4	Timer general register D	TGRD	16	16	2 to 3 PCLK*8
0008 862Ch	MTU3	Timer status register	TSR	8	8	2 to 3 PCLK*8
0008 862Dh	MTU4	Timer status register	TSR	8	8	2 to 3 PCLK*8
0008 8630h	MTUA	Timer interrupt skipping set register	TITCR	8	8	2 to 3 PCLK*8
0008 8631h	MTUA	Timer interrupt skipping counter	TITCNT	8	8	2 to 3 PCLK*8
0008 8632h	MTUA	Timer buffer transfer set register	TBTER	8	8	2 to 3 PCLK*8
0008 8634h	MTUA	Timer dead time enable register	TDER	8	8	2 to 3 PCLK*8
0008 8636h	MTUA	Timer output level buffer register	TOLBR	8	8	2 to 3 PCLK*8
0008 8638h	MTU3	Timer buffer operation transfer mode register	TBTM	8	8	2 to 3 PCLK*8
0008 8639h	MTU4	Timer buffer operation transfer mode register	TBTM	8	8	2 to 3 PCLK*8
0008 8640h	MTU4	Timer A/D converter start request control register	TADCR	16	16	2 to 3 PCLK*8
0008 8644h	MTU4	Timer A/D converter start request cycle set register A	TADCORA	16	16	2 to 3 PCLK*8
0008 8646h	MTU4	Timer A/D converter start request cycle set register B	TADCORB	16	16	2 to 3 PCLK*8
0008 8648h	MTU4	Timer A/D converter start request cycle set buffer register A	TADCOBRA	16	16	2 to 3 PCLK*8
0008 864Ah	MTU4	Timer A/D converter start request cycle set buffer register B	TADCOBRB	16	16	2 to 3 PCLK*8
0008 8660h	MTUA	Timer waveform control register	TWCR	8	8	2 to 3 PCLK*8
0008 8680h	MTUA	Timer start register	TSTR	8	8	2 to 3 PCLK*8
0008 8681h	MTUA	Timer synchronous register	TSYR	8	8	2 to 3 PCLK*8
0008 8684h	MTUA	Timer read/write enable register	TRWER	8	8	2 to 3 PCLK*8
0008 8700h	MTU0	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8701h	MTU0	Timer mode register	TMDR	8	8	2 to 3 PCLK*8
0008 8702h	MTU0	Timer I/O control register H	TIORH	8	8	2 to 3 PCLK*8
0008 8703h	MTU0	Timer I/O control register L	TIORL	8	8	2 to 3 PCLK*8
0008 8704h	MTU0	Timer interrupt enable register	TIER	8	8	2 to 3 PCLK*8
0008 8705h	MTU0	Timer status register	TSR	8	8	2 to 3 PCLK*8
0008 8706h	MTU0	Timer counter	TCNT	16	16	2 to 3 PCLK*8
0008 8708h	MTU0	Timer general register A	TGRA	16	16	2 to 3 PCLK*8
0008 870Ah	MTU0	Timer general register B	TGRB	16	16	2 to 3 PCLK*8
0008 870Ch	MTU0	Timer general register C	TGRC	16	16	2 to 3 PCLK*8
0008 870Eh	MTU0	Timer general register D	TGRD	16	16	2 to 3 PCLK*8
0008 8720h	MTU0	Timer general register E	TGRE	16	16	2 to 3 PCLK*8
0008 8722h	MTU0	Timer general register F	TGRF	16	16	2 to 3 PCLK*8
0008 8724h	MTU0	Timer interrupt enable register 2	TIER2	8	8	2 to 3 PCLK*8
0008 8726h	MTU0	Timer buffer operation transfer mode register	TBTM	8	8	2 to 3 PCLK*8
0008 8780h	MTU1	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8781h	MTU1	Timer mode register	TMDR	8	8	2 to 3 PCLK*8
0008 8782h	MTU1	Timer I/O control register	TIOR	8	8	2 to 3 PCLK*8
0008 8784h	MTU1	Timer interrupt enable register	TIER	8	8	2 to 3 PCLK*8
0008 8785h	MTU1	Timer status register	TSR	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (22 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 8786h	MTU1	Timer counter	TCNT	16	16	2 to 3 PCLK*8
0008 8788h	MTU1	Timer general register A	TGRA	16	16	2 to 3 PCLK*8
0008 878Ah	MTU1	Timer general register B	TGRB	16	16	2 to 3 PCLK*8
0008 8790h	MTU1	Timer input capture control register	TICCR	8	8	2 to 3 PCLK*8
0008 8800h	MTU2	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8801h	MTU2	Timer mode register	TMDR	8	8	2 to 3 PCLK*8
0008 8802h	MTU2	Timer I/O control register	TIOR	8	8	2 to 3 PCLK*8
0008 8804h	MTU2	Timer interrupt enable register	TIER	8	8	2 to 3 PCLK*8
0008 8805h	MTU2	Timer status register	TSR	8	8	2 to 3 PCLK*8
0008 8806h	MTU2	Timer counter	TCNT	16	16	2 to 3 PCLK*8
0008 8808h	MTU2	Timer general register A	TGRA	16	16	2 to 3 PCLK*8
0008 880Ah	MTU2	Timer general register B	TGRB	16	16	2 to 3 PCLK*8
0008 8880h	MTU5	Timer counter U	TCNTU	16	16	2 to 3 PCLK*8
0008 8882h	MTU5	Timer general register U	TGRU	16	16	2 to 3 PCLK*8
0008 8884h	MTU5	Timer control register U	TCRU	8	8	2 to 3 PCLK*8
0008 8886h	MTU5	Timer I/O control register U	TIORU	8	8	2 to 3 PCLK*8
0008 8890h	MTU5	Timer counter V	TCNTV	16	16	2 to 3 PCLK*8
0008 8892h	MTU5	Timer general register V	TGRV	16	16	2 to 3 PCLK*8
0008 8894h	MTU5	Timer control register V	TCRV	8	8	2 to 3 PCLK*8
0008 8896h	MTU5	Timer I/O control register V	TIORV	8	8	2 to 3 PCLK*8
0008 88A0h	MTU5	Timer counter W	TCNTW	16	16	2 to 3 PCLK*8
0008 88A2h	MTU5	Timer general register W	TGRW	16	16	2 to 3 PCLK*8
0008 88A4h	MTU5	Timer control register W	TCRW	8	8	2 to 3 PCLK*8
0008 88A6h	MTU5	Timer I/O control register W	TIORW	8	8	2 to 3 PCLK*8
0008 88B2h	MTU5	Timer interrupt enable register	TIER	8	8	2 to 3 PCLK*8
0008 88B4h	MTU5	Timer start register	TSTR	8	8	2 to 3 PCLK*8
0008 88B6h	MTU5	Timer compare match clear register	TCNTCMPCLR	8	8	2 to 3 PCLK*8
0008 8900h	POE	Input level control/status register 1	ICSR1	16	16	2 to 3 PCLK*8
0008 8902h	POE	Output level control/status register 1	OCSR1	16	16	2 to 3 PCLK*8
0008 8904h	POE	Input level control/status register 2	ICSR2	16	16	2 to 3 PCLK*8
0008 8906h	POE	Output level control/status register 2	OCSR2	16	16	2 to 3 PCLK*8
0008 8908h	POE	Input level control/status register 3	ICSR3	16	16	2 to 3 PCLK*8
0008 890Ah	POE	Software port output enable register	SPOER	8	8	2 to 3 PCLK*8
0008 890Bh	POE	Port output enable control register 1	POECR1	8	8	2 to 3 PCLK*8
0008 890Ch	POE	Port output enable control register 2	POECR2	16	16	2 to 3 PCLK*8
0008 890Eh	POE	Input level control/status register 4	ICSR4	16	16	2 to 3 PCLK*8
0008 8A00h	MTU9	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8A01h	MTU10	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8A02h	MTU9	Timer mode register	TMDR	8	8	2 to 3 PCLK*8
0008 8A03h	MTU10	Timer mode register	TMDR	8	8	2 to 3 PCLK*8
0008 8A04h	MTU9	Timer I/O control register H	TIORH	8	8	2 to 3 PCLK*8
0008 8A05h	MTU9	Timer I/O control register L	TIORL	8	8	2 to 3 PCLK*8
0008 8A06h	MTU10	Timer I/O control register H	TIORH	8	8	2 to 3 PCLK*8
0008 8A07h	MTU10	Timer I/O control register L	TIORL	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (23 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 8A08h	MTU9	Timer interrupt enable register	TIER	8	8	2 to 3 PCLK*8
0008 8A09h	MTU10	Timer interrupt enable register	TIER	8	8	2 to 3 PCLK*8
0008 8A0Ah	MTUB	Timer output master enable register	TOER	8	8	2 to 3 PCLK*8
0008 8A0Dh	MTUB	Timer gate control register	TGCR	8	8	2 to 3 PCLK*8
0008 8A0Eh	MTUB	Timer output control register 1	TOCR1	8	8	2 to 3 PCLK*8
0008 8A0Fh	MTUB	Timer output control register 2	TOCR2	8	8	2 to 3 PCLK*8
0008 8A10h	MTU9	Timer counter	TCNT	16	16	2 to 3 PCLK*8
0008 8A12h	MTU10	Timer counter	TCNT	16	16	2 to 3 PCLK*8
0008 8A14h	MTUB	Timer cycle data register	TCDR	16	16	2 to 3 PCLK*8
0008 8A16h	MTUB	Timer dead time data register	TDDR	16	16	2 to 3 PCLK*8
0008 8A18h	MTU9	Timer general register A	TGRA	16	16	2 to 3 PCLK*8
0008 8A1Ah	MTU9	Timer general register B	TGRB	16	16	2 to 3 PCLK*8
0008 8A1Ch	MTU10	Timer general register A	TGRA	16	16	2 to 3 PCLK*8
0008 8A1Eh	MTU10	Timer general register B	TGRB	16	16	2 to 3 PCLK*8
0008 8A20h	MTUB	Timer subcounter	TCNTS	16	16	2 to 3 PCLK*8
0008 8A22h	MTUB	MTUB Timer cycle buffer register	TCBR	16	16	2 to 3 PCLK*8
0008 8A24h	MTU9	Timer general register C	TGRC	16	16	2 to 3 PCLK*8
0008 8A26h	MTU9	Timer general register D	TGRD	16	16	2 to 3 PCLK*8
0008 8A28h	MTU10	Timer general register C	TGRC	16	16	2 to 3 PCLK*8
0008 8A2Ah	MTU10	Timer general register D	TGRD	16	16	2 to 3 PCLK*8
0008 8A2Ch	MTU9	Timer status register	TSR	8	8	2 to 3 PCLK*8
0008 8A2Dh	MTU10	Timer status register	TSR	8	8	2 to 3 PCLK*8
0008 8A30h	MTUB	Timer interrupt skipping set register	TITCR	8	8	2 to 3 PCLK*8
0008 8A31h	MTUB	Timer interrupt skipping counter	TITCNT	8	8	2 to 3 PCLK*8
0008 8A32h	MTUB	TUB Timer dead time enable register	TBTER	8	8	2 to 3 PCLK*8
0008 8A34h	MTUB	Timer dead time enable register	TDER	8	8	2 to 3 PCLK*8
0008 8A36h	MTUB	Timer output level buffer register	TOLBR	8	8	2 to 3 PCLK*8
0008 8A38h	MTU9	Timer buffer operation transfer mode register	TBTM	8	8	2 to 3 PCLK*8
0008 8A39h	MTU10	Timer buffer operation transfer mode register	TBTM	8	8	2 to 3 PCLK*8
0008 8A40h	MTU10	Timer A/D converter start request control register	TADCR	16	16	2 to 3 PCLK*8
0008 8A44h	MTU10	Timer A/D converter start request cycle set register A	TADCORA	16	16	2 to 3 PCLK*8
0008 8A46h	MTU10	Timer A/D converter start request cycle set register B	TADCORB	16	16	2 to 3 PCLK*8
0008 8A48h	MTU10	Timer A/D converter start request cycle set buffer register A	TADCOBRA	16	16	2 to 3 PCLK*8
0008 8A4Ah	MTU10	Timer A/D converter start request cycle set buffer register B	TADCOBRB	16	16	2 to 3 PCLK*8
0008 8A60h	MTUB	Timer waveform control register	TWCR	8	8	2 to 3 PCLK*8
0008 8A80h	MTUB	Timer start register	TSTR	8	8	2 to 3 PCLK*8
0008 8A81h	MTUB	MTUB Timer synchronous register	TSYR	8	8	2 to 3 PCLK*8
0008 8A84h	MTUB	MTUB Timer read/write enable register	TRWER	8	8	2 to 3 PCLK*8
0008 8B00h	MTU6	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8B01h	MTU6	Timer mode register	TMDR	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (24 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 8B02h	MTU6	Timer I/O control register H	TIORH	8	8	2 to 3 PCLK*8
0008 8B03h	MTU6	Timer I/O control register L	TIORL	8	8	2 to 3 PCLK*8
0008 8B04h	MTU6	Timer interrupt enable register	TIER	8	8	2 to 3 PCLK*8
0008 8B05h	MTU6	Timer status register	TSR	8	8	2 to 3 PCLK*8
0008 8B06h	MTU6	Timer counter	TCNT	16	16	2 to 3 PCLK*8
0008 8B08h	MTU6	Timer general register A	TGRA	16	16	2 to 3 PCLK*8
0008 8B0Ah	MTU6	Timer general register B	TGRB	16	16	2 to 3 PCLK*8
0008 8B0Ch	MTU6	Timer general register C	TGRC	16	16	2 to 3 PCLK*8
0008 8B0Eh	MTU6	Timer general register D	TGRD	16	16	2 to 3 PCLK*8
0008 8B20h	MTU6	Timer general register E	TGRE	16	16	2 to 3 PCLK*8
0008 8B22h	MTU6	Timer general register F	TGRF	16	16	2 to 3 PCLK*8
0008 8B24h	MTU6	Timer interrupt enable register 2	TIER2	8	8	2 to 3 PCLK*8
0008 8B26h	MTU6	Timer buffer operation transfer mode register	TBTM	8	8	2 to 3 PCLK*8
0008 8B80h	MTU7	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8B81h	MTU7	Timer mode register	TMDR	8	8	2 to 3 PCLK*8
0008 8B82h	MTU7	Timer I/O control register	TIOR	8	8	2 to 3 PCLK*8
0008 8B84h	MTU7	Timer interrupt enable register	TIER	8	8	2 to 3 PCLK*8
0008 8B85h	MTU7	Timer status register	TSR	8	8	2 to 3 PCLK*8
0008 8B86h	MTU7	Timer counter	TCNT	16	16	2 to 3 PCLK*8
0008 8B88h	MTU7	Timer general register A	TGRA	16	16	2 to 3 PCLK*8
0008 8B8Ah	MTU7	Timer general register B	TGRB	16	16	2 to 3 PCLK*8
0008 8B90h	MTU7	Timer input capture control register	TICCR	8	8	2 to 3 PCLK*8
0008 8C00h	MTU8	Timer control register	TCR	8	8	2 to 3 PCLK*8
0008 8C01h	MTU8	Timer mode register	TMDR	8	8	2 to 3 PCLK*8
0008 8C02h	MTU8	Timer I/O control register	TIOR	8	8	2 to 3 PCLK*8
0008 8C04h	MTU8	Timer interrupt enable register	TIER	8	8	2 to 3 PCLK*8
0008 8C05h	MTU8	Timer status register	TSR	8	8	2 to 3 PCLK*8
0008 8C06h	MTU8	Timer counter	TCNT	16	16	2 to 3 PCLK*8
0008 8C08h	MTU8	Timer general register A	TGRA	16	16	2 to 3 PCLK*8
0008 8C0Ah	MTU8	Timer general register B	TGRB	16	16	2 to 3 PCLK*8
0008 8C80h	MTU11	Timer counter U	TCNTU	16	16	2 to 3 PCLK*8
0008 8C82h	MTU11	Timer general register U	TGRU	16	16	2 to 3 PCLK*8
0008 8C84h	MTU11	Timer control register U	TCRU	8	8	2 to 3 PCLK*8
0008 8C86h	MTU11	Timer I/O control register U	TIORU	8	8	2 to 3 PCLK*8
0008 8C90h	MTU11	Timer counter V	TCNTV	16	16	2 to 3 PCLK*8
0008 8C92h	MTU11	Timer general register V	TGRV	16	16	2 to 3 PCLK*8
0008 8C94h	MTU11	Timer control register V	TCRV	8	8	2 to 3 PCLK*8
0008 8C96h	MTU11	Timer I/O control register V	TIORV	8	8	2 to 3 PCLK*8
0008 8CA0h	MTU11	Timer counter W	TCNTW	16	16	2 to 3 PCLK*8
0008 8CA2h	MTU11	Timer general register W	TGRW	16	16	2 to 3 PCLK*8
0008 8CA4h	MTU11	Timer control register W	TCRW	8	8	2 to 3 PCLK*8
0008 8CA6h	MTU11	Timer I/O control register W	TIORW	8	8	2 to 3 PCLK*8
0008 8CB2h	MTU11	Timer interrupt enable register	TIER	8	8	2 to 3 PCLK*8
0008 8CB4h	MTU11	Timer start register	TSTR	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (25 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 8CB6h	MTU11	Timer compare match clear register	TCNTCMPCLR	8	8	2 to 3 PCLK*8
0008 9000h	S12AD	A/D control register	ADCSR	8	8	2 to 3 PCLK*8
0008 9004h	S12AD	A/D channel select register	ADANS	16	16	2 to 3 PCLK*8
0008 9008h	S12AD	A/D-converted value addition mode select register	ADADS	16	16	2 to 3 PCLK*8
0008 900Ch	S12AD	A/D-converted value addition count select register	ADADC	8	8	2 to 3 PCLK*8
0008 900Eh	S12AD	A/D control extended register	ADCER	16	16	2 to 3 PCLK*8
0008 9010h	S12AD	A/D start trigger select register	ADSTRGR	8	8	2 to 3 PCLK*8
0008 9020h	S12AD	A/D data register 0	ADDR0	16	16	2 to 3 PCLK*8
0008 9022h	S12AD	A/D data register 1	ADDR1	16	16	2 to 3 PCLK*8
0008 9024h	S12AD	A/D data register 2	ADDR2	16	16	2 to 3 PCLK*8
0008 9026h	S12AD	A/D data register 3	ADDR3	16	16	2 to 3 PCLK*8
0008 9028h	S12AD	A/D data register 4	ADDR4	16	16	2 to 3 PCLK*8
0008 902Ah	S12AD	A/D data register 5	ADDR5	16	16	2 to 3 PCLK*8
0008 902Ch	S12AD	A/D data register 6	ADDR6	16	16	2 to 3 PCLK*8
0008 902Eh	S12AD	A/D data register 7	ADDR7	16	16	2 to 3 PCLK*8
0008 C000h	PORT0	Data direction register	DDR	8	8	2 to 3 PCLK*8
0008 C001h	PORT1	Data direction register	DDR	8	8	2 to 3 PCLK*8
0008 C002h	PORT2	Data direction register	DDR	8	8	2 to 3 PCLK*8
0008 C003h	PORT3	Data direction register	DDR	8	8	2 to 3 PCLK*8
0008 C004h	PORT4	Data direction register	DDR	8	8	2 to 3 PCLK*8
0008 C005h	PORT5	Data direction register	DDR	8	8	2 to 3 PCLK*8
0008 C006h	PORT6	Data direction register	DDR*6*7	8	8	2 to 3 PCLK*8
0008 C007h	PORT7	Data direction register	DDR*6*7	8	8	2 to 3 PCLK*8
0008 C008h	PORT8	Data direction register	DDR*6*7	8	8	2 to 3 PCLK*8
0008 C009h	PORT9	Data direction register	DDR*6*7	8	8	2 to 3 PCLK*8
0008 C00Ah	PORTA	Data direction register	DDR	8	8	2 to 3 PCLK*8
0008 C00Bh	PORTB	Data direction register	DDR	8	8	2 to 3 PCLK*8
0008 C00Ch	PORTC	Data direction register	DDR	8	8	2 to 3 PCLK*8
0008 C00Dh	PORTD	Data direction register	DDR	8	8	2 to 3 PCLK*8
0008 C00Eh	PORTE	Data direction register	DDR*7	8	8	2 to 3 PCLK*8
0008 C00Fh	PORTF	Data direction register	DDR*5*6*7	8	8	2 to 3 PCLK*8
0008 C010h	PORTG	Data direction register	DDR*5*6*7	8	8	2 to 3 PCLK*8
0008 C020h	PORT0	Data register	DR	8	8	2 to 3 PCLK*8
0008 C021h	PORT1	Data register	DR	8	8	2 to 3 PCLK*8
0008 C022h	PORT2	Data register	DR	8	8	2 to 3 PCLK*8
0008 C023h	PORT3	Data register	DR	8	8	2 to 3 PCLK*8
0008 C024h	PORT4	Data register	DR	8	8	2 to 3 PCLK*8
0008 C025h	PORT5	Data register	DR	8	8	2 to 3 PCLK*8
0008 C026h	PORT6	Data register	DR*6*7	8	8	2 to 3 PCLK*8
0008 C027h	PORT7	Data register	DR*6*7	8	8	2 to 3 PCLK*8
0008 C028h	PORT8	Data register	DR*6*7	8	8	2 to 3 PCLK*8
0008 C029h	PORT9	Data register	DR*6*7	8	8	2 to 3 PCLK*8
0008 C02Ah	PORTA	Data register	DR	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (26 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 C02Bh	PORTB	Data register	DR	8	8	2 to 3 PCLK*8
0008 C02Ch	PORTC	Data register	DR	8	8	2 to 3 PCLK*8
0008 C02Dh	PORTD	Data register	DR	8	8	2 to 3 PCLK*8
0008 C02Eh	PORTE	Data register	DR*7	8	8	2 to 3 PCLK*8
0008 C02Fh	PORTF	Data register	DR*5*6*7	8	8	2 to 3 PCLK*8
0008 C030h	PORTG	Data register	DR**5*6*7	8	8	2 to 3 PCLK*8
0008 C040h	PORT0	Port register	PORT	8	8	2 to 3 PCLK*8
0008 C041h	PORT1	Port register	PORT	8	8	2 to 3 PCLK*8
0008 C042h	PORT2	Port register	PORT	8	8	2 to 3 PCLK*8
0008 C043h	PORT3	Port register	PORT	8	8	2 to 3 PCLK*8
0008 C044h	PORT4	Port register	PORT	8	8	2 to 3 PCLK*8
0008 C045h	PORT5	Port register	PORT	8	8	2 to 3 PCLK*8
0008 C046h	PORT6	Port register	PORT*6*7	8	8	2 to 3 PCLK*8
0008 C047h	PORT7	Port register	PORT*6*7	8	8	2 to 3 PCLK*8
0008 C048h	PORT8	Port register	PORT*6*7	8	8	2 to 3 PCLK*8
0008 C049h	PORT9	Port register	PORT*6*7	8	8	2 to 3 PCLK*8
0008 C04Ah	PORTA	Port register	PORT	8	8	2 to 3 PCLK*8
0008 C04Bh	PORTB	Port register	PORT	8	8	2 to 3 PCLK*8
0008 C04Ch	PORTC	Port register	PORT	8	8	2 to 3 PCLK*8
0008 C04Dh	PORTD	Port register	PORT	8	8	2 to 3 PCLK*8
0008 C04Eh	PORTE	Port register	PORT*7	8	8	2 to 3 PCLK*8
0008 C04Fh	PORTF	Port register	PORT*5*6*7	8	8	2 to 3 PCLK*8
0008 C050h	PORTG	Port register	PORT*5*6*7	8	8	2 to 3 PCLK*8
0008 C060h	PORT0	Input buffer control register	ICR	8	8	2 to 3 PCLK*8
0008 C061h	PORT1	Input buffer control register	ICR	8	8	2 to 3 PCLK*8
0008 C062h	PORT2	Input buffer control register	ICR	8	8	2 to 3 PCLK*8
0008 C063h	PORT3	Input buffer control register	ICR	8	8	2 to 3 PCLK*8
0008 C064h	PORT4	Input buffer control register	ICR	8	8	2 to 3 PCLK*8
0008 C065h	PORT5	Input buffer control register	ICR	8	8	2 to 3 PCLK*8
0008 C066h	PORT6	Input buffer control register	ICR*6*7	8	8	2 to 3 PCLK*8
0008 C067h	PORT7	Input buffer control register	ICR*6*7	8	8	2 to 3 PCLK*8
0008 C068h	PORT8	Input buffer control register	ICR*6*7	8	8	2 to 3 PCLK*8
0008 C069h	PORT9	Input buffer control register	ICR*6*7	8	8	2 to 3 PCLK*8
0008 C06Ah	PORTA	Input buffer control register	ICR	8	8	2 to 3 PCLK*8
0008 C06Bh	PORTB	Input buffer control register	ICR	8	8	2 to 3 PCLK*8
0008 C06Ch	PORTC	Input buffer control register	ICR	8	8	2 to 3 PCLK*8
0008 C06Dh	PORTD	Input buffer control register	ICR	8	8	2 to 3 PCLK*8
0008 C06Eh	PORTE	Input buffer control register	ICR*7	8	8	2 to 3 PCLK*8
0008 C06Fh	PORTF	Input buffer control register	ICR*5*6*7	8	8	2 to 3 PCLK*8
0008 C070h	PORTG	Input buffer control register	ICR*5*6*7	8	8	2 to 3 PCLK*8
0008 C080h	PORT0	Open drain control register	ODR	8	8	2 to 3 PCLK*8
0008 C081h	PORT1	Open drain control register	ODR	8	8	2 to 3 PCLK*8
0008 C082h	PORT2	Open drain control register	ODR	8	8	2 to 3 PCLK*8
0008 C083h	PORT3	Open drain control register	ODR	8	8	2 to 3 PCLK*8
0008 C08Ch	PORTC	Open drain control register	ODR	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (27 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 C0C9h	PORT9	Pull-up resistor control register	PCR*6*7	8	8	2 to 3 PCLK*8
0008 C0CAh	PORTA	Pull-up resistor control register	PCR	8	8	2 to 3 PCLK*8
0008 C0CBh	PORTB	Pull-up resistor control register	PCR	8	8	2 to 3 PCLK*8
0008 C0CCh	PORTC	Pull-up resistor control register	PCR	8	8	2 to 3 PCLK*8
0008 C0CDh	PORTD	Pull-up resistor control register	PCR	8	8	2 to 3 PCLK*8
0008 C0CEh	PORTE	Pull-up resistor control register	PCR*7	8	8	2 to 3 PCLK*8
0008 C0D0h	PORTG	Pull-up resistor control register	PCR*5*6*7	8	8	2 to 3 PCLK*8
0008 C100h	IOPORT	Port function register 0	PF0CSE	8	8	2 to 3 PCLK*8
0008 C101h	IOPORT	Port function register 1	PF1CSS*6*7	8	8	2 to 3 PCLK*8
0008 C102h	IOPORT	Port function register 2	PF2CSS*6*7	8	8	2 to 3 PCLK*8
0008 C103h	IOPORT	Port function register 3	PF3BUS	8	8	2 to 3 PCLK*8
0008 C104h	IOPORT	Port function register 4	PF4BUS	8	8	2 to 3 PCLK*8
0008 C105h	IOPORT	Port function register 5	PF5BUS	8	8	2 to 3 PCLK*8
0008 C106h	IOPORT	Port function register 6	PF6BUS	8	8	2 to 3 PCLK*8
0008 C107h	IOPORT	Port function register 7	PF7DMA	8	8	2 to 3 PCLK*8
0008 C108h	IOPORT	Port function register 8	PF8IRQ	8	8	2 to 3 PCLK*8
0008 C109h	IOPORT	Port function register 9	PF9IRQ	8	8	2 to 3 PCLK*8
0008 C10Ah	IOPORT	Port function register A	PFAADC	8	8	2 to 3 PCLK*8
0008 C10Bh	IOPORT	Port function register B	PFBTMR	8	8	2 to 3 PCLK*8
0008 C10Ch	IOPORT	Port function register C	PFCMTU	8	8	2 to 3 PCLK*8
0008 C10Dh	IOPORT	Port function register D	PFDMTU	8	8	2 to 3 PCLK*8
0008 C10Eh	IOPORT	Port function register E	PFENET	8	8	2 to 3 PCLK*8
0008 C10Fh	IOPORT	Port function register F	PFFSCI	8	8	2 to 3 PCLK*8
0008 C110h	IOPORT	Port function register G	PFGSPI	8	8	2 to 3 PCLK*8
0008 C111h	IOPORT	Port function register H	PFHSPI	8	8	2 to 3 PCLK*8
0008 C113h	IOPORT	Port function register J	PFJCAN	8	8	2 to 3 PCLK*8
0008 C114h	IOPORT	Port function register K	PFKUSB	8	8	2 to 3 PCLK*8
0008 C115h	IOPORT	Port function register L	PFLUSB*6*7	8	8	2 to 3 PCLK*8
0008 C116h	IOPORT	Port function register M	PFMPOE*7	8	8	2 to 3 PCLK*8
0008 C117h	IOPORT	Port function register N	PFNPOE*7	8	8	2 to 3 PCLK*8
0008 C280h	SYSTEM	Deep standby control register	DPSBYCR	8	8	4 to 5 PCLK*8
0008 C281h	SYSTEM	Deep standby wait control register	DPSWCR	8	8	4 to 5 PCLK*8
0008 C282h	SYSTEM	Deep standby interrupt enable register	DPSIER	8	8	4 to 5 PCLK*8
0008 C283h	SYSTEM	Deep standby interrupt flag register	DPSIFR	8	8	4 to 5 PCLK*8
0008 C284h	SYSTEM	Deep standby interrupt edge register	DPSIEGR	8	8	4 to 5 PCLK*8
0008 C285h	SYSTEM	Reset status register	RSTSR	8	8	4 to 5 PCLK*8
0008 C289h	FLASH	Flash write erase protection register	FWEPROR	8	8	4 to 5 PCLK*8
0008 C28Ah	SYSTEM	Sub-clock oscillator control register	SUBOSCCR	8	8	4 to 5 PCLK*8
0008 C28Ch	SYSTEM	Key code register for voltage detection control register	LVDKEYR	8	8	4 to 5 PCLK*8
0008 C28Dh	SYSTEM	Voltage detection control register	LVDCCR	8	8	4 to 5 PCLK*8
0008 C290h	SYSTEM	Deep standby backup register 0	DPSBKR0	8	8	4 to 5 PCLK*8
0008 C291h	SYSTEM	Deep standby backup register 1	DPSBKR1	8	8	4 to 5 PCLK*8
0008 C292h	SYSTEM	Deep standby backup register 2	DPSBKR2	8	8	4 to 5 PCLK*8
0008 C293h	SYSTEM	Deep standby backup register 3	DPSBKR3	8	8	4 to 5 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (28 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 C294h	SYSTEM	Deep standby backup register 4	DPSBKR4	8	8	4 to 5 PCLK*8
0008 C295h	SYSTEM	Deep standby backup register 5	DPSBKR5	8	8	4 to 5 PCLK*8
0008 C296h	SYSTEM	Deep standby backup register 6	DPSBKR6	8	8	4 to 5 PCLK*8
0008 C297h	SYSTEM	Deep standby backup register 7	DPSBKR7	8	8	4 to 5 PCLK*8
0008 C298h	SYSTEM	Deep standby backup register 8	DPSBKR8	8	8	4 to 5 PCLK*8
0008 C299h	SYSTEM	Deep standby backup register 9	DPSBKR9	8	8	4 to 5 PCLK*8
0008 C29Ah	SYSTEM	Deep standby backup register 10	DPSBKR10	8	8	4 to 5 PCLK*8
0008 C29Bh	SYSTEM	Deep standby backup register 11	DPSBKR11	8	8	4 to 5 PCLK*8
0008 C29Ch	SYSTEM	Deep standby backup register 12	DPSBKR12	8	8	4 to 5 PCLK*8
0008 C29Dh	SYSTEM	Deep standby backup register 13	DPSBKR13	8	8	4 to 5 PCLK*8
0008 C29Eh	SYSTEM	Deep standby backup register 14	DPSBKR14	8	8	4 to 5 PCLK*8
0008 C29Fh	SYSTEM	Deep standby backup register 15	DPSBKR15	8	8	4 to 5 PCLK*8
0008 C2A0h	SYSTEM	Deep standby backup register 16	DPSBKR16	8	8	4 to 5 PCLK*8
0008 C2A1h	SYSTEM	Deep standby backup register 17	DPSBKR17	8	8	4 to 5 PCLK*8
0008 C2A2h	SYSTEM	Deep standby backup register 18	DPSBKR18	8	8	4 to 5 PCLK*8
0008 C2A3h	SYSTEM	Deep standby backup register 19	DPSBKR19	8	8	4 to 5 PCLK*8
0008 C2A4h	SYSTEM	Deep standby backup register 20	DPSBKR20	8	8	4 to 5 PCLK*8
0008 C2A5h	SYSTEM	Deep standby backup register 21	DPSBKR21	8	8	4 to 5 PCLK*8
0008 C2A6h	SYSTEM	Deep standby backup register 22	DPSBKR22	8	8	4 to 5 PCLK*8
0008 C2A7h	SYSTEM	Deep standby backup register 23	DPSBKR23	8	8	4 to 5 PCLK*8
0008 C2A8h	SYSTEM	Deep standby backup register 24	DPSBKR24	8	8	4 to 5 PCLK*8
0008 C2A9h	SYSTEM	Deep standby backup register 25	DPSBKR25	8	8	4 to 5 PCLK*8
0008 C2AAh	SYSTEM	Deep standby backup register 26	DPSBKR26	8	8	4 to 5 PCLK*8
0008 C2ABh	SYSTEM	Deep standby backup register 27	DPSBKR27	8	8	4 to 5 PCLK*8
0008 C2ACh	SYSTEM	Deep standby backup register 28	DPSBKR28	8	8	4 to 5 PCLK*8
0008 C2ADh	SYSTEM	Deep standby backup register 29	DPSBKR29	8	8	4 to 5 PCLK*8
0008 C2AEh	SYSTEM	Deep standby backup register 30	DPSBKR30	8	8	4 to 5 PCLK*8
0008 C2AFh	SYSTEM	Deep standby backup register 31	DPSBKR31	8	8	4 to 5 PCLK*8
0008 C400h	RTC	64-Hz counter	R64CNT	8	8	2 to 3 PCLK*8
0008 C402h	RTC	Second counter	RSECNT	8	8	2 to 3 PCLK*8
0008 C404h	RTC	Minute counter	RMINCNT	8	8	2 to 3 PCLK*8
0008 C406h	RTC	Hour counter	RHRCNT	8	8	2 to 3 PCLK*8
0008 C408h	RTC	Day-of-week counter	RWKCNT	8	8	2 to 3 PCLK*8
0008 C40Ah	RTC	Date counter	RDAYCNT	8	8	2 to 3 PCLK*8
0008 C40Ch	RTC	Month counter	RMONCNT	8	8	2 to 3 PCLK*8
0008 C40Eh	RTC	Year counter	RYRCNT	16	16	2 to 3 PCLK*8
0008 C410h	RTC	Second alarm register	RSECAR	8	8	2 to 3 PCLK*8
0008 C412h	RTC	Minute alarm register	RMINAR	8	8	2 to 3 PCLK*8
0008 C414h	RTC	Hour alarm register	RHRAR	8	8	2 to 3 PCLK*8
0008 C416h	RTC	Day-of-week alarm register	RWKAR	8	8	2 to 3 PCLK*8
0008 C418h	RTC	Date alarm register	RDAYAR	8	8	2 to 3 PCLK*8
0008 C41Ah	RTC	Month alarm register	RMONAR	8	8	2 to 3 PCLK*8
0008 C41Ch	RTC	Year alarm register	RYRAR	16	16	2 to 3 PCLK*8
0008 C41Eh	RTC	Year alarm enable register	RYRAREN	8	8	2 to 3 PCLK*8
0008 C422h	RTC	RTC control register 1	RCR1	8	8	2 to 3 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (29 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
0008 C424h	RTC	RTC control register 2	RCR2	8	8	2 to 3 PCLK*8
0009 0200h to 0009 03FFh	CAN0	Mailbox registers 0 to 31	MB0 to MB31	128	8, 16, 32	2 to 3 PCLK*8
0009 0400h	CAN0	Mask register 0	MKR0	32	8, 16, 32	2 to 3 PCLK*8
0009 0404h	CAN0	Mask register 1	MKR1	32	8, 16, 32	2 to 3 PCLK*8
0009 0408h	CAN0	Mask register 2	MKR2	32	8, 16, 32	2 to 3 PCLK*8
0009 040Ch	CAN0	Mask register 3	MKR3	32	8, 16, 32	2 to 3 PCLK*8
0009 0410h	CAN0	Mask register 4	MKR4	32	8, 16, 32	2 to 3 PCLK*8
0009 0414h	CAN0	Mask register 5	MKR5	32	8, 16, 32	2 to 3 PCLK*8
0009 0418h	CAN0	Mask register 6	MKR6	32	8, 16, 32	2 to 3 PCLK*8
0009 041Ch	CAN0	Mask register 7	MKR7	32	8, 16, 32	2 to 3 PCLK*8
0009 0420h	CAN0	FIFO received ID compare register 0	FIDCR0	32	8, 16, 32	2 to 3 PCLK*8
0009 0424h	CAN0	FIFO received ID compare register 1	FIDCR1	32	8, 16, 32	2 to 3 PCLK*8
0009 0428h	CAN0	Mask invalid register	MKIVLR	32	8, 16, 32	2 to 3 PCLK*8
0009 042Ch	CAN0	Mailbox interrupt enable register	MIER	32	8, 16, 32	2 to 3 PCLK*8
0009 0820h to 0009 083Fh	CAN0	Message control registers 0 to 31	MCTL0 to MCTL31	8	8	2 to 3 PCLK*8
0009 0840h	CAN0	Control register	CTLR	16	8, 16	2 to 3 PCLK*8
0009 0842h	CAN0	Status register	STR	16	8, 16	2 to 3 PCLK*8
0009 0844h	CAN0	Bit configuration register	BCR	32	8, 16, 32	2 to 3 PCLK*8
0009 0848h	CAN0	Receive FIFO control register	RFCR	8	8	2 to 3 PCLK*8
0009 0849h	CAN0	Receive FIFO pointer control register	RFPCR	8	8	2 to 3 PCLK*8
0009 084Ah	CAN0	Transmit FIFO control register	TFCR	8	8	2 to 3 PCLK*8
0009 084Bh	CAN0	Transmit FIFO pointer control register	TFPCR	8	8	2 to 3 PCLK*8
0009 084Ch	CAN0	Error interrupt enable register	EIER	8	8	2 to 3 PCLK*8
0009 084Dh	CAN0	Error interrupt factor judge register	EIFR	8	8	2 to 3 PCLK*8
0009 084Eh	CAN0	Receive error count register	RECR	8	8	2 to 3 PCLK*8
0009 084Fh	CAN0	Transmit error count register	TECR	8	8	2 to 3 PCLK*8
0009 0850h	CAN0	Error code store register	ECSR	8	8	2 to 3 PCLK*8
0009 0851h	CAN0	Channel search support register	CSSR	8	8	2 to 3 PCLK*8
0009 0852h	CAN0	Mailbox search status register	MSSR	8	8	2 to 3 PCLK*8
0009 0853h	CAN0	Mailbox search mode register	MSMR	8	8	2 to 3 PCLK*8
0009 0854h	CAN0	Time stamp register	TSR	16	8, 16	2 to 3 PCLK*8
0009 0856h	CAN0	Acceptance filter support register	AFSR	16	8, 16	2 to 3 PCLK*8
0009 0858h	CAN0	Test control register	TCR	8	8	2 to 3 PCLK*8
000A 0000h	USB0	System configuration control register	SYSCFG	16	16	3 to 4 PCLK*8
000A 0004h	USB0	System configuration status register 0	SYSSTS0	16	16	at least 9 PCLK*9
000A 0008h	USB0	Device state control register 0	DVSTCTR0	16	16	at least 9 PCLK*9
000A 0014h	USB0	CFIFO port register	CFIFO	16	8, 16	3 to 4 PCLK*8
000A 0018h	USB0	D0FIFO port register	D0FIFO	16	8, 16	3 to 4 PCLK*8
000A 001Ch	USB0	D1FIFO port register	D1FIFO	16	8, 16	3 to 4 PCLK*8
000A 0020h	USB0	CFIFO port select register	CFIFOSEL	16	16	3 to 4 PCLK*8
000A 0022h	USB0	CFIFO port control register	CFIFOCTR	16	16	3 to 4 PCLK*8
000A 0028h	USB0	D0FIFO port select register	D0FIFOSEL	16	16	3 to 4 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (30 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
000A 002Ah	USB0	D0FIFO port control register	D0FIFOCTR	16	16	3 to 4 PCLK*8
000A 002Ch	USB0	D1FIFO port select register	D1FIFOSEL	16	16	3 to 4 PCLK*8
000A 002Eh	USB0	D1FIFO port control register	D1FIFOCTR	16	16	3 to 4 PCLK*8
000A 0030h	USB0	Interrupt enable register 0	INTENB0	16	16	at least 9 PCLK*9
000A 0032h	USB0	Interrupt enable register 1	INTENB1	16	16	at least 9 PCLK*9
000A 0036h	USB0	BRDY interrupt enable register	BRDYENB	16	16	at least 9 PCLK*9
000A 0038h	USB0	NRDY interrupt enable register	NRDYENB	16	16	at least 9 PCLK*9
000A 003Ah	USB0	BEMP interrupt enable register	BEMPENB	16	16	at least 9 PCLK*9
000A 003Ch	USB0	SOF output configuration register	SOFCFG	16	16	at least 9 PCLK*9
000A 0040h	USB0	Interrupt status register 0	INTSTS0	16	16	at least 9 PCLK*9
000A 0042h	USB0	Interrupt status register 1	INTSTS1	16	16	at least 9 PCLK*9
000A 0046h	USB0	BRDY interrupt status register	BRDYSTS	16	16	at least 9 PCLK*9
000A 0048h	USB0	NRDY interrupt status register	NRDYSTS	16	16	at least 9 PCLK*9
000A 004Ah	USB0	BEMP interrupt status register	BEMPSTS	16	16	at least 9 PCLK*9
000A 004Ch	USB0	Frame number register	FRMNUM	16	16	at least 9 PCLK*9
000A 004Eh	USB0	Device state change register	DVCHGR	16	16	at least 9 PCLK*9
000A 0050h	USB0	USB address register	USBADDR	16	16	at least 9 PCLK*9
000A 0054h	USB0	USB request type register	USBREQ	16	16	at least 9 PCLK*9
000A 0056h	USB0	USB request value register	USBVAL	16	16	at least 9 PCLK*9
000A 0058h	USB0	USB request index register	USBINDX	16	16	at least 9 PCLK*9
000A 005Ah	USB0	USB request length register	USBLENG	16	16	at least 9 PCLK*9
000A 005Ch	USB0	DCP configuration register	DCPCFG	16	16	at least 9 PCLK*9
000A 005Eh	USB0	DCP maximum packet size register	DCPMAXP	16	16	at least 9 PCLK*9
000A 0060h	USB0	DCP control register	DCPCTR	16	16	at least 9 PCLK*9
000A 0064h	USB0	Pipe window select register	PIPESEL	16	16	at least 9 PCLK*9
000A 0068h	USB0	Pipe configuration register	PIPECFG	16	16	at least 9 PCLK*9
000A 006Ch	USB0	Pipe maximum packet size register	PIPEMAXP	16	16	at least 9 PCLK*9

Table 5.1 List of I/O Registers (Address Order) (31 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
000A 006Eh	USB0	Pipe cycle control register	PIPEPERI	16	16	at least 9 PCLK*9
000A 0070h	USB0	Pipe 1 control register	PIPE1CTR	16	16	at least 9 PCLK*9
000A 0072h	USB0	Pipe 2 control register	PIPE2CTR	16	16	at least 9 PCLK*9
000A 0074h	USB0	Pipe 3 control register	PIPE3CTR	16	16	at least 9 PCLK*9
000A 0076h	USB0	Pipe 4 control register	PIPE4CTR	16	16	at least 9 PCLK*9
000A 0078h	USB0	Pipe 5 control register	PIPE5CTR	16	16	at least 9 PCLK*9
000A 007Ah	USB0	Pipe 6 control register	PIPE6CTR	16	16	at least 9 PCLK*9
000A 007Ch	USB0	Pipe 7 control register	PIPE7CTR	16	16	at least 9 PCLK*9
000A 007Eh	USB0	Pipe 8 control register	PIPE8CTR	16	16	at least 9 PCLK*9
000A 0080h	USB0	Pipe 9 control register	PIPE9CTR	16	16	at least 9 PCLK*9
000A 0090h	USB0	Pipe 1 transaction counter enable register	PIPE1TRE	16	16	at least 9 PCLK*9
000A 0092h	USB0	Pipe 1 transaction counter register	PIPE1TRN	16	16	at least 9 PCLK*9
000A 0094h	USB0	Pipe 2 transaction counter enable register	PIPE2TRE	16	16	at least 9 PCLK*9
000A 0096h	USB0	Pipe 2 transaction counter register	PIPE2TRN	16	16	at least 9 PCLK*9
000A 0098h	USB0	Pipe 3 transaction counter enable register	PIPE3TRE	16	16	at least 9 PCLK*9
000A 009Ah	USB0	Pipe 3 transaction counter register	PIPE3TRN	16	16	at least 9 PCLK*9
000A 009Ch	USB0	Pipe 4 transaction counter enable register	PIPE4TRE	16	16	at least 9 PCLK*9
000A 009Eh	USB0	Pipe 4 transaction counter register	PIPE4TRN	16	16	at least 9 PCLK*9
000A 00A0h	USB0	Pipe 5 transaction counter enable register	PIPE5TRE	16	16	at least 9 PCLK*9
000A 00A2h	USB0	Pipe 5 transaction counter register	PIPE5TRN	16	16	at least 9 PCLK*9
000A 00D0h	USB0	Device address 0 configuration register	DEVADD0	16	16	at least 9 PCLK*9
000A 00D2h	USB0	Device address 1 configuration register	DEVADD1	16	16	at least 9 PCLK*9
000A 00D4h	USB0	Device address 2 configuration register	DEVADD2	16	16	at least 9 PCLK*9
000A 00D6h	USB0	Device address 3 configuration register	DEVADD3	16	16	at least 9 PCLK*9
000A 00D8h	USB0	Device address 4 configuration register	DEVADD4	16	16	at least 9 PCLK*9
000A 00DAh	USB0	Device address 5 configuration register	DEVADD5	16	16	at least 9 PCLK*9
000A 0200h	USB1	System configuration control register	SYSCFG	16	16	3 to 4 PCLK*8

Table 5.1 List of I/O Registers (Address Order) (32 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
000A 0204h	USB1	System configuration status register 0	SYSSTS0	16	16	at least 9 PCLK* ⁹
000A 0208h	USB1	Device state control register 0	DVSTCTR0	16	16	at least 9 PCLK* ⁹
000A 0214h	USB1	CFIFO port register	CFIFO	16	8, 16	3 to 4 PCLK* ⁸
000A 0218h	USB1	D0FIFO port register	D0FIFO	16	8, 16	3 to 4 PCLK* ⁸
000A 021Ch	USB1	D1FIFO port register	D1FIFO	16	8, 16	3 to 4 PCLK* ⁸
000A 0220h	USB1	CFIFO port select register	CFIFOSEL	16	16	3 to 4 PCLK* ⁸
000A 0222h	USB1	CFIFO port control register	CFIFOCTR	16	16	3 to 4 PCLK* ⁸
000A 0228h	USB1	D0FIFO port select register	D0FIFOSEL	16	16	3 to 4 PCLK* ⁸
000A 022Ah	USB1	D0FIFO port control register	D0FIFOCTR	16	16	3 to 4 PCLK* ⁸
000A 022Ch	USB1	D1FIFO port select register	D1FIFOSEL	16	16	3 to 4 PCLK* ⁸
000A 022Eh	USB1	D1FIFO port control register	D1FIFOCTR	16	16	3 to 4 PCLK* ⁸
000A 0230h	USB1	Interrupt enable register 0	INTENB0	16	16	at least 9 PCLK* ⁹
000A 0232h	USB1	Interrupt enable register 1	INTENB1	16	16	at least 9 PCLK* ⁹
000A 0236h	USB1	BRDY interrupt enable register	BRDYENB	16	16	at least 9 PCLK* ⁹
000A 0238h	USB1	NRDY interrupt enable register	NRDYENB	16	16	at least 9 PCLK* ⁹
000A 023Ah	USB1	BEMP interrupt enable register	BEMPENB	16	16	at least 9 PCLK* ⁹
000A 023Ch	USB1	SOF output configuration register	SOFCFG	16	16	at least 9 PCLK* ⁹
000A 0240h	USB1	Interrupt status register 0	INTSTS0	16	16	at least 9 PCLK* ⁹
000A 0242h	USB1	Interrupt status register 1	INTSTS1	16	16	at least 9 PCLK* ⁹
000A 0246h	USB1	BRDY interrupt status register	BRDYSTS	16	16	at least 9 PCLK* ⁹
000A 0248h	USB1	NRDY interrupt status register	NRDYSTS	16	16	at least 9 PCLK* ⁹
000A 024Ah	USB1	BEMP interrupt status register	BEMPSTS	16	16	at least 9 PCLK* ⁹
000A 024Ch	USB1	Frame number register	FRMNUM	16	16	at least 9 PCLK* ⁹
000A 024Eh	USB1	Device state change register	DVCHGR	16	16	at least 9 PCLK* ⁹
000A 0250h	USB1	USB address register	USBADDR	16	16	at least 9 PCLK* ⁹
000A 0254h	USB1	USB request type register	USBREQ	16	16	at least 9 PCLK* ⁹
000A 0256h	USB1	USB request value register	USBVAL	16	16	at least 9 PCLK* ⁹
000A 0258h	USB1	USB request index register	USBINDX	16	16	at least 9 PCLK* ⁹
000A 025Ah	USB1	USB request length register	USBLENG	16	16	at least 9 PCLK* ⁹
000A 025Ch	USB1	DCP configuration register	DCPCFG	16	16	at least 9 PCLK* ⁹

Table 5.1 List of I/O Registers (Address Order) (33 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
000A 025Eh	USB1	DCP maximum packet size register	DCPMAXP	16	16	at least 9 PCLK*9
000A 0260h	USB1	DCP control register	DCPCTR	16	16	at least 9 PCLK*9
000A 0264h	USB1	Pipe window select register	PIPESEL	16	16	at least 9 PCLK*9
000A 0268h	USB1	Pipe configuration register	PIPECFG	16	16	at least 9 PCLK*9
000A 026Ch	USB1	Pipe maximum packet size register	PIPEMAXP	16	16	at least 9 PCLK*9
000A 026Eh	USB1	Pipe cycle control register	PIPEPERI	16	16	at least 9 PCLK*9
000A 0270h	USB1	Pipe 1 control register	PIPE1CTR	16	16	at least 9 PCLK*9
000A 0272h	USB1	Pipe 2 control register	PIPE2CTR	16	16	at least 9 PCLK*9
000A 0274h	USB1	Pipe 3 control register	PIPE3CTR	16	16	at least 9 PCLK*9
000A 0276h	USB1	Pipe 4 control register	PIPE4CTR	16	16	at least 9 PCLK*9
000A 0278h	USB1	Pipe 5 control register	PIPE5CTR	16	16	at least 9 PCLK*9
000A 027Ah	USB1	Pipe 6 control register	PIPE6CTR	16	16	at least 9 PCLK*9
000A 027Ch	USB1	Pipe 7 control register	PIPE7CTR	16	16	at least 9 PCLK*9
000A 027Eh	USB1	Pipe 8 control register	PIPE8CTR	16	16	at least 9 PCLK*9
000A 0280h	USB1	Pipe 9 control register	PIPE9CTR	16	16	at least 9 PCLK*9
000A 0290h	USB1	Pipe 1 transaction counter enable register	PIPE1TRE	16	16	at least 9 PCLK*9
000A 0292h	USB1	Pipe 1 transaction counter register	PIPE1TRN	16	16	at least 9 PCLK*9
000A 0294h	USB1	Pipe 2 transaction counter enable register	PIPE2TRE	16	16	at least 9 PCLK*9
000A 0296h	USB1	Pipe 2 transaction counter register	PIPE2TRN	16	16	at least 9 PCLK*9
000A 0298h	USB1	Pipe 3 transaction counter enable register	PIPE3TRE	16	16	at least 9 PCLK*9
000A 029Ah	USB1	Pipe 3 transaction counter register	PIPE3TRN	16	16	at least 9 PCLK*9
000A 029Ch	USB1	Pipe 4 transaction counter enable register	PIPE4TRE	16	16	at least 9 PCLK*9
000A 029Eh	USB1	Pipe 4 transaction counter register	PIPE4TRN	16	16	at least 9 PCLK*9
000A 02A0h	USB1	Pipe 5 transaction counter enable register	PIPE5TRE	16	16	at least 9 PCLK*9
000A 02A2h	USB1	Pipe 5 transaction counter register	PIPE5TRN	16	16	at least 9 PCLK*9
000A 02D0h	USB1	Device address 0 configuration register	DEVADD0	16	16	at least 9 PCLK*9

Table 5.1 List of I/O Registers (Address Order) (34 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
000A 02D2h	USB1	Device address 1 configuration register	DEVADD1	16	16	at least 9 PCLK*9
000A 02D4h	USB1	Device address 2 configuration register	DEVADD2	16	16	at least 9 PCLK*9
000A 02D6h	USB1	Device address 3 configuration register	DEVADD3	16	16	at least 9 PCLK*9
000A 02D8h	USB1	Device address 4 configuration register	DEVADD4	16	16	at least 9 PCLK*9
000A 02DAh	USB1	Device address 5 configuration register	DEVADD5	16	16	at least 9 PCLK*9
000A 0400h	USB	Deep standby USB transceiver control/pin monitor register	DPUSR0R	32	32	1 to 2PCLK*8
000A 0404h	USB	Deep standby USB suspend/resume interrupt register	DPUSR1R	32	32	1 to 2PCLK*8
000C 0000h	EDMAC	EDMAC mode register	EDMR	32	32	4 to 5 ICLK
000C 0008h	EDMAC	EDMAC transmit request register	EDTRR	32	32	4 to 5 ICLK
000C 0010h	EDMAC	EDMAC receive request register	EDRRR	32	32	4 to 5 ICLK
000C 0018h	EDMAC	Transmit descriptor list start address register	TDLAR	32	32	4 to 5 ICLK
000C 0020h	EDMAC	Receive descriptor list start address register	RDLAR	32	32	4 to 5 ICLK
000C 0028h	EDMAC	ETHERC/EDMAC status register	EESR	32	32	4 to 5 ICLK
000C 0030h	EDMAC	ETHERC/EDMAC status interrupt permission register	EESIPR	32	32	4 to 5 ICLK
000C 0038h	EDMAC	Transmit/receive status copy enable register	TRSCER	32	32	4 to 5 ICLK
000C 0040h	EDMAC	Receive missed-frame counter register	RMFCR	32	32	4 to 5 ICLK
000C 0048h	EDMAC	Transmit FIFO threshold register	TFTR	32	32	4 to 5 ICLK
000C 0050h	EDMAC	FIFO depth register	FDR	32	32	4 to 5 ICLK
000C 0058h	EDMAC	Receiving method control register	RMCR	32	32	4 to 5 ICLK
000C 0064h	EDMAC	Transmit FIFO underrun counter	TFUCR	32	32	4 to 5 ICLK
000C 0068h	EDMAC	Receive FIFO overflow counter	RFOCR	32	32	4 to 5 ICLK
000C 006Ch	EDMAC	Independent output signal setting register	IOSR	32	32	4 to 5 ICLK
000C 0070h	EDMAC	Flow control start FIFO threshold setting register	FCFTR	32	32	4 to 5 ICLK
000C 0078h	EDMAC	Receive data padding insert register	RPADIR	32	32	4 to 5 ICLK
000C 007Ch	EDMAC	Transmit interrupt setting register	TRIMD	32	32	4 to 5 ICLK
000C 00C8h	EDMAC	Receive buffer write address register	RBWAR	32	32	4 to 5 ICLK
000C 00CCh	EDMAC	Receive descriptor fetch address register	RDFAR	32	32	4 to 5 ICLK
000C 00D4h	EDMAC	Transmit buffer read address register	TBRAR	32	32	4 to 5 ICLK
000C 00D8h	EDMAC	Transmit descriptor fetch address register	TDFAR	32	32	4 to 5 ICLK
000C 0100h	ETHERC	ETHERC mode register	ECMR	32	32	4 to 5 ICLK
000C 0108h	ETHERC	Receive frame length register	RFLR	32	32	4 to 5 ICLK
000C 0110h	ETHERC	ETHERC status register	ECSR	32	32	4 to 5 ICLK
000C 0118h	ETHERC	ETHERC interrupt enable register	ECSIPR	32	32	4 to 5 ICLK
000C 0120h	ETHERC	PHY interface register	PIR	32	32	4 to 5 ICLK
000C 0128h	ETHERC	PHY status register	PSR	32	32	4 to 5 ICLK

Table 5.1 List of I/O Registers (Address Order) (35 / 36)

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
000C 0140h	ETHERC	Random number generation counter upper limit setting register	RDMLR	32	32	4 to 5 ICLK
000C 0150h	ETHERC	IPG register	IPGR	32	32	4 to 5 ICLK
000C 0154h	ETHERC	Automatic PAUSE frame register	APR	32	32	4 to 5 ICLK
000C 0158h	ETHERC	Manual PAUSE frame register	MPR	32	32	4 to 5 ICLK
000C 0160h	ETHERC	PAUSE frame receive counter register	RFCF	32	32	4 to 5 ICLK
000C 0164h	ETHERC	Automatic PAUSE frame retransmit count register	TPAUSER	32	32	4 to 5 ICLK
000C 0168h	ETHERC	PAUSE frame retransmit counter register	TPAUSECR	32	32	4 to 5 ICLK
000C 016Ch	ETHERC	Broadcast frame receive count setting register	BCFRR	32	32	4 to 5 ICLK
000C 01C0h	ETHERC	MAC address high register	MAHR	32	32	4 to 5 ICLK
000C 01C8h	ETHERC	MAC address low register	MALR	32	32	4 to 5 ICLK
000C 01D0h	ETHERC	Transmit retry over counter register	TROCR	32	32	4 to 5 ICLK
000C 01D4h	ETHERC	Delayed collision detect counter register	CDCR	32	32	4 to 5 ICLK
000C 01D8h	ETHERC	Lost carrier counter register	LCCR	32	32	4 to 5 ICLK
000C 01DCh	ETHERC	Carrier not detect counter register	CNDCR	32	32	4 to 5 ICLK
000C 01E4h	ETHERC	CRC error frame receive counter register	CEFCR	32	32	4 to 5 ICLK
000C 01E8h	ETHERC	Frame receive error counter register	FRECR	32	32	4 to 5 ICLK
000C 01ECh	ETHERC	Too-short frame receive counter register	TSFRCR	32	32	4 to 5 ICLK
000C 01F0h	ETHERC	Too-long frame receive counter register	TLFRCR	32	32	4 to 5 ICLK
000C 01F4h	ETHERC	Residual-bit frame receive counter register	RFCR	32	32	4 to 5 ICLK
000C 01F8h	ETHERC	Multicast address frame receive counter register	MAFCR	32	32	4 to 5 ICLK
007F C402h	FLASH	Flash mode register	FMODR	8	8	2 to 3 PCLK*8
007F C410h	FLASH	Flash access status register	FASTAT	8	8	2 to 3 PCLK*8
007F C411h	FLASH	Flash access error interrupt enable register	FAEINT	8	8	2 to 3 PCLK*8
007F C412h	FLASH	Flash ready interrupt enable register	FRDYIE	8	8	2 to 3 PCLK*8
007F C440h	FLASH	Data flash read enable register0	DFLRE0	16	16	2 to 3 PCLK*8
007F C442h	FLASH	Data flash read enable register1	DFLRE1	16	16	2 to 3 PCLK*8
007F C450h	FLASH	Data flash programming/erasure enable register0	DFLWE0	16	16	2 to 3 PCLK*8
007F C452h	FLASH	Data flash programming/erasure enable register1	DFLWE1	16	16	2 to 3 PCLK*8
007F C454h	FLASH	FCU RAM enable register	FCURAME	16	16	2 to 3 PCLK*8
007F FFB0h	FLASH	Flash status register 0	FSTATR0	8	8	2 to 3 PCLK*8
007F FFB1h	FLASH	Flash status register 1	FSTATR1	8	8	2 to 3 PCLK*8
007F FFB2h	FLASH	Flash P/E mode entry register	FENTRYR	16	16	2 to 3 PCLK*8
007F FFB4h	FLASH	Flash protect register	FPROTR	16	16	2 to 3 PCLK*8
007F FFB6h	FLASH	Flash reset register	FRESETR	16	16	2 to 3 PCLK*8
007F FFBAh	FLASH	FCU command register	FCMDR	16	16	2 to 3 PCLK*8
007F FFC8h	FLASH	FCU processing switching register	FCPSR	16	16	2 to 3 PCLK*8
007F FFCAh	FLASH	Data flash blank check control register	DFLBCCNT	16	16	2 to 3 PCLK*8
007F FFCCh	FLASH	Flash P/E status register	FPESTAT	16	16	2 to 3 PCLK*8
007F FFCEh	FLASH	Data flash blank check status register	DFLBCSTAT	16	16	2 to 3 PCLK*8

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

Address	Module Abbreviation	Register Name	Register Abbreviation	Number of Bits	Access Size	Number of Access Cycles
007F FFE8h	FLASH	Peripheral clock notification register	PCKAR	16	16	2 to 3 PCLK* ⁸

Note 1. When the same output trigger is specified for pulse output groups 2 and 3 by the PPG0.PCR setting, the PPG0.NDRH address is 000881ECh. When different output triggers are specified, the PPG0.NDRH2 addresses for pulse output groups 2 and 3 are 000881EEh and 000881ECh, respectively.

Note 2. When the same output trigger is specified for pulse output groups 0 and 1 by the PPG0.PCR setting, the PPG0.NDRL address is 000881EDh. When different output triggers are specified, the PPG0.NDRL2 addresses for pulse output groups 0 and 1 are 000881EFh and 000881EDh, respectively.

Note 3. When the same output trigger is specified for pulse output groups 6 and 7 by the PPG1.PCR setting, the PPG1.NDRH address is 000881FCh. When different output triggers are specified, the PPG1.NDRH2 addresses for pulse output groups 6 and 7 are 000881FEh and 000881FCh, respectively.

Note 4. When the same output trigger is specified for pulse output groups 4 and 5 by the PPG1.PCR setting, the PPG1.NDRL address is 000881FDh. When different output triggers are specified, the PPG1.NDRL2 addresses for pulse output groups 4 and 5 are 000881FFh and 000881FDh, respectively.

Note 5. This register is not supported by the 145-pin TFLGA or 144-pin LQFP version.

Note 6. This register is not supported by the 100-pin LQFP version.

Note 7. This register is not supported by the 85-pin TFLGA version.

Note 8. The number of access states depends on the number of divided cycles for clock synchronization (0 to 1 PCLK, 0 to 1 BCLK).

Note 9. Access may be disabled if a register is accessed during the USB operation.

5.2 I/O Register Bits

Register addresses and bit names of the peripheral modules are described below.

Each line cover eight bits, and 16-bit and 32-bit registers are shown as 2 or 4 lines, respectively.

Table 5.2 List of I/O Registers (Bit Order) (1 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
SYSTEM	MDMONR	—	—	—	—	—	—	—	—
		MDE	—	—	—	—	—	MD1	MD0
SYSTEM	MDSR	—	—	—	—	—	—	—	—
		—	UBTS	—	BOTS	BSW[1:0]		EXB	IROM
SYSTEM	SYSCR0	KEY[7:0]							
		—	—	—	—	—	—	EXBE	ROME
SYSTEM	SYSCR1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	RAME
SYSTEM	SBYCR	SSBY	OPE	—	STS[4:0]				
		—	—	—	—	—	—	—	—
SYSTEM	MSTPCRA	ACSE	—	MSTPA29	MSTPA28	—	—	—	—
		MSTPA23	MSTPA22	—	—	MSTPA19	—	MSTPA17	—
		MSTPA15	MSTPA14	—	—	MSTPA11	MSTPA10	MSTPA9	MSTPA8
		—	—	MSTPA5	MSTPA4	—	—	—	—
SYSTEM	MSTPCRB	MSTPB31	MSTPB30	MSTPB29	MSTPB28	—	MSTPB26	MSTPB25	—
		MSTPB23	—	MSTPB21	MSTPB20	MSTPB19	MSTPB18	MSTPB17	MSTPB16
		MSTPB15	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	MSTPB0
SYSTEM	MSTPCRC	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	MSTPC1	MSTPC0
SYSTEM	SCKCR	—	—	—	—	ICK[3:0]			
		PSTOP1	PSTOP0	—	—	BCK[3:0]			
		—	—	—	—	PCK[3:0]			
		—	—	—	—	—	—	—	—
SYSTEM	BCKCR	—	—	—	—	—	—	—	BCLKDIV
SYSTEM	OSTDCR	KEY[7:0]							
		OSTDE	OSTDF	—	—	—	—	—	—
BSC	BERCLR	—	—	—	—	—	—	—	STSCCLR
BSC	BEREN	—	—	—	—	—	—	TOEN	IGAEN
BSC	BERSR1	—	MST[2:0]			—	—	TO	IA
BSC	BERSR2	ADDR[12:0]							
		ADDR[12:0]						—	—
DMAC0	DMSAR								

Table 5.2 List of I/O Registers (Bit Order) (2 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
DMAC0	DMDAR								
DMAC0	DMCRA	—	—	—	—	—	—		
DMAC0	DMCRB	—	—	—	—	—	—		
DMAC0	DMTMD	MD[1:0]		DTS[1:0]		—	—	SZ[1:0]	
		—	—	—	—	—	—	DCTG[1:0]	
DMAC0	DMINT	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
DMAC0	DMAMD	SM[1:0]		—	SARA[4:0]				
		DM[1:0]		—	DARA[4:0]				
DMAC0	DMOFR								
DMAC0	DMCNT	—	—	—	—	—	—	—	DTE
DMAC0	DMREQ	—	—	—	CLRS	—	—	—	SWREQ
DMAC0	DMSTS	ACT	—	—	DTIF	—	—	—	ESIF
DMAC0	DMCSL	—	—	—	—	—	—	—	DISEL
DMAC1	DMSAR								
DMAC1	DMDAR								
DMAC1	DMCRA	—	—	—	—	—	—		
DMAC1	DMCRB	—	—	—	—	—	—		
DMAC1	DMTMD	MD[1:0]		DTS[1:0]		—	—	SZ[1:0]	
		—	—	—	—	—	—	DCTG[1:0]	
DMAC1	DMINT	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
DMAC1	DMAMD	SM[1:0]		—	SARA[4:0]				
		DM[1:0]		—	DARA[4:0]				
DMAC1	DMCNT	—	—	—	—	—	—	—	DTE

Table 5.2 List of I/O Registers (Bit Order) (3 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
DMAC1	DMREQ	—	—	—	CLRS	—	—	—	SWREQ
DMAC1	DMSTS	ACT	—	—	DTIF	—	—	—	ESIF
DMAC1	DMCSL	—	—	—	—	—	—	—	DISEL
DMAC2	DMSAR								
DMAC2	DMDAR								
DMAC2	DMCRA	—	—	—	—	—	—		
DMAC2	DMCRB	—	—	—	—	—	—		
DMAC2	DMTMD	MD[1:0]		DTS[1:0]		—	—	SZ[1:0]	
		—	—	—	—	—	—	DCTG[1:0]	
DMAC2	DMINT	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
DMAC2	DMAMD	SM[1:0]		—	SARA[4:0]				
		DM[1:0]		—	DARA[4:0]				
DMAC2	DMCNT	—	—	—	—	—	—	—	DTE
DMAC2	DMREQ	—	—	—	CLRS	—	—	—	SWREQ
DMAC2	DMSTS	ACT	—	—	DTIF	—	—	—	ESIF
DMAC2	DMCSL	—	—	—	—	—	—	—	DISEL
DMAC3	DMSAR								
DMAC3	DMDAR								
DMAC3	DMCRA	—	—	—	—	—	—		
DMAC3	DMCRB	—	—	—	—	—	—		
DMAC3	DMTMD	MD[1:0]		DTS[1:0]		—	—	SZ[1:0]	
		—	—	—	—	—	—	DCTG[1:0]	
DMAC3	DMINT	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE

Table 5.2 List of I/O Registers (Bit Order) (4 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
DMAC3	DMAMD	SM[1:0]		—	SARA[4:0]				
		DM[1:0]		—	DARA[4:0]				
DMAC3	DMCNT	—	—	—	—	—	—	—	DTE
DMAC3	DMREQ	—	—	—	CLRS	—	—	—	SWREQ
DMAC3	DMSTS	ACT	—	—	DTIF	—	—	—	ESIF
DMAC3	DMCSL	—	—	—	—	—	—	—	DISEL
DMAC	DMAST	—	—	—	—	—	—	—	DMST
DTC	DTCCR	—	—	—	RRS	—	—	—	—
DTC	DTCVBR								
DTC	DTCADMOD	—	—	—	—	—	—	—	SHORT
DTC	DTCST	—	—	—	—	—	—	—	DTCST
DTC	DTCSTS	ACT	—	—	—	—	—	—	—
		VECN[7:0]							
EXDMAC0	EDMSAR								
EXDMAC0	EDMDAR								
EXDMAC0	EDMCRA	—	—	—	—	—	—		
EXDMAC0	EDMCRB	—	—	—	—	—	—		
EXDMAC0	EDMTMD	MD[1:0]		DTS[1:0]		—	—	SZ[1:0]	
		—	—	—	—	—	—	DCTG[1:0]	
EXDMAC0	EDMOMD	—	—	—	—	DACKS	DACKE	DACKW	—
EXDMAC0	EDMINT	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
EXDMAC0	EDMAMD	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	AMS	DIR
		SM[1:0]		—	SARA[4:0]				
		DM[1:0]		—	DARA[4:0]				
EXDMAC0	EDMOFR								
EXDMAC0	EDMCNT	—	—	—	—	—	—	—	DTE
EXDMAC0	EDMREQ	—	—	—	CLRS	—	—	—	SWREQ
EXDMAC0	EDMSTS	ACT	—	—	DTIF	—	—	—	ESIF

Table 5.2 List of I/O Registers (Bit Order) (5 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
EXDMAC0	EDMRMD	—	—	—	—	—	—	DREQS[1:0]	
EXDMAC0	EDMERF	—	—	—	—	—	—	—	EREQ
EXDMAC0	EDMPRF	—	—	—	—	—	—	—	PREQ
EXDMAC1	EDMSAR								
EXDMAC1	EDMDAR								
EXDMAC1	EDMCRA	—	—	—	—	—	—		
EXDMAC1	EDMCRB	—	—	—	—	—	—		
EXDMAC1	EDMTMD	MD[1:0]		DTS[1:0]		—	—	SZ[1:0]	
		—	—	—	—	—	—	DCTG[1:0]	
EXDMAC1	EDMOMD	—	—	—	—	DACKS	DACKE	DACKW	—
EXDMAC1	EDMINT	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
EXDMAC1	EDMAMD	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	AMS	DIR
		SM[1:0]		—	SARA[4:0]				
		DM[1:0]		—	DARA[4:0]				
EXDMAC1	EDMCNT	—	—	—	—	—	—	—	DTE
EXDMAC1	EDMREQ	—	—	—	CLRS	—	—	—	SWREQ
EXDMAC1	EDMSTS	ACT	—	—	DTIF	—	—	—	ESIF
EXDMAC1	EDMRMD	—	—	—	—	—	—	DREQS[1:0]	
EXDMAC1	EDMERF	—	—	—	—	—	—	—	EREQ
EXDMAC1	EDMPRF	—	—	—	—	—	—	—	PREQ
EXDMAC	EDMAST	—	—	—	—	—	—	—	DMST
EXDMAC	CLSBR0								
EXDMAC	CLSBR1								
EXDMAC	CLSBR2								

Table 5.2 List of I/O Registers (Bit Order) (6 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	
EXDMAC	CLSBR3									
EXDMAC	CLSBR4									
EXDMAC	CLSBR5									
EXDMAC	CLSBR6									
BSC	CS0MOD	PRMOD	—	—	—	—	—	PWENB	PRENB	
		—	—	—	—	EWENB	—	—	WRMOD	
BSC	CS0WCR1	—	—	—	CSRWAIT[4:0]					
		—	—	—	CSWWAIT[4:0]					
		—	—	—	—	—	CSPRWAIT[2:0]			
		—	—	—	—	—	CSPWWAIT[2:0]			
BSC	CS0WCR2	—	CSON[2:0]			—	WDON[2:0]			
		—	WRON[2:0]			—	RDON[2:0]			
		—	—	—	—	—	WDOFF[2:0]			
		—	CSWOFF[2:0]			—	CSROFF[2:0]			
BSC	CS1MOD	PRMOD	—	—	—	—	—	PWENB	PRENB	
		—	—	—	—	EWENB	—	—	WRMOD	
BSC	CS1WCR1	—	—	—	CSRWAIT[4:0]					
		—	—	—	CSWWAIT[4:0]					
		—	—	—	—	—	CSPRWAIT[2:0]			
		—	—	—	—	—	CSPWWAIT[2:0]			
BSC	CS1WCR2	—	CSON[2:0]			—	WDON[2:0]			
		—	WRON[2:0]			—	RDON[2:0]			
		—	—	—	—	—	WDOFF[2:0]			
		—	CSWOFF[2:0]			—	CSROFF[2:0]			
BSC	CS2MOD	PRMOD	—	—	—	—	—	PWENB	PRENB	
		—	—	—	—	EWENB	—	—	WRMOD	
BSC	CS2WCR1	—	—	—	CSRWAIT[4:0]					
		—	—	—	CSWWAIT[4:0]					
		—	—	—	—	—	CSPRWAIT[2:0]			
		—	—	—	—	—	CSPWWAIT[2:0]			

Table 5.2 List of I/O Registers (Bit Order) (7 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
BSC	CS2WCR2	—	CSON[2:0]			—	WDON[2:0]		
		—	WRON[2:0]			—	RDON[2:0]		
		—	—	—	—	—	WDOFF[2:0]		
		—	CSWOFF[2:0]			—	CSROFF[2:0]		
BSC	CS3MOD	PRMOD	—	—	—	—	—	PWENB	PRENB
		—	—	—	—	EWENB	—	—	WRMOD
BSC	CS3WCR1	—	—	—	CSRWAIT[4:0]				
		—	—	—	CSWWAIT[4:0]				
		—	—	—	—	—	CSPRWAIT[2:0]		
		—	—	—	—	—	CSPWWAIT[2:0]		
BSC	CS3WCR2	—	CSON[2:0]			—	WDON[2:0]		
		—	WRON[2:0]			—	RDON[2:0]		
		—	—	—	—	—	WDOFF[2:0]		
		—	CSWOFF[2:0]			—	CSROFF[2:0]		
BSC	CS4MOD	PRMOD	—	—	—	—	—	PWENB	PRENB
		—	—	—	—	EWENB	—	—	WRMOD
BSC	CS4WCR1	—	—	—	CSRWAIT[4:0]				
		—	—	—	CSWWAIT[4:0]				
		—	—	—	—	—	CSPRWAIT[2:0]		
		—	—	—	—	—	CSPWWAIT[2:0]		
BSC	CS4WCR2	—	CSON[2:0]			—	WDON[2:0]		
		—	WRON[2:0]			—	RDON[2:0]		
		—	—	—	—	—	WDOFF[2:0]		
		—	CSWOFF[2:0]			—	CSROFF[2:0]		
BSC	CS5MOD	PRMOD	—	—	—	—	—	PWENB	PRENB
		—	—	—	—	EWENB	—	—	WRMOD
BSC	CS5WCR1	—	—	—	CSRWAIT[4:0]				
		—	—	—	CSWWAIT[4:0]				
		—	—	—	—	—	CSPRWAIT[2:0]		
		—	—	—	—	—	CSPWWAIT[2:0]		
BSC	CS5WCR2	—	CSON[2:0]			—	WDON[2:0]		
		—	WRON[2:0]			—	RDON[2:0]		
		—	—	—	—	—	WDOFF[2:0]		
		—	CSWOFF[2:0]			—	CSROFF[2:0]		
BSC	CS6MOD	PRMOD	—	—	—	—	—	PWENB	PRENB
		—	—	—	—	EWENB	—	—	WRMOD
BSC	CS6WCR1	—	—	—	CSRWAIT[4:0]				
		—	—	—	CSWWAIT[4:0]				
		—	—	—	—	—	CSPRWAIT[2:0]		
		—	—	—	—	—	CSPWWAIT[2:0]		
BSC	CS6WCR2	—	CSON[2:0]			—	WDON[2:0]		
		—	WRON[2:0]			—	RDON[2:0]		
		—	—	—	—	—	WDOFF[2:0]		
		—	CSWOFF[2:0]			—	CSROFF[2:0]		

Table 5.2 List of I/O Registers (Bit Order) (8 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
BSC	CS7MOD	PRMOD	—	—	—	—	—	PWENB	PRENB
		—	—	—	—	EWENB	—	—	WRMOD
BSC	CS7WCR1	—	—	—	CSRWAIT[4:0]				
		—	—	—	CSWWAIT[4:0]				
		—	—	—	—	CSPRWAIT[2:0]			
		—	—	—	—	CSPWWAIT[2:0]			
BSC	CS7WCR2	—	CSON[2:0]			—	WDON[2:0]		
		—	WRON[2:0]			—	RDON[2:0]		
		—	—	—	—	WDOFF[2:0]			
		—	CSWOFF[2:0]			—	CSROFF[2:0]		
BSC	CS0CR	—	—	—	—	—	—	—	EMODE
		—	—	BSIZE[1:0]			—	—	—
BSC	CS0REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
BSC	CS1CR	—	—	—	—	—	—	—	EMODE
		—	—	BSIZE[1:0]			—	—	—
BSC	CS1REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
BSC	CS2CR	—	—	—	—	—	—	—	EMODE
		—	—	BSIZE[1:0]			—	—	—
BSC	CS2REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
BSC	CS3CR	—	—	—	—	—	—	—	EMODE
		—	—	BSIZE[1:0]			—	—	—
BSC	CS3REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
BSC	CS4CR	—	—	—	—	—	—	—	EMODE
		—	—	BSIZE[1:0]			—	—	—
BSC	CS4REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
BSC	CS5CR	—	—	—	—	—	—	—	EMODE
		—	—	BSIZE[1:0]			—	—	—
BSC	CS5REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
BSC	CS6CR	—	—	—	—	—	—	—	EMODE
		—	—	BSIZE[1:0]			—	—	—
BSC	CS6REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
BSC	CS7CR	—	—	—	—	—	—	—	EMODE
		—	—	BSIZE[1:0]			—	—	—
BSC	CS7REC	—	—	—	—	WRCV[3:0]			
		—	—	—	—	RRCV[3:0]			
BSC	SDCCR	—	—	BSIZE[1:0]			—	—	EXENB
BSC	SDCMOD	—	—	—	—	—	—	—	EMODE
BSC	SDAMOD	—	—	—	—	—	—	—	BE

Table 5.2 List of I/O Registers (Bit Order) (9 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
BSC	SDSELF	—	—	—	—	—	—	—	SFEN
BSC	SDRFCR	REFW[3:0]				RFC[11:0]			
		RFC[11:0]							
BSC	SDRFEN	—	—	—	—	—	—	—	RFEN
BSC	SDICR	—	—	—	—	—	—	—	INIRQ
BSC	SDIR	—		—		—		PRC[2:0]	
		ARFC[3:0]				ARF[3:0]			
BSC	SDADR	—	—	—	—	—	—	MXC[1:0]	
BSC	SDTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	RAS[2:0]		
		—	—	RCD[1:0]		RP[2:0]		WR	
		—	—	—	—	—	CL[2:0]		
BSC	SDMOD	MR[14:0]							
		MR[14:0]							
BSC	SDSR	—	—	—	SRFST	INIST	—	—	MRSST
MPU	RSPAGE0	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—	—	—	—
MPU	REPAGE0	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]		V	
MPU	RSPAGE1	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—	—	—	—
MPU	REPAGE1	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]		V	
MPU	RSPAGE2	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—	—	—	—
MPU	REPAGE2	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]		V	
MPU	RSPAGE3	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—	—	—	—

Table 5.2 List of I/O Registers (Bit Order) (10 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MPU	REPAGE3	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			V
MPU	RSPAGE4	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—	—	—	—
MPU	REPAGE4	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			V
MPU	RSPAGE5	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—	—	—	—
MPU	REPAGE5	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			V
MPU	RSPAGE6	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—	—	—	—
MPU	REPAGE6	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			V
MPU	RSPAGE7	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—	—	—	—
MPU	REPAGE7	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			V
MPU	MPEN	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	MPEN
MPU	MPBAC	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	UBAC[2:0]			—

Table 5.2 List of I/O Registers (Bit Order) (11 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MPU	MPECLR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CLR
MPU	MPESTS	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	DRW	DA	IA
MPU	MPDEA	DEA[31:0]							
		DEA[31:0]							
		DEA[31:0]							
		DEA[31:0]							
MPU	MPSA	SA[31:0]							
		SA[31:0]							
		SA[31:0]							
		SA[31:0]							
MPU	MPOPS	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	S
MPU	MPOPI	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	INV
MPU	MHITI	—	—	—	—	—	—	—	—
		HITI[7:0]							
		—	—	—	—	—	—	—	—
		UHACI[2:0]						—	—
MPU	MHITD	—	—	—	—	—	—	—	—
		HITD[7:0]							
		—	—	—	—	—	—	—	—
		UHACD[2:0]						—	—
ICU	IR016	—	—	—	—	—	—	—	IR
ICU	IR021	—	—	—	—	—	—	—	IR
ICU	IR023	—	—	—	—	—	—	—	IR
ICU	IR027	—	—	—	—	—	—	—	IR
ICU	IR028	—	—	—	—	—	—	—	IR
ICU	IR029	—	—	—	—	—	—	—	IR
ICU	IR030	—	—	—	—	—	—	—	IR
ICU	IR031	—	—	—	—	—	—	—	IR
ICU	IR032	—	—	—	—	—	—	—	IR
ICU	IR036	—	—	—	—	—	—	—	IR
ICU	IR037	—	—	—	—	—	—	—	IR
ICU	IR038	—	—	—	—	—	—	—	IR
ICU	IR040	—	—	—	—	—	—	—	IR
ICU	IR041	—	—	—	—	—	—	—	IR
ICU	IR042	—	—	—	—	—	—	—	IR
ICU	IR044	—	—	—	—	—	—	—	IR
ICU	IR045	—	—	—	—	—	—	—	IR

Table 5.2 List of I/O Registers (Bit Order) (12 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ICU	IR046	—	—	—	—	—	—	—	IR
ICU	IR047	—	—	—	—	—	—	—	IR
ICU	IR048	—	—	—	—	—	—	—	IR
ICU	IR049	—	—	—	—	—	—	—	IR
ICU	IR050	—	—	—	—	—	—	—	IR
ICU	IR051	—	—	—	—	—	—	—	IR
ICU	IR056	—	—	—	—	—	—	—	IR
ICU	IR057	—	—	—	—	—	—	—	IR
ICU	IR058	—	—	—	—	—	—	—	IR
ICU	IR059	—	—	—	—	—	—	—	IR
ICU	IR060	—	—	—	—	—	—	—	IR
ICU	IR062	—	—	—	—	—	—	—	IR
ICU	IR063	—	—	—	—	—	—	—	IR
ICU	IR064	—	—	—	—	—	—	—	IR
ICU	IR065	—	—	—	—	—	—	—	IR
ICU	IR066	—	—	—	—	—	—	—	IR
ICU	IR067	—	—	—	—	—	—	—	IR
ICU	IR068	—	—	—	—	—	—	—	IR
ICU	IR069	—	—	—	—	—	—	—	IR
ICU	IR070	—	—	—	—	—	—	—	IR
ICU	IR071	—	—	—	—	—	—	—	IR
ICU	IR072	—	—	—	—	—	—	—	IR
ICU	IR073	—	—	—	—	—	—	—	IR
ICU	IR074	—	—	—	—	—	—	—	IR
ICU	IR075	—	—	—	—	—	—	—	IR
ICU	IR076	—	—	—	—	—	—	—	IR
ICU	IR077	—	—	—	—	—	—	—	IR
ICU	IR078	—	—	—	—	—	—	—	IR
ICU	IR079	—	—	—	—	—	—	—	IR
ICU	IR090	—	—	—	—	—	—	—	IR
ICU	IR091	—	—	—	—	—	—	—	IR
ICU	IR092	—	—	—	—	—	—	—	IR
ICU	IR096	—	—	—	—	—	—	—	IR
ICU	IR098	—	—	—	—	—	—	—	IR
ICU	IR099	—	—	—	—	—	—	—	IR
ICU	IR102	—	—	—	—	—	—	—	IR
ICU	IR114	—	—	—	—	—	—	—	IR
ICU	IR115	—	—	—	—	—	—	—	IR
ICU	IR116	—	—	—	—	—	—	—	IR
ICU	IR117	—	—	—	—	—	—	—	IR
ICU	IR118	—	—	—	—	—	—	—	IR
ICU	IR119	—	—	—	—	—	—	—	IR
ICU	IR120	—	—	—	—	—	—	—	IR
ICU	IR121	—	—	—	—	—	—	—	IR
ICU	IR122	—	—	—	—	—	—	—	IR

Table 5.2 List of I/O Registers (Bit Order) (13 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ICU	IR123	—	—	—	—	—	—	—	IR
ICU	IR124	—	—	—	—	—	—	—	IR
ICU	IR125	—	—	—	—	—	—	—	IR
ICU	IR126	—	—	—	—	—	—	—	IR
ICU	IR127	—	—	—	—	—	—	—	IR
ICU	IR128	—	—	—	—	—	—	—	IR
ICU	IR129	—	—	—	—	—	—	—	IR
ICU	IR130	—	—	—	—	—	—	—	IR
ICU	IR131	—	—	—	—	—	—	—	IR
ICU	IR132	—	—	—	—	—	—	—	IR
ICU	IR133	—	—	—	—	—	—	—	IR
ICU	IR134	—	—	—	—	—	—	—	IR
ICU	IR135	—	—	—	—	—	—	—	IR
ICU	IR136	—	—	—	—	—	—	—	IR
ICU	IR137	—	—	—	—	—	—	—	IR
ICU	IR138	—	—	—	—	—	—	—	IR
ICU	IR139	—	—	—	—	—	—	—	IR
ICU	IR140	—	—	—	—	—	—	—	IR
ICU	IR141	—	—	—	—	—	—	—	IR
ICU	IR142	—	—	—	—	—	—	—	IR
ICU	IR143	—	—	—	—	—	—	—	IR
ICU	IR144	—	—	—	—	—	—	—	IR
ICU	IR145	—	—	—	—	—	—	—	IR
ICU	IR146	—	—	—	—	—	—	—	IR
ICU	IR147	—	—	—	—	—	—	—	IR
ICU	IR148	—	—	—	—	—	—	—	IR
ICU	IR149	—	—	—	—	—	—	—	IR
ICU	IR150	—	—	—	—	—	—	—	IR
ICU	IR151	—	—	—	—	—	—	—	IR
ICU	IR152	—	—	—	—	—	—	—	IR
ICU	IR153	—	—	—	—	—	—	—	IR
ICU	IR154	—	—	—	—	—	—	—	IR
ICU	IR155	—	—	—	—	—	—	—	IR
ICU	IR156	—	—	—	—	—	—	—	IR
ICU	IR157	—	—	—	—	—	—	—	IR
ICU	IR158	—	—	—	—	—	—	—	IR
ICU	IR159	—	—	—	—	—	—	—	IR
ICU	IR160	—	—	—	—	—	—	—	IR
ICU	IR161	—	—	—	—	—	—	—	IR
ICU	IR162	—	—	—	—	—	—	—	IR
ICU	IR163	—	—	—	—	—	—	—	IR
ICU	IR164	—	—	—	—	—	—	—	IR
ICU	IR165	—	—	—	—	—	—	—	IR
ICU	IR166	—	—	—	—	—	—	—	IR
ICU	IR167	—	—	—	—	—	—	—	IR

Table 5.2 List of I/O Registers (Bit Order) (14 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ICU	IR168	—	—	—	—	—	—	—	IR
ICU	IR169	—	—	—	—	—	—	—	IR
ICU	IR170	—	—	—	—	—	—	—	IR
ICU	IR171	—	—	—	—	—	—	—	IR
ICU	IR172	—	—	—	—	—	—	—	IR
ICU	IR173	—	—	—	—	—	—	—	IR
ICU	IR174	—	—	—	—	—	—	—	IR
ICU	IR175	—	—	—	—	—	—	—	IR
ICU	IR176	—	—	—	—	—	—	—	IR
ICU	IR177	—	—	—	—	—	—	—	IR
ICU	IR178	—	—	—	—	—	—	—	IR
ICU	IR179	—	—	—	—	—	—	—	IR
ICU	IR180	—	—	—	—	—	—	—	IR
ICU	IR181	—	—	—	—	—	—	—	IR
ICU	IR182	—	—	—	—	—	—	—	IR
ICU	IR183	—	—	—	—	—	—	—	IR
ICU	IR184	—	—	—	—	—	—	—	IR
ICU	IR185	—	—	—	—	—	—	—	IR
ICU	IR198	—	—	—	—	—	—	—	IR
ICU	IR199	—	—	—	—	—	—	—	IR
ICU	IR200	—	—	—	—	—	—	—	IR
ICU	IR201	—	—	—	—	—	—	—	IR
ICU	IR202	—	—	—	—	—	—	—	IR
ICU	IR203	—	—	—	—	—	—	—	IR
ICU	IR214	—	—	—	—	—	—	—	IR
ICU	IR215	—	—	—	—	—	—	—	IR
ICU	IR216	—	—	—	—	—	—	—	IR
ICU	IR217	—	—	—	—	—	—	—	IR
ICU	IR218	—	—	—	—	—	—	—	IR
ICU	IR219	—	—	—	—	—	—	—	IR
ICU	IR220	—	—	—	—	—	—	—	IR
ICU	IR221	—	—	—	—	—	—	—	IR
ICU	IR222	—	—	—	—	—	—	—	IR
ICU	IR223	—	—	—	—	—	—	—	IR
ICU	IR224	—	—	—	—	—	—	—	IR
ICU	IR225	—	—	—	—	—	—	—	IR
ICU	IR226	—	—	—	—	—	—	—	IR
ICU	IR227	—	—	—	—	—	—	—	IR
ICU	IR228	—	—	—	—	—	—	—	IR
ICU	IR229	—	—	—	—	—	—	—	IR
ICU	IR234	—	—	—	—	—	—	—	IR
ICU	IR235	—	—	—	—	—	—	—	IR
ICU	IR236	—	—	—	—	—	—	—	IR
ICU	IR237	—	—	—	—	—	—	—	IR
ICU	IR238	—	—	—	—	—	—	—	IR

Table 5.2 List of I/O Registers (Bit Order) (15 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ICU	IR239	—	—	—	—	—	—	—	IR
ICU	IR240	—	—	—	—	—	—	—	IR
ICU	IR241	—	—	—	—	—	—	—	IR
ICU	IR246	—	—	—	—	—	—	—	IR
ICU	IR247	—	—	—	—	—	—	—	IR
ICU	IR248	—	—	—	—	—	—	—	IR
ICU	IR249	—	—	—	—	—	—	—	IR
ICU	IR250	—	—	—	—	—	—	—	IR
ICU	IR251	—	—	—	—	—	—	—	IR
ICU	IR252	—	—	—	—	—	—	—	IR
ICU	IR253	—	—	—	—	—	—	—	IR
ICU	DTCER027	—	—	—	—	—	—	—	DTCE
ICU	DTCER028	—	—	—	—	—	—	—	DTCE
ICU	DTCER029	—	—	—	—	—	—	—	DTCE
ICU	DTCER030	—	—	—	—	—	—	—	DTCE
ICU	DTCER031	—	—	—	—	—	—	—	DTCE
ICU	DTCER036	—	—	—	—	—	—	—	DTCE
ICU	DTCER037	—	—	—	—	—	—	—	DTCE
ICU	DTCER040	—	—	—	—	—	—	—	DTCE
ICU	DTCER041	—	—	—	—	—	—	—	DTCE
ICU	DTCER045	—	—	—	—	—	—	—	DTCE
ICU	DTCER046	—	—	—	—	—	—	—	DTCE
ICU	DTCER049	—	—	—	—	—	—	—	DTCE
ICU	DTCER050	—	—	—	—	—	—	—	DTCE
ICU	DTCER064	—	—	—	—	—	—	—	DTCE
ICU	DTCER065	—	—	—	—	—	—	—	DTCE
ICU	DTCER066	—	—	—	—	—	—	—	DTCE
ICU	DTCER067	—	—	—	—	—	—	—	DTCE
ICU	DTCER068	—	—	—	—	—	—	—	DTCE
ICU	DTCER069	—	—	—	—	—	—	—	DTCE
ICU	DTCER070	—	—	—	—	—	—	—	DTCE
ICU	DTCER071	—	—	—	—	—	—	—	DTCE
ICU	DTCER072	—	—	—	—	—	—	—	DTCE
ICU	DTCER073	—	—	—	—	—	—	—	DTCE
ICU	DTCER074	—	—	—	—	—	—	—	DTCE
ICU	DTCER075	—	—	—	—	—	—	—	DTCE
ICU	DTCER076	—	—	—	—	—	—	—	DTCE
ICU	DTCER077	—	—	—	—	—	—	—	DTCE
ICU	DTCER078	—	—	—	—	—	—	—	DTCE
ICU	DTCER079	—	—	—	—	—	—	—	DTCE
ICU	DTCER098	—	—	—	—	—	—	—	DTCE
ICU	DTCER099	—	—	—	—	—	—	—	DTCE
ICU	DTCER102	—	—	—	—	—	—	—	DTCE
ICU	DTCER114	—	—	—	—	—	—	—	DTCE
ICU	DTCER115	—	—	—	—	—	—	—	DTCE

Table 5.2 List of I/O Registers (Bit Order) (16 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ICU	DTCER116	—	—	—	—	—	—	—	DTCE
ICU	DTCER117	—	—	—	—	—	—	—	DTCE
ICU	DTCER121	—	—	—	—	—	—	—	DTCE
ICU	DTCER122	—	—	—	—	—	—	—	DTCE
ICU	DTCER125	—	—	—	—	—	—	—	DTCE
ICU	DTCER126	—	—	—	—	—	—	—	DTCE
ICU	DTCER129	—	—	—	—	—	—	—	DTCE
ICU	DTCER130	—	—	—	—	—	—	—	DTCE
ICU	DTCER131	—	—	—	—	—	—	—	DTCE
ICU	DTCER132	—	—	—	—	—	—	—	DTCE
ICU	DTCER134	—	—	—	—	—	—	—	DTCE
ICU	DTCER135	—	—	—	—	—	—	—	DTCE
ICU	DTCER136	—	—	—	—	—	—	—	DTCE
ICU	DTCER137	—	—	—	—	—	—	—	DTCE
ICU	DTCER138	—	—	—	—	—	—	—	DTCE
ICU	DTCER139	—	—	—	—	—	—	—	DTCE
ICU	DTCER140	—	—	—	—	—	—	—	DTCE
ICU	DTCER141	—	—	—	—	—	—	—	DTCE
ICU	DTCER142	—	—	—	—	—	—	—	DTCE
ICU	DTCER143	—	—	—	—	—	—	—	DTCE
ICU	DTCER144	—	—	—	—	—	—	—	DTCE
ICU	DTCER145	—	—	—	—	—	—	—	DTCE
ICU	DTCER149	—	—	—	—	—	—	—	DTCE
ICU	DTCER150	—	—	—	—	—	—	—	DTCE
ICU	DTCER153	—	—	—	—	—	—	—	DTCE
ICU	DTCER154	—	—	—	—	—	—	—	DTCE
ICU	DTCER157	—	—	—	—	—	—	—	DTCE
ICU	DTCER158	—	—	—	—	—	—	—	DTCE
ICU	DTCER159	—	—	—	—	—	—	—	DTCE
ICU	DTCER160	—	—	—	—	—	—	—	DTCE
ICU	DTCER162	—	—	—	—	—	—	—	DTCE
ICU	DTCER163	—	—	—	—	—	—	—	DTCE
ICU	DTCER164	—	—	—	—	—	—	—	DTCE
ICU	DTCER165	—	—	—	—	—	—	—	DTCE
ICU	DTCER166	—	—	—	—	—	—	—	DTCE
ICU	DTCER167	—	—	—	—	—	—	—	DTCE
ICU	DTCER168	—	—	—	—	—	—	—	DTCE
ICU	DTCER169	—	—	—	—	—	—	—	DTCE
ICU	DTCER174	—	—	—	—	—	—	—	DTCE
ICU	DTCER175	—	—	—	—	—	—	—	DTCE
ICU	DTCER177	—	—	—	—	—	—	—	DTCE
ICU	DTCER178	—	—	—	—	—	—	—	DTCE
ICU	DTCER180	—	—	—	—	—	—	—	DTCE
ICU	DTCER181	—	—	—	—	—	—	—	DTCE
ICU	DTCER183	—	—	—	—	—	—	—	DTCE

Table 5.2 List of I/O Registers (Bit Order) (17 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ICU	DTCER184	—	—	—	—	—	—	—	DTCE
ICU	DTCER198	—	—	—	—	—	—	—	DTCE
ICU	DTCER199	—	—	—	—	—	—	—	DTCE
ICU	DTCER200	—	—	—	—	—	—	—	DTCE
ICU	DTCER201	—	—	—	—	—	—	—	DTCE
ICU	DTCER202	—	—	—	—	—	—	—	DTCE
ICU	DTCER203	—	—	—	—	—	—	—	DTCE
ICU	DTCER215	—	—	—	—	—	—	—	DTCE
ICU	DTCER216	—	—	—	—	—	—	—	DTCE
ICU	DTCER219	—	—	—	—	—	—	—	DTCE
ICU	DTCER220	—	—	—	—	—	—	—	DTCE
ICU	DTCER223	—	—	—	—	—	—	—	DTCE
ICU	DTCER224	—	—	—	—	—	—	—	DTCE
ICU	DTCER227	—	—	—	—	—	—	—	DTCE
ICU	DTCER228	—	—	—	—	—	—	—	DTCE
ICU	DTCER235	—	—	—	—	—	—	—	DTCE
ICU	DTCER236	—	—	—	—	—	—	—	DTCE
ICU	DTCER239	—	—	—	—	—	—	—	DTCE
ICU	DTCER240	—	—	—	—	—	—	—	DTCE
ICU	DTCER247	—	—	—	—	—	—	—	DTCE
ICU	DTCER248	—	—	—	—	—	—	—	DTCE
ICU	DTCER251	—	—	—	—	—	—	—	DTCE
ICU	DTCER252	—	—	—	—	—	—	—	DTCE
ICU	IER02	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER03	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER04	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER05	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER06	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER07	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER08	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER09	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER0B	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER0C	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER0E	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER0F	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER10	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER11	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER12	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER13	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER14	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER15	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER16	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER17	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER18	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER19	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0

Table 5.2 List of I/O Registers (Bit Order) (18 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ICU	IER1A	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER1B	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER1C	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER1D	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER1E	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER1F	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	SWINTR	—	—	—	—	—	—	—	SWINT
ICU	FIR	FIEN	—	—	—	—	—	—	—
		FVCT[7:0]							
ICU	IPR00	—	—	—	—	IPR[3:0]			
ICU	IPR01	—	—	—	—	IPR[3:0]			
ICU	IPR02	—	—	—	—	IPR[3:0]			
ICU	IPR03	—	—	—	—	IPR[3:0]			
ICU	IPR04	—	—	—	—	IPR[3:0]			
ICU	IPR05	—	—	—	—	IPR[3:0]			
ICU	IPR06	—	—	—	—	IPR[3:0]			
ICU	IPR07	—	—	—	—	IPR[3:0]			
ICU	IPR08	—	—	—	—	IPR[3:0]			
ICU	IPR0C	—	—	—	—	IPR[3:0]			
ICU	IPR0D	—	—	—	—	IPR[3:0]			
ICU	IPR0E	—	—	—	—	IPR[3:0]			
ICU	IPR10	—	—	—	—	IPR[3:0]			
ICU	IPR11	—	—	—	—	IPR[3:0]			
ICU	IPR12	—	—	—	—	IPR[3:0]			
ICU	IPR14	—	—	—	—	IPR[3:0]			
ICU	IPR15	—	—	—	—	IPR[3:0]			
ICU	IPR18	—	—	—	—	IPR[3:0]			
ICU	IPR1E	—	—	—	—	IPR[3:0]			
ICU	IPR1F	—	—	—	—	IPR[3:0]			
ICU	IPR20	—	—	—	—	IPR[3:0]			
ICU	IPR21	—	—	—	—	IPR[3:0]			
ICU	IPR22	—	—	—	—	IPR[3:0]			
ICU	IPR23	—	—	—	—	IPR[3:0]			
ICU	IPR24	—	—	—	—	IPR[3:0]			
ICU	IPR25	—	—	—	—	IPR[3:0]			
ICU	IPR26	—	—	—	—	IPR[3:0]			
ICU	IPR27	—	—	—	—	IPR[3:0]			
ICU	IPR28	—	—	—	—	IPR[3:0]			
ICU	IPR29	—	—	—	—	IPR[3:0]			
ICU	IPR2A	—	—	—	—	IPR[3:0]			
ICU	IPR2B	—	—	—	—	IPR[3:0]			
ICU	IPR2C	—	—	—	—	IPR[3:0]			
ICU	IPR2D	—	—	—	—	IPR[3:0]			
ICU	IPR2E	—	—	—	—	IPR[3:0]			
ICU	IPR2F	—	—	—	—	IPR[3:0]			

Table 5.2 List of I/O Registers (Bit Order) (19 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ICU	IPR3A	—	—	—	—	IPR[3:0]			
ICU	IPR3B	—	—	—	—	IPR[3:0]			
ICU	IPR3C	—	—	—	—	IPR[3:0]			
ICU	IPR40	—	—	—	—	IPR[3:0]			
ICU	IPR44	—	—	—	—	IPR[3:0]			
ICU	IPR45	—	—	—	—	IPR[3:0]			
ICU	IPR48	—	—	—	—	IPR[3:0]			
ICU	IPR51	—	—	—	—	IPR[3:0]			
ICU	IPR52	—	—	—	—	IPR[3:0]			
ICU	IPR53	—	—	—	—	IPR[3:0]			
ICU	IPR54	—	—	—	—	IPR[3:0]			
ICU	IPR55	—	—	—	—	IPR[3:0]			
ICU	IPR56	—	—	—	—	IPR[3:0]			
ICU	IPR57	—	—	—	—	IPR[3:0]			
ICU	IPR58	—	—	—	—	IPR[3:0]			
ICU	IPR59	—	—	—	—	IPR[3:0]			
ICU	IPR5A	—	—	—	—	IPR[3:0]			
ICU	IPR5B	—	—	—	—	IPR[3:0]			
ICU	IPR5C	—	—	—	—	IPR[3:0]			
ICU	IPR5D	—	—	—	—	IPR[3:0]			
ICU	IPR5E	—	—	—	—	IPR[3:0]			
ICU	IPR5F	—	—	—	—	IPR[3:0]			
ICU	IPR60	—	—	—	—	IPR[3:0]			
ICU	IPR61	—	—	—	—	IPR[3:0]			
ICU	IPR62	—	—	—	—	IPR[3:0]			
ICU	IPR63	—	—	—	—	IPR[3:0]			
ICU	IPR64	—	—	—	—	IPR[3:0]			
ICU	IPR65	—	—	—	—	IPR[3:0]			
ICU	IPR66	—	—	—	—	IPR[3:0]			
ICU	IPR67	—	—	—	—	IPR[3:0]			
ICU	IPR68	—	—	—	—	IPR[3:0]			
ICU	IPR69	—	—	—	—	IPR[3:0]			
ICU	IPR6A	—	—	—	—	IPR[3:0]			
ICU	IPR6B	—	—	—	—	IPR[3:0]			
ICU	IPR70	—	—	—	—	IPR[3:0]			
ICU	IPR71	—	—	—	—	IPR[3:0]			
ICU	IPR72	—	—	—	—	IPR[3:0]			
ICU	IPR73	—	—	—	—	IPR[3:0]			
ICU	IPR74	—	—	—	—	IPR[3:0]			
ICU	IPR75	—	—	—	—	IPR[3:0]			
ICU	IPR80	—	—	—	—	IPR[3:0]			
ICU	IPR81	—	—	—	—	IPR[3:0]			
ICU	IPR82	—	—	—	—	IPR[3:0]			
ICU	IPR83	—	—	—	—	IPR[3:0]			
ICU	IPR85	—	—	—	—	IPR[3:0]			

Table 5.2 List of I/O Registers (Bit Order) (20 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ICU	IPR86	—	—	—	—	IPR[3:0]			
ICU	IPR88	—	—	—	—	IPR[3:0]			
ICU	IPR89	—	—	—	—	IPR[3:0]			
ICU	IPR8A	—	—	—	—	IPR[3:0]			
ICU	IPR8B	—	—	—	—	IPR[3:0]			
ICU	IPR8C	—	—	—	—	IPR[3:0]			
ICU	IPR8D	—	—	—	—	IPR[3:0]			
ICU	IPR8E	—	—	—	—	IPR[3:0]			
ICU	IPR8F	—	—	—	—	IPR[3:0]			
ICU	DMRSR0								
ICU	DMRSR1								
ICU	DMRSR2								
ICU	DMRSR3								
ICU	IRQCR0	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR1	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR2	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR3	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR4	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR5	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR6	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR7	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR8	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR9	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR10	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR11	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR12	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR13	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR14	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR15	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	NMISR	—	—	—	—	—	OSTST	LVDST	NMIST
ICU	NMIER	—	—	—	—	—	OSTEN	LVDEN	NMIEN
ICU	NMICLR	—	—	—	—	—	OSTCLR	—	NMICLR
ICU	NMICR	—	—	—	—	NMIMD	—	—	—
CMT	CMSTR0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	STR1	STR0
CMT0	CMCR	—	—	—	—	—	—	—	—
		—	CMIE	—	—	—	—	CKS[1:0]	
CMT0	CMCNT								
CMT0	CMCOR								
CMT1	CMCR	—	—	—	—	—	—	—	—
		—	CMIE	—	—	—	—	CKS[1:0]	
CMT1	CMCNT								

Table 5.2 List of I/O Registers (Bit Order) (21 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
CMT1	CMCOR								
CMT	CMSTR1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	STR3	STR2
CMT2	CMCR	—	—	—	—	—	—	—	—
		—	CMIE	—	—	—	—	CKS[1:0]	
CMT2	CMCNT								
CMT2	CMCOR								
CMT3	CMCR	—	—	—	—	—	—	—	—
		—	CMIE	—	—	—	—	CKS[1:0]	
CMT3	CMCNT								
CMT3	CMCOR								
WDT	TCSR	—	TMS	TME	—	—	CKS[2:0]		
WDT	WINA								
WDT	TCNT								
WDT	WINB								
WDT	RSTCSR	WOVF	RSTE	—	—	—	—	—	—
IWDT	IWDTRR								
IWDT	IWDTCR	—	—	—	—	—	—	—	—
		CKS[3:0]					—	—	TOPS[1:0]
IWDT	IWDTSR	—	UNDF	CNTVAL[13:0]					
		CNTVAL[13:0]							
AD0	ADDRA *5	—	—	—	—	—	—		
AD0	ADDRB *5	—	—	—	—	—	—		
AD0	ADDRC *5	—	—	—	—	—	—		
AD0	ADDRD *5	—	—	—	—	—	—		
AD0	ADCSR	—	ADIE	ADST	—	CH[3:0]			
AD0	ADCR	TRGS[2:0]			—	CKS[1:0]		MODE[1:0]	
AD0	ADDPR	DPSEL	—	—	—	—	—	—	—
AD0	ADSSTR								
AD0	ADDIAGR	—	—	—	—	—	—	DIAG[1:0]	
AD1	ADDRA *5	—	—	—	—	—	—		
AD1	ADDRB *5	—	—	—	—	—	—		

Table 5.2 List of I/O Registers (Bit Order) (22 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
AD1	ADDRC *5	—	—	—	—	—	—		
AD1	ADDRD *5	—	—	—	—	—	—		
AD1	ADCSR	—	ADIE	ADST	—	CH[3:0]			
AD1	ADCR	TRGS[2:0]			—	CKS[1:0]		MODE[1:0]	
AD1	ADDP	DPSEL	—	—	—	—	—	—	—
AD1	ADSSTR								
AD1	ADDIAGR	—	—	—	—	—	—	DIAG[1:0]	
DA	DADR0								
DA	DADR1								
DA	DACR	DAOE1	DAOE0	DAE	—	—	—	—	—
DA	DADPR	DPSEL	—	—	—	—	—	—	—
PPG0	PCR	G3CMS[1:0]		G2CMS[1:0]		G1CMS[1:0]		G0CMS[1:0]	
PPG0	PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
PPG0	NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
PPG0	NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
PPG0	PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
PPG0	PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
PPG0	NDRH *1	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
PPG0	NDRL *2	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
PPG0	NDRH2 *1	—	—	—	—	NDR11	NDR10	NDR9	NDR8
PPG0	NDRL2 *2	—	—	—	—	NDR3	NDR2	NDR1	NDR0
PPG1	PTRSLR	—	—	—	—	—	—	—	PTRSL
PPG1	PCR	G3CMS[1:0]		G2CMS[1:0]		G1CMS[1:0]		G0CMS[1:0]	
PPG1	PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
PPG1	NDERH	NDER31	NDER30	NDER29	NDER28	NDER27	NDER26	NDER25	NDER24
PPG1	NDERL	NDER23	NDER22	NDER21	NDER20	NDER19	NDER18	NDER17	NDER16
PPG1	PODRH	POD31	POD30	POD29	POD28	POD27	POD26	POD25	POD24
PPG1	PODRL	POD23	POD22	POD21	POD20	POD19	POD18	POD17	POD16
PPG1	NDRH *3	NDR31	NDR30	NDR29	NDR28	NDR27	NDR26	NDR25	NDR24
PPG1	NDRL *4	NDR23	NDR22	NDR21	NDR20	NDR19	NDR18	NDR17	NDR16
PPG1	NDRH2 *3	—	—	—	—	NDR27	NDR26	NDR25	NDR24
PPG1	NDRL2 *4	—	—	—	—	NDR19	NDR18	NDR17	NDR16
TMR0	TCR	CMIEB	CMIEA	OVIE	CCLR[1:0]		—	—	—
TMR1	TCR	CMIEB	CMIEA	OVIE	CCLR[1:0]		—	—	—
TMR0	TCSR	—	—	—	ADTE	OSB[1:0]		OSA[1:0]	

Table 5.2 List of I/O Registers (Bit Order) (23 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
TMR1	TCSR	—	—	—	—	OSB[1:0]		OSA[1:0]	
TMR0	TCORA								
TMR1	TCORA								
TMR0	TCORB								
TMR1	TCORB								
TMR0	TCNT								
TMR1	TCNT								
TMR0	TCCR	TMRIS	—	—	CSS[1:0]		CKS[2:0]		
TMR1	TCCR	TMRIS	—	—	CSS[1:0]		CKS[2:0]		
TMR01	TCORA								
TMR01	TCORB								
TMR01	TCNT								
TMR01	TCCR								
TMR2	TCR	CMIEB	CMIEA	OVIE	CCLR[1:0]		—	—	—
TMR3	TCR	CMIEB	CMIEA	OVIE	CCLR[1:0]		—	—	—
TMR2	TCSR	—	—	—	ADTE	OSB[1:0]		OSA[1:0]	
TMR3	TCSR	—	—	—	—	OSB[1:0]		OSA[1:0]	
TMR2	TCORA								
TMR3	TCORA								
TMR2	TCORB								
TMR3	TCORB								
TMR2	TCNT								
TMR3	TCNT								
TMR2	TCCR	TMRIS	—	—	CSS[1:0]		CKS[2:0]		
TMR3	TCCR	TMRIS	—	—	CSS[1:0]		CKS[2:0]		
TMR23	TCORA								
TMR23	TCORB								
TMR23	TCNT								
TMR23	TCCR								
SCI0	SMR	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
SCI0	BRR								
SCI0	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SCI0	TDR								
SCI0	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
SCI0	RDR								
SCI0	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI0	SEMR	—	—	—	ABCS	—	—	—	ACS0

Table 5.2 List of I/O Registers (Bit Order) (24 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
SMCI0	SMR	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
SMCI0	BRR								
SMCI0	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SMCI0	TDR								
SMCI0	SSR	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
SMCI0	RDR								
SMCI0	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI1	SMR	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
SCI1	BRR								
SCI1	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SCI1	TDR								
SCI1	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
SCI1	RDR								
SCI1	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI1	SEMR	—	—	—	ABCS	—	—	—	ACS0
SMCI1	SMR	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
SMCI1	BRR								
SMCI1	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SMCI1	TDR								
SMCI1	SSR	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
SMCI1	RDR								
SMCI1	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI2	SMR	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
SCI2	BRR								
SCI2	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SCI2	TDR								
SCI2	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
SCI2	RDR								
SCI2	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI2	SEMR	—	—	—	ABCS	—	—	—	ACS0
SMCI2	SMR	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
SMCI2	BRR								
SMCI2	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SMCI2	TDR								
SMCI2	SSR	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
SMCI2	RDR								
SMCI2	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI3	SMR	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
SCI3	BRR								
SCI3	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SCI3	TDR								
SCI3	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
SCI3	RDR								
SCI3	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI3	SEMR	—	—	—	ABCS	—	—	—	ACS0

Table 5.2 List of I/O Registers (Bit Order) (25 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
SMCI3	SMR	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
SMCI3	BRR								
SMCI3	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SMCI3	TDR								
SMCI3	SSR	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
SMCI3	RDR								
SMCI3	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI5	SMR	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
SCI5	BRR								
SCI5	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SCI5	TDR								
SCI5	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
SCI5	RDR								
SCI5	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI5	SEMR	—	—	—	ABCS	—	—	—	ACS0
SMCI5	SMR	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
SMCI5	BRR								
SMCI5	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SMCI5	TDR								
SMCI5	SSR	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
SMCI5	RDR								
SMCI5	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI6	SMR	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
SCI6	BRR								
SCI6	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SCI6	TDR								
SCI6	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
SCI6	RDR								
SCI6	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI6	SEMR	—	—	—	ABCS	—	—	—	ACS0
SMCI6	SMR	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
SMCI6	BRR								
SMCI6	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SMCI6	TDR								
SMCI6	SSR	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
SMCI6	RDR								
SMCI6	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
CRC	CRCCR	DORCLR	—	—	—	—	LMS	GPS[1:0]	
CRC	CRCDIR								
CRC	CRCDOR								
RIIC0	ICCR1	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
RIIC0	ICCR2	BBSY	MST	TRS	—	SP	RS	ST	—
RIIC0	ICMR1	MTWP	CKS[2:0]			BCWP	BC[2:0]		
RIIC0	ICMR2	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS

Table 5.2 List of I/O Registers (Bit Order) (26 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
RIIC0	ICMR3	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
RIIC0	ICFER	FMPE	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
RIIC0	ICSER	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
RIIC0	ICIER	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
RIIC0	ICSR1	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
RIIC0	ICSR2	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
RIIC0	SARL0	SVA[6:0]							SVA0
RIIC0	SARU0	—	—	—	—	—	SVA[1:0]	FS	
RIIC0	SARL1	SVA[6:0]							SVA0
RIIC0	SARU1	—	—	—	—	—	SVA[1:0]	FS	
RIIC0	SARL2	SVA[6:0]							SVA0
RIIC0	SARU2	—	—	—	—	—	SVA[1:0]	FS	
RIIC0	ICBRL	—	—	—	BRL[4:0]				
RIIC0	ICBRH	—	—	—	BRH[4:0]				
RIIC0	ICDRT								
RIIC0	ICDRR								
RIIC1	ICCR1	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
RIIC1	ICCR2	BBSY	MST	TRS	—	SP	RS	ST	—
RIIC1	ICMR1	MTWP	CKS[2:0]			BCWP	BC[2:0]		
RIIC1	ICMR2	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
RIIC1	ICMR3	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
RIIC1	ICFER	FMPE	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
RIIC1	ICSER	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
RIIC1	ICIER	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
RIIC1	ICSR1	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
RIIC1	ICSR2	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
RIIC1	SARL0	SVA[6:0]							SVA0
RIIC1	SARU0	—	—	—	—	—	SVA[1:0]	FS	
RIIC1	SARL1	SVA[6:0]							SVA0
RIIC1	SARU1	—	—	—	—	—	SVA[1:0]	FS	
RIIC1	SARL2	SVA[6:0]							SVA0
RIIC1	SARU2	—	—	—	—	—	SVA[1:0]	FS	
RIIC1	ICBRL	—	—	—	BRL[4:0]				
RIIC1	ICBRH	—	—	—	BRH[4:0]				
RIIC1	ICDRT								
RIIC1	ICDRR								
RSPIO	SPCR	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
RSPIO	SSLP	—	—	—	—	SSLP3	SSLP2	SSLP1	SSLP0
RSPIO	SPPCR	—	—	MOIFE	MOIFV	—	SPOM	SPLP2	SPLP
RSPIO	SPSR	SPRF	—	SPTEF	—	PERF	MODF	IDLNF	OVRF
RSPIO	SPDR	H[15:0]							
		H[15:0]							
		L[15:0]							
		L[15:0]							
RSPIO	SPSCR	—	—	—	—	—	SPSLN[2:0]		

Table 5.2 List of I/O Registers (Bit Order) (27 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
RSPI0	SPSSR	—	SPECM[2:0]			—	SPCP[2:0]		
RSPI0	SPBR	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
RSPI0	SPDCR	—	—	SPLW	SPRDTD	SLSEL[1:0]		SPFC[1:0]	
RSPI0	SPCKD	—	—	—	—	—	SCKDL[2:0]		
RSPI0	SSLND	—	—	—	—	—	SLNDL[2:0]		
RSPI0	SPND	—	—	—	—	—	SPNDL[2:0]		
RSPI0	SPCR2	—	—	—	—	PTE	SPIIE	SPOE	SPPE
RSPI0	SPCMD0	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI0	SPCMD1	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI0	SPCMD2	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI0	SPCMD3	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI0	SPCMD4	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI0	SPCMD5	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI0	SPCMD6	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI0	SPCMD7	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI1	SPCR	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
RSPI1	SSLP	—	—	—	—	SSLP3	SSLP2	SSLP1	SSLP0
RSPI1	SPPCR	—	—	MOIFE	MOIFV	—	SPOM	SPLP2	SPLP
RSPI1	SPSR	SPRF	—	SPTEF	—	PERF	MODF	IDLNF	OVRF
RSPI1	SPDR	H[15:0]							
		H[15:0]							
		L[15:0]							
		L[15:0]							
RSPI1	SPSCR	—	—	—	—	—	SPSLN[2:0]		
RSPI1	SPSSR	—	SPECM[2:0]			—	SPCP[2:0]		
RSPI1	SPBR	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
RSPI1	SPDCR	—	—	SPLW	SPRDTD	SLSEL[1:0]		SPFC[1:0]	
RSPI1	SPCKD	—	—	—	—	—	SCKDL[2:0]		
RSPI1	SSLND	—	—	—	—	—	SLNDL[2:0]		
RSPI1	SPND	—	—	—	—	—	SPNDL[2:0]		
RSPI1	SPCR2	—	—	—	—	PTE	SPIIE	SPOE	SPPE
RSPI1	SPCMD0	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI1	SPCMD1	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI1	SPCMD2	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	

Table 5.2 List of I/O Registers (Bit Order) (28 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
RSP11	SPCMD3	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]		BRDV[1:0]		CPOL	CPHA	
RSP11	SPCMD4	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]		BRDV[1:0]		CPOL	CPHA	
RSP11	SPCMD5	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]		BRDV[1:0]		CPOL	CPHA	
RSP11	SPCMD6	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]		BRDV[1:0]		CPOL	CPHA	
RSP11	SPCMD7	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]		BRDV[1:0]		CPOL	CPHA	
MTU3	TCR	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
MTU4	TCR	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
MTU3	TMDR	—	—	BFB	BFA	MD[3:0]			
MTU4	TMDR	—	—	BFB	BFA	MD[3:0]			
MTU3	TIORH	IOB[3:0]				IOA[3:0]			
MTU3	TIORL	IOD[3:0]				IOC[3:0]			
MTU4	TIORH	IOB[3:0]				IOA[3:0]			
MTU4	TIORL	IOD[3:0]				IOC[3:0]			
MTU3	TIER	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
MTU4	TIER	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
MTUA	TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
MTUA	TGCR	—	BDC	N	P	FB	WF	VF	UF
MTUA	TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
MTUA	TOCR2	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
MTU3	TCNT								
MTU4	TCNT								
MTUA	TCDR								
MTUA	TDDR								
MTU3	TGRA								
MTU3	TGRB								
MTU4	TGRA								
MTU4	TGRB								
MTUA	TCNTS								
MTUA	TCBR								

Table 5.2 List of I/O Registers (Bit Order) (29 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTU3	TGRC								
MTU3	TGRD								
MTU4	TGRC								
MTU4	TGRD								
MTU3	TSR	TCFD	—	—	—	—	—	—	—
MTU4	TSR	TCFD	—	—	—	—	—	—	—
MTUA	TITCR	T3AEN	T3ACOR[2:0]			T4VEN	T4VCOR[2:0]		
MTUA	TITCNT	—	T3ACNT[2:0]			—	T4VCNT[2:0]		
MTUA	TBTER	—	—	—	—	—	—	BTE[1:0]	
MTUA	TDER	—	—	—	—	—	—	—	TDER
MTUA	TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
MTU3	TBTM	—	—	—	—	—	—	TTSB	TTSA
MTU4	TBTM	—	—	—	—	—	—	TTSB	TTSA
MTU4	TADCR	BF[1:0]		—	—	—	—	—	—
		UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
MTU4	TADCORA								
MTU4	TADCORB								
MTU4	TADCOBRA								
MTU4	TADCOBRB								
MTUA	TWCR	CCE	—	—	—	—	—	—	WRE
MTUA	TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0
MTUA	TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
MTUA	TRWER	—	—	—	—	—	—	—	RWE
MTU0	TCR	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
MTU0	TMDR	—	BFE	BFB	BFA	MD[3:0]			
MTU0	TIORH	IOB[3:0]				IOA[3:0]			
MTU0	TIORL	IOD[3:0]				IOC[3:0]			
MTU0	TIER	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
MTU0	TSR	—	—	—	—	—	—	—	—
MTU0	TCNT								
MTU0	TGRA								
MTU0	TGRB								
MTU0	TGRC								

Table 5.2 List of I/O Registers (Bit Order) (30 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTU0	TGRD								
MTU0	TGRE								
MTU0	TGRF								
MTU0	TIER2	—	—	—	—	—	—	TGIEF	TGIEE
MTU0	TBTM	—	—	—	—	—	TTSE	TTSB	TTSA
MTU1	TCR	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]		
MTU1	TMDR	—	—	—	—	MD[3:0]			
MTU1	TIOR	IOB[3:0]				IOA[3:0]			
MTU1	TIER	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
MTU1	TSR	TCFD	—	—	—	—	—	—	—
MTU1	TCNT								
MTU1	TGRA								
MTU1	TGRB								
MTU1	TICCR	—	—	—	—	I2BE	I2AE	I1BE	I1AE
MTU2	TCR	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]		
MTU2	TMDR	—	—	—	—	MD[3:0]			
MTU2	TIOR	IOB[3:0]				IOA[3:0]			
MTU2	TIER	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
MTU2	TSR	TCFD	—	—	—	—	—	—	—
MTU2	TCNT								
MTU2	TGRA								
MTU2	TGRB								
MTU5	TCNTU								
MTU5	TGRU								
MTU5	TCRU	—	—	—	—	—	—	TPSC[1:0]	
MTU5	TIORU	—	—	—	IOC[4:0]				
MTU5	TCNTV								
MTU5	TGRV								
MTU5	TCRV	—	—	—	—	—	—	TPSC[1:0]	
MTU5	TIORV	—	—	—	IOC[4:0]				
MTU5	TCNTW								

Table 5.2 List of I/O Registers (Bit Order) (31 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTU5	TGRW								
MTU5	TCRW	—	—	—	—	—	—	TPSC[1:0]	
MTU5	TIORW	—	—	—	IOC[4:0]				
MTU5	TIER	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
MTU5	TSTR	—	—	—	—	—	CSTU5	CSTV5	CSTW5
MTU5	TCNTCMPCLR	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
POE	ICSR1	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE1
		POE3M[1:0]		POE2M[1:0]		POE1M[1:0]		POE0M[1:0]	
POE	OCSR1	OSF1	—	—	—	—	—	OCE1	OIE1
		—	—	—	—	—	—	—	—
POE	ICSR2	POE7F	POE6F	POE5F	POE4F	—	—	—	PIE2
		POE7M[1:0]		POE6M[1:0]		POE5M[1:0]		POE4M[1:0]	
POE	OCSR2	OSF2	—	—	—	—	—	OCE2	OIE2
		—	—	—	—	—	—	—	—
POE	ICSR3	—	—	—	POE8F	—	—	POE8E	PIE3
		—	—	—	—	—	—	POE8M[1:0]	
POE	SPOER	—	—	—	—	CH6HIZ	CH910HIZ	CH0HIZ	CH34HIZ
POE	POECR1	PE7ZE	PE6ZE	PE5ZE	PE4ZE	PE3ZE	PE2ZE	PE1ZE	PE0ZE
POE	POECR2	—	P1CZEA	P2CZEA	P3CZEA	—	P1CZEB	P2CZEB	P3CZEB
		—	P4CZE	P5CZE	P6CZE	—	—	—	—
POE	ICSR4	—	—	—	POE9F	—	—	POE9E	PIE4
		—	—	—	—	—	—	POE9M[1:0]	
MTU9	TCR	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
MTU10	TCR	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
MTU9	TMDR	—	—	BFB	BFA	MD[3:0]			
MTU10	TMDR	—	—	BFB	BFA	MD[3:0]			
MTU9	TIORH	IOB[3:0]				IOA[3:0]			
MTU9	TIORL	IOD[3:0]				IOC[3:0]			
MTU10	TIORH	IOB[3:0]				IOA[3:0]			
MTU10	TIORL	IOD[3:0]				IOC[3:0]			
MTU9	TIER	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
MTU10	TIER	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
MTUB	TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
MTUB	TGCR	—	BDC	N	P	FB	WF	VF	UF
MTUB	TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
MTUB	TOCR2	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
MTU9	TCNT								
MTU10	TCNT								
MTUB	TCDR								

Table 5.2 List of I/O Registers (Bit Order) (32 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTUB	TDDR								
MTU9	TGRA								
MTU9	TGRB								
MTU10	TGRA								
MTU10	TGRB								
MTUB	TCNTS								
MTUB	TCBR								
MTU9	TGRC								
MTU9	TGRD								
MTU10	TGRC								
MTU10	TGRD								
MTU9	TSR	TCFD	—	—	—	—	—	—	—
MTU10	TSR	TCFD	—	—	—	—	—	—	—
MTUB	TITCR	T3AEN	T3ACOR[2:0]			T4VEN	T4VCOR[2:0]		
MTUB	TITCNT	—	T3ACNT[2:0]			—	T4VCNT[2:0]		
MTUB	TBTER	—	—	—	—	—	—	BTE[1:0]	
MTUB	TDER	—	—	—	—	—	—	—	TDER
MTUB	TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
MTU9	TBTM	—	—	—	—	—	—	TTSB	TTSA
MTU10	TBTM	—	—	—	—	—	—	TTSB	TTSA
MTU10	TADCR	BF[1:0]		—	—	—	—	—	—
		UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
MTU10	TADCORA								
MTU10	TADCORB								
MTU10	TADCOBRA								
MTU10	TADCOBRB								
MTUB	TWCR	CCE	—	—	—	—	—	—	WRE
MTUB	TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0
MTUB	TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
MTUB	TRWER	—	—	—	—	—	—	—	RWE

Table 5.2 List of I/O Registers (Bit Order) (33 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTU6	TCR	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
MTU6	TMDR	—	BFE	BFB	BFA	MD[3:0]			
MTU6	TIORH	IOB[3:0]				IOA[3:0]			
MTU6	TIORL	IOD[3:0]				IOC[3:0]			
MTU6	TIER	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
MTU6	TSR	—	—	—	—	—	—	—	—
MTU6	TCNT								
MTU6	TGRA								
MTU6	TGRB								
MTU6	TGRC								
MTU6	TGRD								
MTU6	TGRE								
MTU6	TGRF								
MTU6	TIER2	—	—	—	—	—	—	TGIEF	TGIEE
MTU6	TBTM	—	—	—	—	—	TTSE	TTSB	T TSA
MTU7	TCR	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]		
MTU7	TMDR	—	—	—	—	MD[3:0]			
MTU7	TIOR	IOB[3:0]				IOA[3:0]			
MTU7	TIER	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
MTU7	TSR	TCFD	—	—	—	—	—	—	—
MTU7	TCNT								
MTU7	TGRA								
MTU7	TGRB								
MTU7	TICCR	—	—	—	—	I2BE	I2AE	I1BE	I1AE
MTU8	TCR	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]		
MTU8	TMDR	—	—	—	—	MD[3:0]			
MTU8	TIOR	IOB[3:0]				IOA[3:0]			
MTU8	TIER	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
MTU8	TSR	TCFD	—	—	—	—	—	—	—
MTU8	TCNT								
MTU8	TGRA								
MTU8	TGRB								

Table 5.2 List of I/O Registers (Bit Order) (34 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
MTU11	TCNTU								
MTU11	TGRU								
MTU11	TCRU	—	—	—	—	—	—	TPSC[1:0]	
MTU11	TIORU	—	—	—	IOC[4:0]				
MTU11	TCNTV								
MTU11	TGRV								
MTU11	TCRV	—	—	—	—	—	—	TPSC[1:0]	
MTU11	TIORV	—	—	—	IOC[4:0]				
MTU11	TCNTW								
MTU11	TGRW								
MTU11	TCRW	—	—	—	—	—	—	TPSC[1:0]	
MTU11	TIORW	—	—	—	IOC[4:0]				
MTU11	TIER	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
MTU11	TSTR	—	—	—	—	—	CSTU5	CSTV5	CSTW5
MTU11	TCNTCMPCLR	—	—	—	—	—	CMPCLR 5U	CMPCLR 5V	CMPCLR 5W
S12AD	ADCSR	ADST	ADCS	—	ADIE	CKS[1:0]		TRGE	EXTRG
S12AD	ADANS	—	—	—	—	—	—	—	—
		ANS[7:0]							
S12AD	ADADS	—	—	—	—	—	—	—	—
		ADS[7:0]							
S12AD	ADADC	—	—	—	—	—	—	ADC[1:0]	
S12AD	ADCER	ADRFMT	—	—	—	—	—	—	—
		—	—	ACE	—	—	—	—	—
S12AD	ADSTRGR	—	—	—	—	ADSTRS[3:0]			
S12AD	ADDR0								
S12AD	ADDR1								
S12AD	ADDR2								
S12AD	ADDR3								
S12AD	ADDR4								
S12AD	ADDR5								
S12AD	ADDR6								

Table 5.2 List of I/O Registers (Bit Order) (35 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
S12AD	ADDR7								
PORT0	DDR	B7 *8	—	B5	—	B3 *7	B2 *7*8	B1 *7*8	B0 *7*8
PORT1	DDR	B7 *7*8	B6	B5 *7*8	B4	B3	B2	B1 *6*7*8	B0 *6*7*8
PORT2	DDR	B7	B6	B5	B4	B3	B2	B1	B0
PORT3	DDR	—	—	—	B4	B3	B2	B1	B0
PORT4	DDR	B7	B6	B5	B4	B3	B2	B1	B0
PORT5	DDR	B7 *6*7*8	B6 *7*8	B5 *8	B4 *8	B3	B2	B1	B0
PORT6	DDR	B7 *7*8	B6 *7*8	B5 *7*8	B4 *7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORT7	DDR	B7 *7*8	B6 *7*8	B5 *7*8	B4 *7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORT8	DDR	—	—	B5 *6*7*8	B4 *6*7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORT9	DDR	B7 *6*7*8	B6 *6*7*8	B5 *6*7*8	B4 *6*7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORTA	DDR	B7	B6	B5	B4	B3	B2	B1	B0
PORTB	DDR	B7	B6	B5	B4	B3	B2	B1	B0
PORTC	DDR	B7 *8	B6 *8	B5 *8	B4 *8	B3	B2	B1	B0
PORTD	DDR	B7	B6	B5	B4	B3	B2	B1	B0
PORTE	DDR	B7 *8	B6 *8	B5 *8	B4 *8	B3 *8	B2 *8	B1 *8	B0 *8
PORTF	DDR	—	—	—	B4 *6*7*8	B3 *6*7*8	B2 *6*7*8	B1 *6*7*8	B0 *6*7*8
PORTG	DDR	B7 *6*7*8	B6 *6*7*8	B5 *6*7*8	B4 *6*7*8	B3 *6*7*8	B2 *6*7*8	B1 *6*7*8	B0 *6*7*8
PORT0	DR	B7 *8	—	B5	—	B3 *7	B2 *7*8	B1 *7*8	B0 *7*8
PORT1	DR	B7 *7*8	B6	B5 *7*8	B4	B3	B2	B1 *6*7*8	B0 *6*7*8
PORT2	DR	B7	B6	B5	B4	B3	B2	B1	B0
PORT3	DR	—	—	—	B4	B3	B2	B1	B0
PORT4	DR	B7	B6	B5	B4	B3	B2	B1	B0
PORT5	DR	B7 *6*7*8	B6 *7*8	B5 *8	B4 *8	—	B2	B1	B0
PORT6	DR	B7 *7*8	B6 *7*8	B5 *7*8	B4 *7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORT7	DR	B7 *7*8	B6 *7*8	B5 *7*8	B4 *7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORT8	DR	—	—	B5 *6*7*8	B4 *6*7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORT9	DR	B7 *6*7*8	B6 *6*7*8	B5 *6*7*8	B4 *6*7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORTA	DR	B7	B6	B5	B4	B3	B2	B1	B0
PORTB	DR	B7	B6	B5	B4	B3	B2	B1	B0

Table 5.2 List of I/O Registers (Bit Order) (36 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
PORTC	DR	B7 *8	B6 *8	B5 *8	B4 *8	B3	B2	B1	B0
PORTD	DR	B7	B6	B5	B4	B3	B2	B1	B0
PORTE	DR	B7 *8	B6 *8	B5 *8	B4 *8	B3 *8	B2 *8	B1 *8	B0 *8
PORTF	DR	—	—	—	B4 *6*7*8	B3 *6*7*8	B2 *6*7*8	B1 *6*7*8	B0 *6*7*8
PORTG	DR	B7 *6*7*8	B6 *6*7*8	B5 *6*7*8	B4 *6*7*8	B3 *6*7*8	B2 *6*7*8	B1 *6*7*8	B0 *6*7*8
PORT0	PORT	B7 *8	—	B5	—	B3 *7	B2 *7*8	B1 *7*8	B0 *7*8
PORT1	PORT	B7 *7*8	B6	B5 *7*8	B4	B3	B2	B1 *6*7*8	B0 *6*7*8
PORT2	PORT	B7	B6	B5	B4	B3	B2	B1	B0
PORT3	PORT	—	—	B5	B4	B3	B2	B1	B0
PORT4	PORT	B7	B6	B5	B4	B3	B2	B1	B0
PORT5	PORT	B7 *6*7*8	B6 *7*8	B5 *8	B4 *8	B3	B2	B1	B0
PORT6	PORT	B7 *7*8	B6 *7*8	B5 *7*8	B4 *7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORT7	PORT	B7 *7*8	B6 *7*8	B5 *7*8	B4 *7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORT8	PORT	—	—	B5 *6*7*8	B4 *6*7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORT9	PORT	B7 *6*7*8	B6 *6*7*8	B5 *6*7*8	B4 *6*7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORTA	PORT	B7	B6	B5	B4	B3	B2	B1	B0
PORTB	PORT	B7	B6	B5	B4	B3	B2	B1	B0
PORTC	PORT	B7 *8	B6 *8	B5 *8	B4 *8	B3	B2	B1	B0
PORTD	PORT	B7	B6	B5	B4	B3	B2	B1	B0
PORTE	PORT	B7 *8	B6 *8	B5 *8	B4 *8	B3 *8	B2 *8	B1 *8	B0 *8
PORTF	PORT	—	—	—	B4 *6*7*8	B3 *6*7*8	B2 *6*7*8	B1 *6*7*8	B0 *6*7*8
PORTG	PORT	B7 *6*7*8	B6 *6*7*8	B5 *6*7*8	B4 *6*7*8	B3 *6*7*8	B2 *6*7*8	B1 *6*7*8	B0 *6*7*8
PORT0	ICR	B7 *8	—	B5	—	B3 *7	B2 *7*8	B1 *7*8	B0 *7*8
PORT1	ICR	B7 *7*8	B6	B5 *7*8	B4	B3	B2	B1 *6*7*8	B0 *6*7*8
PORT2	ICR	B7	B6	B5	B4	B3	B2	B1	B0
PORT3	ICR	—	—	—	B4	B3	B2	B1	B0
PORT4	ICR	B7	B6	B5	B4	B3	B2	B1	B0
PORT5	ICR	B7 *6*7*8	B6 *7*8	B5 *8	B4 *8	B3	B2	B1	B0
PORT6	ICR	B7 *7*8	B6 *7*8	B5 *7*8	B4 *7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORT7	ICR	B7 *7*8	B6 *7*8	B5 *7*8	B4 *7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8

Table 5.2 List of I/O Registers (Bit Order) (37 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
PORT8	ICR	—	—	B5 *6*7*8	B4 *6*7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORT9	ICR	B7 *6*7*8	B6 *6*7*8	B5 *6*7*8	B4 *6*7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORTA	ICR	B7	B6	B5	B4	B3	B2	B1	B0
PORTB	ICR	B7	B6	B5	B4	B3	B2	B1	B0
PORTC	ICR	B7 *8	B6 *8	B5 *8	B4 *8	B3	B2	B1	B0
PORTD	ICR	B7	B6	B5	B4	B3	B2	B1	B0
PORTE	ICR	B7 *8	B6 *8	B5 *8	B4 *8	B3 *8	B2 *8	B1 *8	B0 *8
PORTF	ICR	—	—	—	B4 *6*7*8	B3 *6*7*8	B2 *6*7*8	B1 *6*7*8	B0 *6*7*8
PORTG	ICR	B7 *6*7*8	B6 *6*7*8	B5 *6*7*8	B4 *6*7*8	B3 *6*7*8	B2 *6*7*8	B1 *6*7*8	B0 *6*7*8
PORT0	ODR	B7 *8	—	B5	—	B3 *7	B2 *7*8	B1 *7*8	B0 *7*8
PORT1	ODR	B7 *7*8	B6	B5 *7*8	B4	B3	B2	B1 *6*7*8	B0 *6*7*8
PORT2	ODR	B7	B6	B5	B4	B3	B2	B1	B0
PORT3	ODR	—	—	—	B4	B3	B2	B1	B0
PORTC	ODR	B7 *8	B6 *8	B5 *8	B4 *8	B3	B2	B1	B0
PORT9	PCR	B7 *6*7*8	B6 *6*7*8	B5 *6*7*8	B4 *6*7*8	B3 *7*8	B2 *7*8	B1 *7*8	B0 *7*8
PORTA	PCR	B7	B6	B5	B4	B3	B2	B1	B0
PORTB	PCR	B7	B6	B5	B4	B3	B2	B1	B0
PORTC	PCR	B7 *8	B6 *8	B5 *8	B4 *8	B3	B2	B1	B0
PORTD	PCR	B7	B6	B5	B4	B3	B2	B1	B0
PORTE	PCR	B7 *8	B6 *8	B5 *8	B4 *8	B3 *8	B2 *8	B1 *8	B0 *8
PORTG	PCR	B7 *6*7*8	B6 *6*7*8	B5 *6*7*8	B4 *6*7*8	B3 *6*7*8	B2 *6*7*8	B1 *6*7*8	B0 *6*7*8
IOPORT	PF0CSE	CS7E	CS6E	CS5E	CS4E	CS3E *8	CS2E *8	CS1E *8	CS0E *8
IOPORT	PF1CSS	CS7S[1:0] *7*8		CS6S[1:0] *7*8		CS5S[1:0] *7*8		CS4S[1:0] *7*8	
IOPORT	PF2CSS	CS3S[1:0] *7*8		CS2S[1:0] *7*8		CS1S[1:0] *7*8		—	CS0S *7*8
IOPORT	PF3BUS	A23E *8	A22E *8	A21E *8	A20E *8	A19E	A18E	A17E	A16E
IOPORT	PF4BUS	A15E	A14E	A13E	A12E	A11E	A10E	ADRLE[1:0]	
IOPORT	PF5BUS	WR32BC32E *6*7*8	WR1BC1E *8	DH32E *6*7*8	DHE *8	—	—	ADRHMS *7*8	—
IOPORT	PF6BUS	SDCLKE *7*8	DQM1E *7*8	—	MDSDE *7*8	—	—	WAITS[1:0] *8	
IOPORT	PF7DMA	EDMA1S[1:0] *7*8		EDMA0S[1:0] *7*8		—	—	—	—
IOPORT	PF8IRQ	ITS15 *8	—	ITS13	—	ITS11 *7	ITS10 *7*8	ITS9 *7*8	ITS8 *7*8

Table 5.2 List of I/O Registers (Bit Order) (38 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	
IOPORT	PF9IRQ	ITS7 *7*8	ITS6 *8	ITS5 *7*8	ITS4	ITS3	ITS2	ITS1 *6*7*8	ITS0 *6*7*8	
IOPORT	PFAADC	—	—	—	—	—	—	—	ADTRG0S *8	
IOPORT	PFBTMR	—	—	—	—	TMR3S *6*7*8	TMR2S *7*8	TMR1S *7*8	TMR0S *7*8	
IOPORT	PFCMTU	TCLKS *8	MTUS6 *6*7*8	MTUS5 *8	MTUS4 *7*8	MTUS3 *7*8	MTUS2 *7*8	—	—	
IOPORT	PFDMTU	TCLKS	MTUS6 *8	—	—	—	—	—	—	
IOPORT	PFENET	EE *8	—	—	PHYMODE *8	ENETE3 *8	ENETE2 *8	ENETE1 *8	ENETE0 *8	
IOPORT	PFFSCI	—	SCI6S *7*8	—	—	SCI3S *7*8	SCI2S	SCI1S *6*7*8	—	
IOPORT	PFGSPI	SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	RSPIS *8	
IOPORT	PFHSPI	SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	RSPIS *8	
IOPORT	PFJCAN	—	—	—	—	—	—	—	CAN0E	
IOPORT	PFKUSB	—	—	—	USBE	PDHZS	PUPHZS	USBMD[1:0]		
IOPORT	PFLUSB	—	—	—	USBE *6*7*8	PDHZS *6*7*8	PUPHZS *6*7*8	USBMD[1:0] *6*7*8		
IOPORT	PFMPOE	POE7E *8	POE6E *8	POE5E *8	POE4E *8	POE3E *8	POE2E *8	POE1E *8	POE0E *8	
IOPORT	PFNPOE	—	—	—	—	—	—	POE9E *8	POE8E *8	
SYSTEM	DPSBYCR	DPSBY	IOKEEP	RAMCUT2	RAMCUT1	—	—	—	RAMCUT0	
SYSTEM	DPSWCR	—	—	WTSTS[5:0]						—
SYSTEM	DPSIER	DNMIE	DUSBIE	DRTCE	DLVDE	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E	
SYSTEM	DPSIFR	DNMIF	DUSBF	DRTCF	DLVDF	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F	
SYSTEM	DPSIEGR	DNMIEG	—	—	—	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG	
SYSTEM	RSTSR	DPSRSTF	—	—	—	—	LVD2F	LVD1F	PORF	
FLASH	FWEPROR	—	—	—	—	—	—	FLWE[1:0]		
SYSTEM	SUBOSCCR	—	—	—	—	—	—	—	SUBSTOP	
SYSTEM	LVDKEYR	—	—	—	—	—	—	—	—	
SYSTEM	LVD2CR	LVD2E	LVD2RI	—	—	LVD1E	LVD1RI	—	—	
SYSTEM	DPSBKR0	—	—	—	—	—	—	—	—	
SYSTEM	DPSBKR1	—	—	—	—	—	—	—	—	
SYSTEM	DPSBKR2	—	—	—	—	—	—	—	—	
SYSTEM	DPSBKR3	—	—	—	—	—	—	—	—	
SYSTEM	DPSBKR4	—	—	—	—	—	—	—	—	
SYSTEM	DPSBKR5	—	—	—	—	—	—	—	—	
SYSTEM	DPSBKR6	—	—	—	—	—	—	—	—	
SYSTEM	DPSBKR7	—	—	—	—	—	—	—	—	
SYSTEM	DPSBKR8	—	—	—	—	—	—	—	—	
SYSTEM	DPSBKR9	—	—	—	—	—	—	—	—	
SYSTEM	DPSBKR10	—	—	—	—	—	—	—	—	
SYSTEM	DPSBKR11	—	—	—	—	—	—	—	—	

Table 5.2 List of I/O Registers (Bit Order) (39 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
SYSTEM	DPSBKR12								
SYSTEM	DPSBKR13								
SYSTEM	DPSBKR14								
SYSTEM	DPSBKR15								
SYSTEM	DPSBKR16								
SYSTEM	DPSBKR17								
SYSTEM	DPSBKR18								
SYSTEM	DPSBKR19								
SYSTEM	DPSBKR20								
SYSTEM	DPSBKR21								
SYSTEM	DPSBKR22								
SYSTEM	DPSBKR23								
SYSTEM	DPSBKR24								
SYSTEM	DPSBKR25								
SYSTEM	DPSBKR26								
SYSTEM	DPSBKR27								
SYSTEM	DPSBKR28								
SYSTEM	DPSBKR29								
SYSTEM	DPSBKR30								
SYSTEM	DPSBKR31								
RTC	R64CNT	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
RTC	RSECCNT	—	SEC10[2:0]			SEC1[3:0]			
RTC	RMINCNT	—	MIN10[2:0]			MIN1[3:0]			
RTC	RHRCNT	—	—	HOUR10[1:0]		HOUR1[3:0]			
RTC	RWKCNT	—	—	—	—	—	DAY[2:0]		
RTC	RDAYCNT	—	—	DAY10[1:0]		DAY1[3:0]			
RTC	RMONCNT	—	—	—	MON10	MON1[3:0]			
RTC	RYRCNT	YEAR1000[3:0]				YEAR100[3:0]			
		YEAR10[3:0]				YEAR1[3:0]			
RTC	RSECAR	ENB	SEC10[2:0]			SEC1[3:0]			
RTC	RMINAR	ENB	MIN10[2:0]			MIN1[3:0]			
RTC	RHRAR	ENB	—	HOUR10[1:0]		HOUR1[3:0]			
RTC	RWKAR	ENB	—	—	—	—	DAY[2:0]		
RTC	RDAYAR	ENB	—	DAY10[1:0]		DAY1[3:0]			
RTC	RMONAR	ENB	—	—	MON10	MON1[3:0]			
RTC	RYRAR	YEAR1000[3:0]				YEAR100[3:0]			
		YEAR10[3:0]				YEAR1[3:0]			
RTC	RYRAREN	ENB	—	—	—	—	—	—	—
RTC	RRCR1	—	PES[2:0]			—	PIE	CIE	AIE
RTC	RRCR2	—	—	—	—	RTCOE	ADJ	RESET	START

Table 5.2 List of I/O Registers (Bit Order) (40 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	
CAN0	MB.ID	IDE	RTR	—	SID[10:0]					
		SID[10:0]							EID[17:0]	
		EID[17:0]								
		EID[17:0]								
	MB.DLC	—	—	—	—	—	—	—	—	—
		DLC[3:0]								
	MB.DATA 0 to 7									
MB.TS	TSH[7:0]									
	TSL[7:0]									
CAN0	MKR0	—	—	—	SID[10:0]					
		SID[10:0]							EID[17:0]	
		EID[17:0]								
		EID[17:0]								
CAN0	MKR1	—	—	—	SID[10:0]					
		SID[10:0]							EID[17:0]	
		EID[17:0]								
		EID[17:0]								
CAN0	MKR2	—	—	—	SID[10:0]					
		SID[10:0]							EID[17:0]	
		EID[17:0]								
		EID[17:0]								
CAN0	MKR3	—	—	—	SID[10:0]					
		SID[10:0]							EID[17:0]	
		EID[17:0]								
		EID[17:0]								
CAN0	MKR4	—	—	—	SID[10:0]					
		SID[10:0]							EID[17:0]	
		EID[17:0]								
		EID[17:0]								
CAN0	MKR5	—	—	—	SID[10:0]					
		SID[10:0]							EID[17:0]	
		EID[17:0]								
		EID[17:0]								
CAN0	MKR6	—	—	—	SID[10:0]					
		SID[10:0]							EID[17:0]	
		EID[17:0]								
		EID[17:0]								
CAN0	MKR7	—	—	—	SID[10:0]					
		SID[10:0]							EID[17:0]	
		EID[17:0]								
		EID[17:0]								

Table 5.2 List of I/O Registers (Bit Order) (41 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	
CAN0	FIDCR0	IDE	RTR	—	SID[10:0]					
		SID[10:0]							EID[17:0]	
		EID[17:0]								
		EID[17:0]								
CAN0	FIDCR1	IDE	RTR	—	SID[10:0]					
		SID[10:0]							EID[17:0]	
		EID[17:0]								
		EID[17:0]								
CAN0	MKIVLR									
CAN0	MIER	—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
CAN0	MCTLTX	TRMREQ	RECREQ	—	ONESHOT	—	TRMABT	TRMACTIVE	SENTDATA	
	MCLRFX	TRMREQ	RECREQ	—	ONESHOT	—	MSGLOST	INVALIDDATA	NEWDATA	
CAN0	CTRLR	—	—	RBOC	BOM[1:0]		SLPM	CANM[1:0]		
		TSPS[1:0]		TSRC	TPM	MLM	IDFM[1:0]		MBM	
CAN0	STR	—	RECST	TRMST	BOST	EPST	SLPST	HLTST	RSTST	
		EST	TABST	FMLST	NMLST	TFST	RFST	SDST	NDST	
CAN0	BCR	TSEG1[3:0]				—	—	BRP[9:0]		
		BRP[9:0]								
		—	—	SJW[1:0]		—	TSEG2[2:0]			
		—	—	—	—	—	—	—	—	—
CAN0	RFCR	RFEST	RFWST	RFFST	RFMLF	RFUST[2:0]			RFE	
CAN0	RFPCR	—	—	—	—	—	—	—	—	
CAN0	TFCR	TFEST	TFFST	—	—	TFUST[2:0]			TFE	
CAN0	TFPCR	—	—	—	—	—	—	—	—	
CAN0	EIER	BLIE	OLIE	ORIE	BORIE	BOEIE	EPIE	EWIE	BEIE	
CAN0	EIFR	BLIF	OLIF	ORIF	BORIF	BOEIF	EPIF	EWIF	BEIF	
CAN0	RECR	—	—	—	—	—	—	—	—	
CAN0	TECR	—	—	—	—	—	—	—	—	
CAN0	ECSR	EDPM	ADEF	BE0F	BE1F	CEF	AEF	FEF	SEF	
CAN0	CSSR	—	—	—	—	—	—	—	—	
CAN0	MSSR	SEST	—	—	MBNST[4:0]					
CAN0	MSMR	—	—	—	—	—	—	MBSM[1:0]		
CAN0	TSR	—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
CAN0	AFSR	—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
CAN0	TCR	—	—	—	—	—	TSTM[1:0]		TSTE	
USB0	SYSCFG	—	—	—	—	—	SCKE	—	—	
		—	DCFM	DRPD	DPRPU	—	—	—	USBE	

Table 5.2 List of I/O Registers (Bit Order) (42 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
USB0	SYSSTS0	OVCMON[1:0]		—	—	—	—	—	—
		—	HTACT	—	—	—	IDMON	LNST[1:0]	
USB0	DVSTCTR0	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP
		RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
USB0	CFIFO	L[7:0]							
		H[7:0]							
USB0	D0FIFO	L[7:0]							
		H[7:0]							
USB0	D1FIFO	L[7:0]							
		H[7:0]							
USB0	CFIFOSEL	RCNT	REW	—	—	—	MBW	—	BIGEND
		—	—	ISEL	—	CURPIPE[3:0]			
USB0	CFIFOCTR	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]
		DTLN[8:0]							
USB0	D0FIFOSEL	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND
		—	—	—	—	CURPIPE[3:0]			
USB0	D0FIFOCTR	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]
		DTLN[8:0]							
USB0	D1FIFOSEL	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND
		—	—	—	—	CURPIPE[3:0]			
USB0	D1FIFOCTR	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]
		DTLN[8:0]							
USB0	INTENB0	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE
		—	—	—	—	—	—	—	—
USB0	INTENB1	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—
		—	EOFERRE	SIGNE	SACKE	—	—	—	—
USB0	BRDYENB	—	—	—	—	—	—	PIPE9BRDYE	PIPE8BRDYE
		PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	PIPE3BRDYE	PIPE2BRDYE	PIPE1BRDYE	PIPE0BRDYE
USB0	NRDYENB	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE
		PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
USB0	BEMPENB	—	—	—	—	—	—	PIPE9BEMPE	PIPE8BEMPE
		PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
USB0	SOFCFG	—	—	—	—	—	—	—	TRNENSEL
		—	BRDYM	—	EDGESTS	—	—	—	—
USB0	INTSTS0	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY
		VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
USB0	INTSTS1	OVRCCR	BCHG	—	DTCH	ATTCH	—	—	—
		—	EOFERR	SIGN	SACK	—	—	—	—
USB0	BRDYSTS	—	—	—	—	—	—	PIPE9BRDY	PIPE8BRDY
		PIPE7BRDY	PIPE6BRDY	PIPE5BRDY	PIPE4BRDY	PIPE3BRDY	PIPE2BRDY	PIPE1BRDY	PIPE0BRDY
USB0	NRDYSTS	—	—	—	—	—	—	PIPE9NRDY	PIPE8NRDY
		PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
USB0	BEMPSTS	—	—	—	—	—	—	PIPE9BEMP	PIPE8BEMP
		PIPE7BEMP	PIPE6BEMP	PIPE5BEMP	PIPE4BEMP	PIPE3BEMP	PIPE2BEMP	PIPE1BEMP	PIPE0BEMP

Table 5.2 List of I/O Registers (Bit Order) (43 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
USB0	FRMNUM	OVRN	CRCE	—	—	—	FRNM[10:0]		
		FRNM[10:0]							
USB0	DVCHGR	DVCHG	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
USB0	USBADDR	—	—	—	—	STSRECOV[3:0]			
		—	USBADDR[6:0]						
USB0	USBREQ	BREQUEST[7:0]							
		BMREQUESTTYPE[7:0]							
USB0	USBVAL								
USB0	USBINDX								
USB0	USBLENG								
USB0	DCPCFG	—	—	—	—	—	—	—	—
		SHTNAK	—	—	DIR	—	—	—	—
USB0	DCPMAXP	DEVSEL[3:0]				—	—	—	—
		—	MXPS[6:0]						
USB0	DCPCTR	BSTS	SUREQ	—	—	SUREQCLR	—	—	SQCLR
		SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
USB0	PIPESEL	—	—	—	—	—	—	—	—
		—	—	—	—	PIPESEL[3:0]			
USB0	PIPECFG	TYPE[1:0]		—	—	—	BFRE	DBLB	—
		SHTNAK	—	—	DIR	EPNUM[3:0]			
USB0	PIPEMAXP	DEVSEL[3:0]				—	—	—	MXPS[8:0]
		MXPS[8:0]							
USB0	PIPEPERI	—	—	—	IFIS	—	—	—	—
		—	—	—	—	—	IITV[2:0]		
USB0	PIPE1CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB0	PIPE2CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB0	PIPE3CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB0	PIPE4CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB0	PIPE5CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB0	PIPE6CTR	BSTS	—	—	—	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB0	PIPE7CTR	BSTS	—	—	—	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB0	PIPE8CTR	BSTS	—	—	—	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	

Table 5.2 List of I/O Registers (Bit Order) (44 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
USB0	PIPE9CTR	BSTS	—	—	—	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB0	PIPE1TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
USB0	PIPE1TRN								
USB0	PIPE2TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
USB0	PIPE2TRN								
USB0	PIPE3TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
USB0	PIPE3TRN								
USB0	PIPE4TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
USB0	PIPE4TRN								
USB0	PIPE5TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
USB0	PIPE5TRN								
USB0	DEVADD0	—	—	—	—	—	—	—	—
		USBSPD[1:0]		—	—	—	—	—	—
USB0	DEVADD1	—	—	—	—	—	—	—	—
		USBSPD[1:0]		—	—	—	—	—	—
USB0	DEVADD2	—	—	—	—	—	—	—	—
		USBSPD[1:0]		—	—	—	—	—	—
USB0	DEVADD3	—	—	—	—	—	—	—	—
		USBSPD[1:0]		—	—	—	—	—	—
USB0	DEVADD4	—	—	—	—	—	—	—	—
		USBSPD[1:0]		—	—	—	—	—	—
USB0	DEVADD5	—	—	—	—	—	—	—	—
		USBSPD[1:0]		—	—	—	—	—	—
USB1	SYSCFG	—	—	—	—	—	—	SCKE	—
		—	DCFM	DRPD	DPRPU	—	—	—	USBE
USB1	SYSSTS0	OVCMON[1:0]		—	—	—	—	—	—
		—	HTACT	—	—	—	—	IDMON	LNST[1:0]
USB1	DVSTCTR0	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP
		RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
USB1	CFIFO	L[7:0]							
		H[7:0]							
USB1	D0FIFO	L[7:0]							
		H[7:0]							

Table 5.2 List of I/O Registers (Bit Order) (45 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
USB1	D1FIFO	L[7:0]							
		H[7:0]							
USB1	CFIFOSEL	RCNT	REW	—	—	—	MBW	—	BIGEND
		—	—	ISEL	—	CURPIPE[3:0]			
USB1	CFIFOCTR	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]
		DTLN[8:0]							
USB1	D0FIFOSEL	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND
		—	—	—	—	CURPIPE[3:0]			
USB1	D0FIFOCTR	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]
		DTLN[8:0]							
USB1	D1FIFOSEL	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND
		—	—	—	—	CURPIPE[3:0]			
USB1	D1FIFOCTR	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]
		DTLN[8:0]							
USB1	INTENB0	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE
		—	—	—	—	—	—	—	—
USB1	INTENB1	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—
		—	EOFERRE	SIGNE	SACKE	—	—	—	—
USB1	BRDYENB	—	—	—	—	—	—	PIPE9BRDYE	PIPE8BRDYE
		PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	PIPE3BRDYE	PIPE2BRDYE	PIPE1BRDYE	PIPE0BRDYE
USB1	NRDYENB	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE
		PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
USB1	BEMPENB	—	—	—	—	—	—	PIPE9BEMPE	PIPE8BEMPE
		PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
USB1	SOFCFG	—	—	—	—	—	—	—	TRNENSEL
		—	BRDYM	—	EDGESTS	—	—	—	—
USB1	INTSTS0	VBINT	RESM	SOFR	DVST	CTRTR	BEMP	NRDY	BRDY
		VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
USB1	INTSTS1	OVRCCR	BCHG	—	DTCH	ATTCH	—	—	—
		—	EOFERR	SIGN	SACK	—	—	—	—
USB1	BRDYSTS	—	—	—	—	—	—	PIPE9BRDY	PIPE8BRDY
		PIPE7BRDY	PIPE6BRDY	PIPE5BRDY	PIPE4BRDY	PIPE3BRDY	PIPE2BRDY	PIPE1BRDY	PIPE0BRDY
USB1	NRDYSTS	—	—	—	—	—	—	PIPE9NRDY	PIPE8NRDY
		PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
USB1	BEMPSTS	—	—	—	—	—	—	PIPE9BEMP	PIPE8BEMP
		PIPE7BEMP	PIPE6BEMP	PIPE5BEMP	PIPE4BEMP	PIPE3BEMP	PIPE2BEMP	PIPE1BEMP	PIPE0BEMP
USB1	FRMNUM	OVRN	CRCE	—	—	—	FRNM[10:0]		
		FRNM[10:0]							
USB1	DVCHGR	DVCHG	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
USB1	USBADDR	—	—	—	—	STSRECOV[3:0]			
		—	USBADDR[6:0]						
USB1	USBREQ	BREQUEST[7:0]							
		BMREQUESTTYPE[7:0]							

Table 5.2 List of I/O Registers (Bit Order) (46 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
USB1	USBVAL								
USB1	USBINDX								
USB1	USBLENG								
USB1	DCPCFG	—	—	—	—	—	—	—	—
		SHTNAK	—	—	DIR	—	—	—	—
USB1	DCPMAXP	DEVSEL[3:0]				—	—	—	—
		—	MXPS[6:0]						
USB1	DPCCTR	BSTS	SUREQ	—	—	SUREQCLR	—	—	SQCLR
		SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
USB1	PIPESEL	—	—	—	—	—	—	—	—
		—	—	—	—	PIPESEL[3:0]			
USB1	PIPECFG	TYPE[1:0]		—	—	—	BFRE	DBLB	—
		SHTNAK	—	—	DIR	EPNUM[3:0]			
USB1	PIPEMAXP	DEVSEL[3:0]				—	—	—	MXPS[8:0]
		MXPS[8:0]							
USB1	PIPEPERI	—	—	—	IFIS	—	—	—	—
		—	—	—	—	—	IITV[2:0]		
USB1	PIPE1CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB1	PIPE2CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB1	PIPE3CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB1	PIPE4CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB1	PIPE5CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB1	PIPE6CTR	BSTS	—	—	—	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB1	PIPE7CTR	BSTS	—	—	—	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB1	PIPE8CTR	BSTS	—	—	—	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB1	PIPE9CTR	BSTS	—	—	—	—	—	ACLRM	SQCLR
		SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
USB1	PIPE1TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
USB1	PIPE1TRN								
USB1	PIPE2TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—

Table 5.2 List of I/O Registers (Bit Order) (47 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
USB1	PIPE2TRN								
USB1	PIPE3TRE	—	—	—	—	—	—	TRENB	TRCLR
USB1	PIPE3TRN								
USB1	PIPE4TRE	—	—	—	—	—	—	TRENB	TRCLR
USB1	PIPE4TRN								
USB1	PIPE5TRE	—	—	—	—	—	—	TRENB	TRCLR
USB1	PIPE5TRN								
USB1	DEVADD0	—	—	—	—	—	—	—	—
		USBSPD[1:0]		—	—	—	—	—	—
USB1	DEVADD1	—	—	—	—	—	—	—	—
		USBSPD[1:0]		—	—	—	—	—	—
USB1	DEVADD2	—	—	—	—	—	—	—	—
		USBSPD[1:0]		—	—	—	—	—	—
USB1	DEVADD3	—	—	—	—	—	—	—	—
		USBSPD[1:0]		—	—	—	—	—	—
USB1	DEVADD4	—	—	—	—	—	—	—	—
		USBSPD[1:0]		—	—	—	—	—	—
USB1	DEVADD5	—	—	—	—	—	—	—	—
		USBSPD[1:0]		—	—	—	—	—	—
USB	DPUSR0R	DVSTS1	—	DOVCB1	DOVCA1	—	—	DM1	DP1
		DVBSTS0	—	DOVCB0	DOVCA0	—	—	DM0	DP0
		—	—	—	FIXPHY1	—	—	—	SRPC1
		—	—	—	FIXPHY0	—	—	—	SRPC0
USB	DPUSR1R	DVBINT1	—	DOVRCRB1	DOVRCRA1	—	—	DMINT1	DPINT1
		DVBINT0	—	DOVRCRB0	DOVRCRA0	—	—	DMINT0	DPINT0
		DVBSE1	—	DOVRCRBE1	DOVRCRAE1	—	—	DMINTE1	DPINTE1
		DVBSE0	—	DOVRCRBE0	DOVRCRAE0	—	—	DMINTE0	DPINTE0
EDMAC	EDMR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	DE	DL[1:0]		—	—	—	SWR
EDMAC	EDTRR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	TR

Table 5.2 List of I/O Registers (Bit Order) (48 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
EDMAC	EDRRR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	RR
EDMAC	TDLAR								
EDMAC	RDLAR								
EDMAC	EESR	—	TWB	—	—	—	TABT	RABT	RFCOF
		ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
		—	—	—	—	CND	DLC	CD	TRO
		RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
EDMAC	EESIPR	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOFIP
		ADEIP	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
		—	—	—	—	CNDIP	DLCIP	CDIP	TROIP
		RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP
EDMAC	TRSCER	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CNDCE	DLCCE	CDCE	TROCE
		RMAFCE	—	—	RRFCE	RTLFCE	RTSFCE	PRECE	CERFCE
EDMAC	RMFCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		MFC[15:0]							
		MFC[15:0]							
EDMAC	TFTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	TFT[10:0]		
		TFT[10:0]							
EDMAC	FDR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	TFD[4:0]			—	—
		—	—	—	RFD[4:0]			—	—
EDMAC	RMCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	RNC	RNR
EDMAC	TFUCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		UNDER[15:0]							
		UNDER[15:0]							

Table 5.2 List of I/O Registers (Bit Order) (49 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	
EDMAC	RFOCR	—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
		OVER[15:0]								
		OVER[15:0]								
EDMAC	IOSR	—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	ELB	
EDMAC	FCFTR	—	—	—	—	—	—	—	—	
		—	—	—	—	—	RFFO[2:0]		—	
		—	—	—	—	—	—	—	—	
		—	—	—	—	—	RFDO[2:0]		—	
EDMAC	RPADIR	—	—	—	—	—	—	—	—	
		—	—	—	—	—	PADS[1:0]		—	
		—	—	—	—	—	—	—	—	
		—	—	PADR[5:0]						—
EDMAC	TRIMD	—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
		—	—	—	TIM	—	—	—	TIS	
EDMAC	RBWAR									
EDMAC	RDFAR									
EDMAC	TBRAR									
EDMAC	TDFAR									
ETHERC	ECMR	—	—	—	—	—	—	—	—	
		—	—	—	TPC	ZPE	PFR	RXF	TXF	
		—	—	—	PRCEF	—	—	MPDE	—	
		—	RE	TE	—	ILB	RTM	DM	PRM	
ETHERC	RFLR	—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
		—	—	—	—	RFL[11:0]				
		RFL[11:0]								

Table 5.2 List of I/O Registers (Bit Order) (50 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ETHERC	ECSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	BFR	PSRTO	—	LCHNG	MPD	ICD
ETHERC	ECSIPR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	BFSIPR	PSRTOIP	—	LCHNGIP	MPDIP	ICDIP
ETHERC	PIR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	MDI	MDO	MMD	MDC
ETHERC	PSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	LMON
ETHERC	RDMLR	—	—	—	—	—	—	—	—
		—	—	—	—	RMD[19:0]			
		RMD[19:0]				RMD[19:0]			
		RMD[19:0]				RMD[19:0]			
ETHERC	IPGR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	IPG[4:0]				
ETHERC	APR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		AP[15:0]				AP[15:0]			
		AP[15:0]				AP[15:0]			
ETHERC	MPR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		MP[15:0]				MP[15:0]			
		MP[15:0]				MP[15:0]			
ETHERC	RFCF	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RPAUSE[7:0]				RPAUSE[7:0]			
ETHERC	TPAUSER	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		TPAUSE[15:0]				TPAUSE[15:0]			
		TPAUSE[15:0]				TPAUSE[15:0]			
ETHERC	TPAUSECR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		TXP[7:0]				TXP[7:0]			

Table 5.2 List of I/O Registers (Bit Order) (51 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	
ETHERC	BCFRR	—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
		BCF[15:0]								
		BCF[15:0]								
ETHERC	MAHR	MA[47:0]								
		MA[47:0]								
		MA[47:0]								
		MA[47:0]								
ETHERC	MALR	—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	—	
		MA[15:0]								
		MA[15:0]								
ETHERC	TROCR									
ETHERC	CDCR									
ETHERC	LCCR									
ETHERC	CNDCR									
ETHERC	CEFCR									
ETHERC	FRECR									
ETHERC	TSFRCR									
ETHERC	TLFRCR									

Table 5.2 List of I/O Registers (Bit Order) (52 / 52)

Module Abbreviation	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
ETHERC	RFCR								
ETHERC	MAFCR								
FLASH	FMODR	—	—	—	FRDMD	—	—	—	—
FLASH	FASTAT	ROMAE	—	—	CMDLK	DFLAE	—	DFLRPE	DFLWPE
FLASH	FAEINT	ROMAEIE	—	—	CMDLKIE	DFLAEIE	—	DFLRPEIE	DFLWPEIE
FLASH	FRDYIE	—	—	—	—	—	—	—	FRDYIE
FLASH	DFLRE0	KEY[7:0]							
		DBRE07	DBRE06	DBRE05	DBRE04	DBRE03	DBRE02	DBRE01	DBRE00
FLASH	DFLRE1	KEY[7:0]							
		DBRE15	DBRE14	DBRE13	DBRE12	DBRE11	DBRE10	DBRE09	DBRE08
FLASH	DFLWE0	KEY[7:0]							
		DBWE07	DBWE06	DBWE05	DBWE04	DBWE03	DBWE02	DBWE01	DBWE00
FLASH	DFLWE1	KEY[7:0]							
		DBWE15	DBWE14	DBWE13	DBWE12	DBWE11	DBWE10	DBWE09	DBWE08
FLASH	FCURAME	KEY[7:0]							
		—	—	—	—	—	—	—	FCRME
FLASH	FSTATR0	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	—	ERSSPD	PRGSPD
FLASH	FSTATR1	FCUERR	—	—	FLOCKST	—	—	—	—
FLASH	FENTRYR	FEKEY[7:0]							
		FENTRYD	—	—	—	—	—	—	FENTRY0
FLASH	FPROTR	FPKEY[7:0]							
		—	—	—	—	—	—	—	FPROTCN
FLASH	FRESETR	FRKEY[7:0]							
		—	—	—	—	—	—	—	FRESET
FLASH	FCMDR	CMDR[7:0]							
		PCMDR[7:0]							
FLASH	FCPSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ESUSPMD
FLASH	DFLBCCNT	—	—	—	—	—	BCADR[7:0]		
		BCADR[7:0]						—	—
FLASH	FPESTAT	—	—	—	—	—	—	—	—
		PEERRST[7:0]							
FLASH	DFLBCSTAT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	BCST
FLASH	PCKAR	—	—	—	—	—	—	—	—
		PCKA[7:0]							

Note: In this section, the I/O port related registers (0008 C000h to 0008 C117h) indicate the bit configuration of the 176-pin LFBGA version. As the configuration of registers and bits differs depending on a package, see section 17, I/O Ports, for details.

- Note 1. When the same output trigger is specified for pulse output groups 2 and 3 by the PPG0.PCR setting, the PPG0.NDRH address is 000881ECh. When different output triggers are specified, the PPG0.NDRH addresses for pulse output groups 2 and 3 are 000881EEh and 000881ECh, respectively.
- Note 2. When the same output trigger is specified for pulse output groups 0 and 1 by the PPG0.PCR setting, the PPG0.NDRL address is 000881EDh. When different output triggers are specified, the PPG0.NDRL addresses for pulse output groups 0 and 1 are 000881EFh and 000881EDh, respectively.
- Note 3. When the same output trigger is specified for pulse output groups 6 and 7 by the PPG1.PCR setting, the PPG1.NDRH address is 000881FCh. When different output triggers are specified, the PPG1.NDRH addresses for pulse output groups 6 and 7 are 000881FEh and 000881FCh, respectively.
- Note 4. When the same output trigger is specified for pulse output groups 4 and 5 by the PPG1.PCR setting, the PPG1.NDRL address is 000881FDh. When different output triggers are specified, the PPG1.NDRL addresses for pulse output groups 4 and 5 are 000881FFh and 000881FDh, respectively.
- Note 5. This shows the bit configuration when ADDR.PDSEL = 0 (flushed at the LSB end). For details, refer to section 35, 10-Bit A/D Converter (ADa).
- Note 6. Not provided in the 145-pin TFLGA version and 144-pin LQFP version.
- Note 7. Not provided in the 100-pin LQFP version.
- Note 8. Not provided in the 85-pin TFLGA version.

6. Resets

6.1 Overview

There are six types of resets: a pin reset, power-on reset, voltage-monitoring reset, deep software standby reset, independent watchdog timer reset, and watchdog timer reset. Table 6.1 shows the reset names and sources.

Table 6.1 Reset Names and Sources

Reset Name	Source
Pin reset	Voltage input to the RES# pin is driven low.
Power-on reset	Vcc rises or falls (voltage detection: V _{POR})
Voltage-monitoring reset	Vcc falls (voltage detection: V _{det1} and V _{det2})*
Deep software standby reset	Deep software standby mode is canceled by an interrupt.
Independent watchdog timer reset	The independent watchdog timer underflows.
Watchdog timer reset	The watchdog timer overflows.

Note: For the voltages to be monitored (V_{det1}, V_{det2}, and V_{POR}), see section 7, Voltage Detection Circuit (LVD) and section 41, Electrical Characteristics.

The internal state and pins are initialized by a reset. Figure 6.1 shows the reset targets to be initialized.

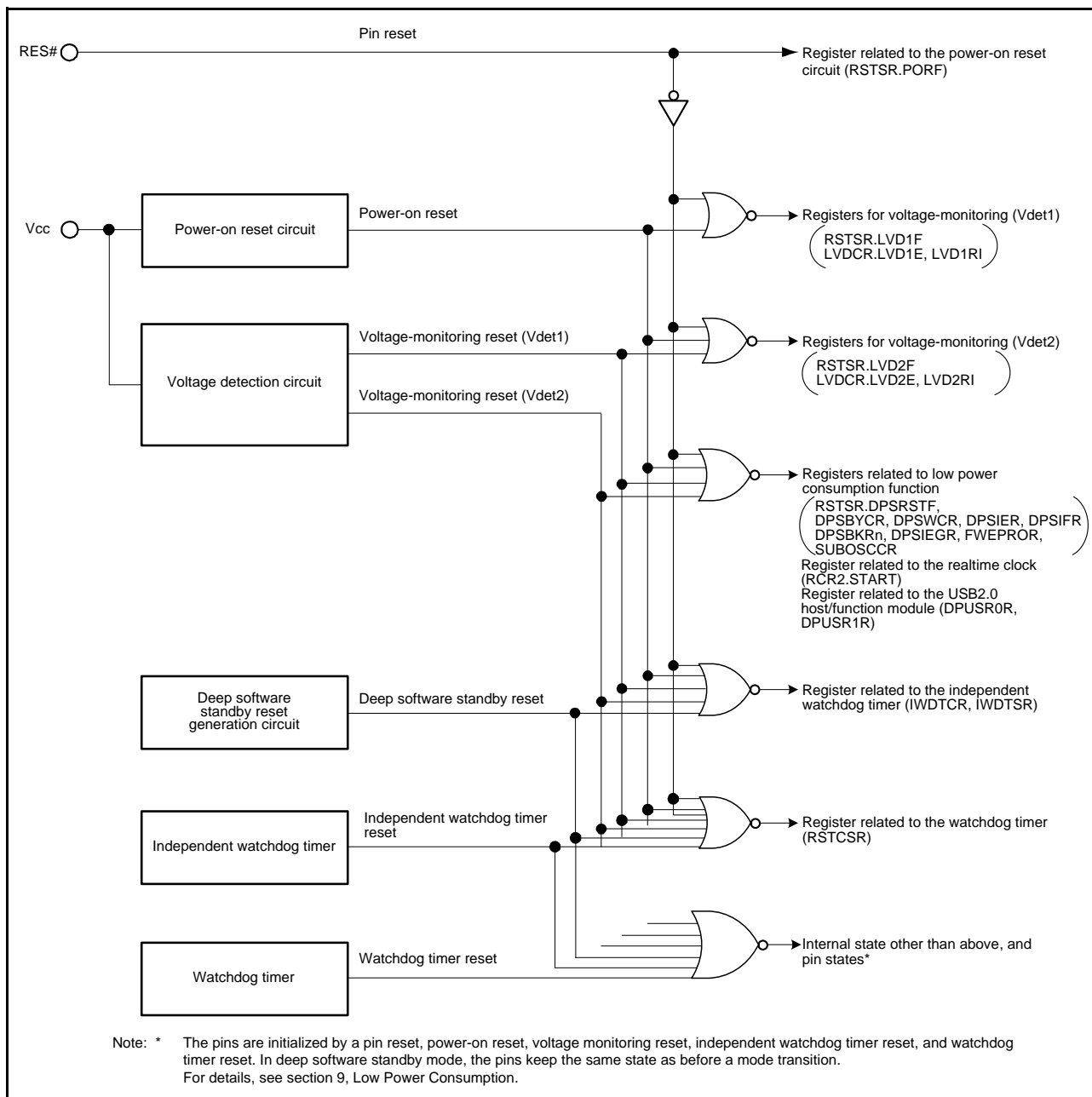


Figure 6.1 Block Diagram of Reset Circuit

Table 6.2 Targets to be Initialized by Each Reset Type

Targets to be Initialized	Reset Type						
	Pin Reset	Power-On Reset	Voltage-Monitoring Reset		Deep Software Standby Reset	Independent Watchdog Timer Reset	Watchdog Timer Reset
			Vdet1	Vdet2			
Register related to the power-on reset circuit RSTSR.PORF	√	—	—	—	—	—	—
Registers related to the voltage monitor function (Vdet1) RSTSR.LVD1F, LVDCR.LVD1E, LVD1RI	√	√	—	—	—	—	—
Registers related to the voltage monitor function (Vdet2) RSTSR.LVD2F, LVDCR.LVD2E, LVD2RI	√	√	√	—	—	—	—
Registers related to the low power-consumption function RSTSR.DPSRSTF, DPSBYCR, DPSWCR, DPSIER, DPSIFR, DPSBKRn, DPSIEGR, FWEPROR, SUBOSCCR Register related to the realtime clock RCR2.START Register related to the USB2.0 host/ function module(DPUSR0R, DPUSR1R)	√	√	√	√	—	—	—
Register related to the independent watchdog timer IWDTCR, IWDTSR	√	√	√	√	√	—	—
Register related to the watchdog timer RSTCSR	√	√	√	√	√	√	—
Registers other than the above and internal state	√	√	√	√	√	√	√
Pin state	√	√	√	√	—	√	√

When a reset is canceled, the reset exception handling is started. For the reset exception handling, see section 10, Exceptions.

Table 6.3 shows the pin related to the resets.

Table 6.3 Pin Related to Reset

Pin Name	I/O	Function
RES#	Input	Reset input

6.2 Register Descriptions

Table 6.4 shows the registers related to the resets.

Each register has the bits to indicate the source of each reset generated.

Table 6.4 Registers Related to Resets

Register Name	Symbol	Value after Reset*	Address	Access Size
Reset status register	RSTSR	00h	0008 C285h	8
Reset control/status register	RSTCSR	1Fh	0008 802Bh	8
IWDT status register	IWDTSR	0000h	0008 8034h	16

Note: The applicable reset type depends on the register; see figure 6.1, Block Diagram of Reset Circuit and table 6.2, Targets to be Initialized by Each Reset Type.

6.2.1 Reset Status Register (RSTSR)

For details on the RSTSR register, see section 9, Low Power Consumption.

6.2.2 Reset Control/Status Register (RSTCSR)

For details on the RSTCSR register, see section 24, Watchdog Timer (WDT).

6.2.3 IWDT Status Register (IWDTSR)

For details on the IWDTSR register, see section 25, Independent Watchdog Timer (IWDT).

6.3 Operation

6.3.1 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 RX62N/RX621 enters a reset state.

In order to firmly reset the LSI, the RES# pin should be held low for the specified oscillation stabilization time at a power-on. During operation, the RES# pin should be held low according to the specified reset pulse width.

For details, refer to section 41, Electrical Characteristics.

6.3.2 Power-On Reset

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

If the RES# pin is in the high level state when power is supplied, a power-on reset is generated. After Vcc has exceeded Vpor and the specified period (power-on reset time) has elapsed, the chip is released from the power-on reset state. The power-on reset time is a period for stabilization of the external power supply and the LSI circuit.

If the RES# pin is at the high level when the power supply voltage (Vcc) falls to or below Vpor, a power-on reset is generated. The chip is released from the power-on reset state after Vcc has risen above Vpor and the power-on reset time has elapsed.

After a power-on reset has been generated, the PORF bit in RSTSR is set to 1. The PORF bit is in a read-only register and is only initialized by a pin reset.

Figure 6.2 shows the operation of a power-on reset.

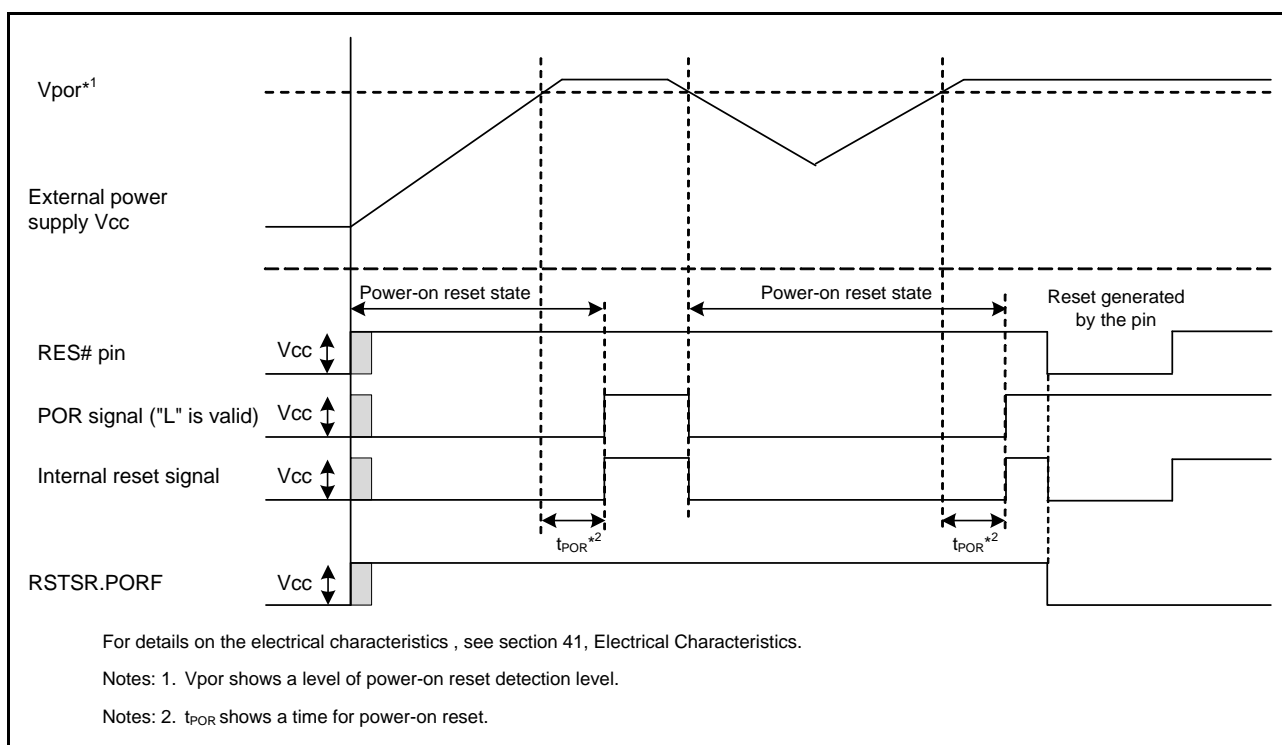


Figure 6.2 Power-On Reset Operation

6.3.3 Voltage-Monitoring Reset

This is an internal reset generated by the voltage detection circuit.

The LVD1F flag is set to 1 if Vcc falls below Vdet1. If the setting of the LVD1E bit in LVDCR is 1 at this time (enabling resets or interrupts from the voltage-detection circuit) and that of the LVD1RI bit in LVDCR is 0 (selecting generation of a reset in response to low-voltage detection), the voltage-detection circuit will generate a voltage-monitoring reset.

In the same way, the LVD2F flag is set to 1 if Vcc falls below Vdet2. If the setting of the LVD2E bit in LVDCR is 1 at this time (enabling resets or interrupts from the voltage-detection circuit) and that of the LVD2RI bit in LVDCR is 0 (selecting generation of a reset in response to low-voltage detection), the voltage-detection circuit will generate a voltage-monitoring reset.

When Vcc subsequently rises above Vdet1 or Vdet2, release from the voltage-monitoring reset proceeds after a specified stabilization time has elapsed.

For details of the voltage-monitoring reset, see section 7, Voltage Detection Circuit (LVD), and section 41, Electrical Characteristics.

6.3.4 Deep Software Standby Reset

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

In other words, a deep software standby reset is generated when a request for release from deep software standby is generated. After that, release from the deep software standby reset state follows once an interval specified by the setting of the DPSWCR.WTSTS[5:0] bits has elapsed. Release from deep software standby accompanies release from the deep software standby reset state.

For details of the deep software standby reset, see section 9, Low Power Consumption.

6.3.5 Independent Watchdog Timer Reset

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

When the independent watchdog timer underflows, an independent watchdog timer reset is generated. A reset is also generated if write operation is performed while refresh operation is not enabled. After a certain time, the independent watchdog timer reset is canceled.

For details of the independent watchdog timer reset, see section 25, Independent Watchdog Timer (IWDT).

6.3.6 Watchdog Timer Reset

This is an internal reset generated by the watchdog timer.

When the RSTE bit in RSTCSR is set to 1, a watchdog timer reset is generated by a watchdog timer overflow. After a certain time, the watchdog timer reset is canceled.

For details of the watchdog timer reset, see section 24, Watchdog Timer (WDT).

6.4 Determination of Reset Generation Source

Reading RSTCSR, IWDTSR, RSTSR, and LVDCR determines which reset was used to execute the reset exception handling.

Figure 6.3 shows an example of the flow to identify a reset generation source.

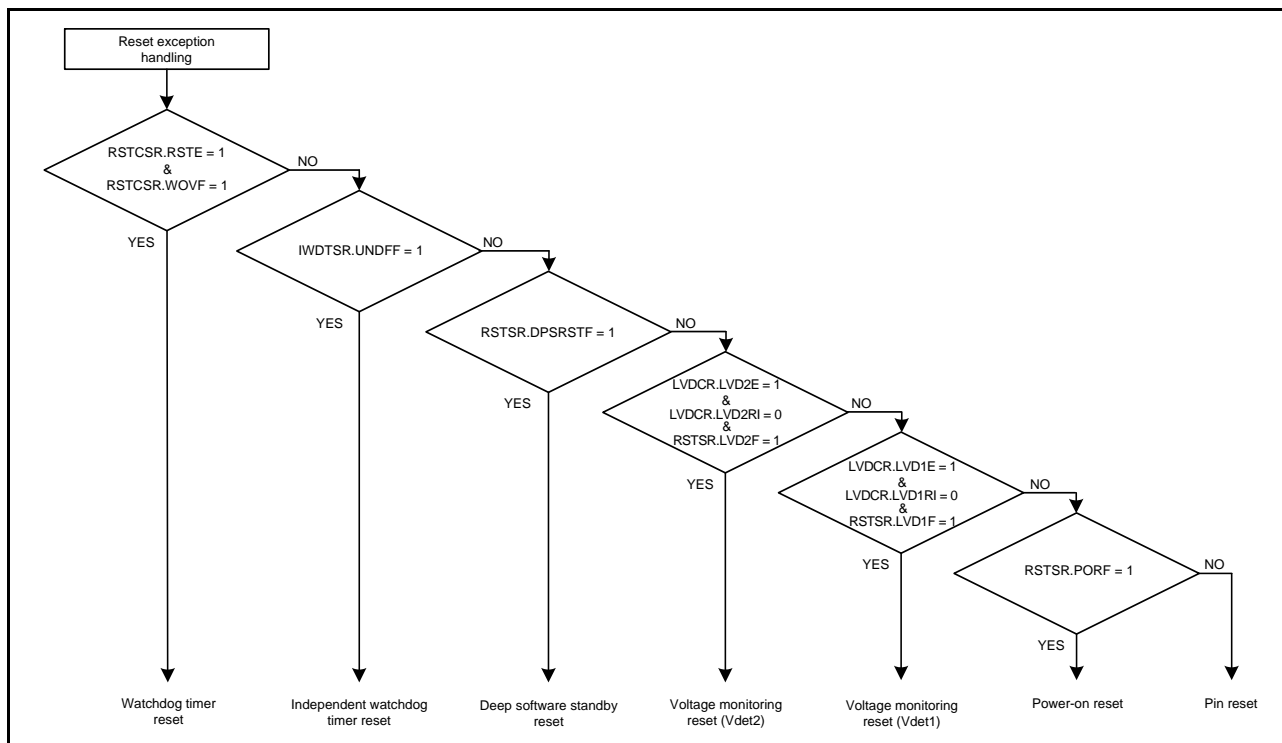


Figure 6.3 Example of Reset Generation Source Determination Flow

6.5 Usage Notes

6.5.1 Notes on Board Design

The XTAL pin and the reset pin are crossly arranged on the RX62N and RX621 Groups. Therefore, to avoid the reference from the clock signal, the reset signal should be guarded by GND.

7. Voltage Detection Circuit (LVD)

7.1 Overview

The voltage detection circuit (LVD) is used to monitor the Vcc voltage level. The LVD is capable of internally resetting the LSI when Vcc falls to the voltage detection level. An interrupt can also be generated.

Table 7.1 Specifications of Voltage Detection Circuit

Item	Specification
Voltage detection circuit 1 (LVD1)	<ul style="list-style-type: none"> Capable of detecting the power-supply voltage (Vcc) becoming less than or equal to Vdet1*. Capable of generating an internal reset or interrupt when a low voltage is detected.
Voltage detection circuit 2 (LVD2)	<ul style="list-style-type: none"> Capable of detecting the power-supply voltage (Vcc) becoming less than or equal to Vdet2*. Capable of generating an internal reset or interrupt when a low voltage is detected.

Note: For Vdet1 and Vdet2, see section 41, Electrical Characteristics.

Figure 7.1 shows a block diagram of the voltage detection circuit.

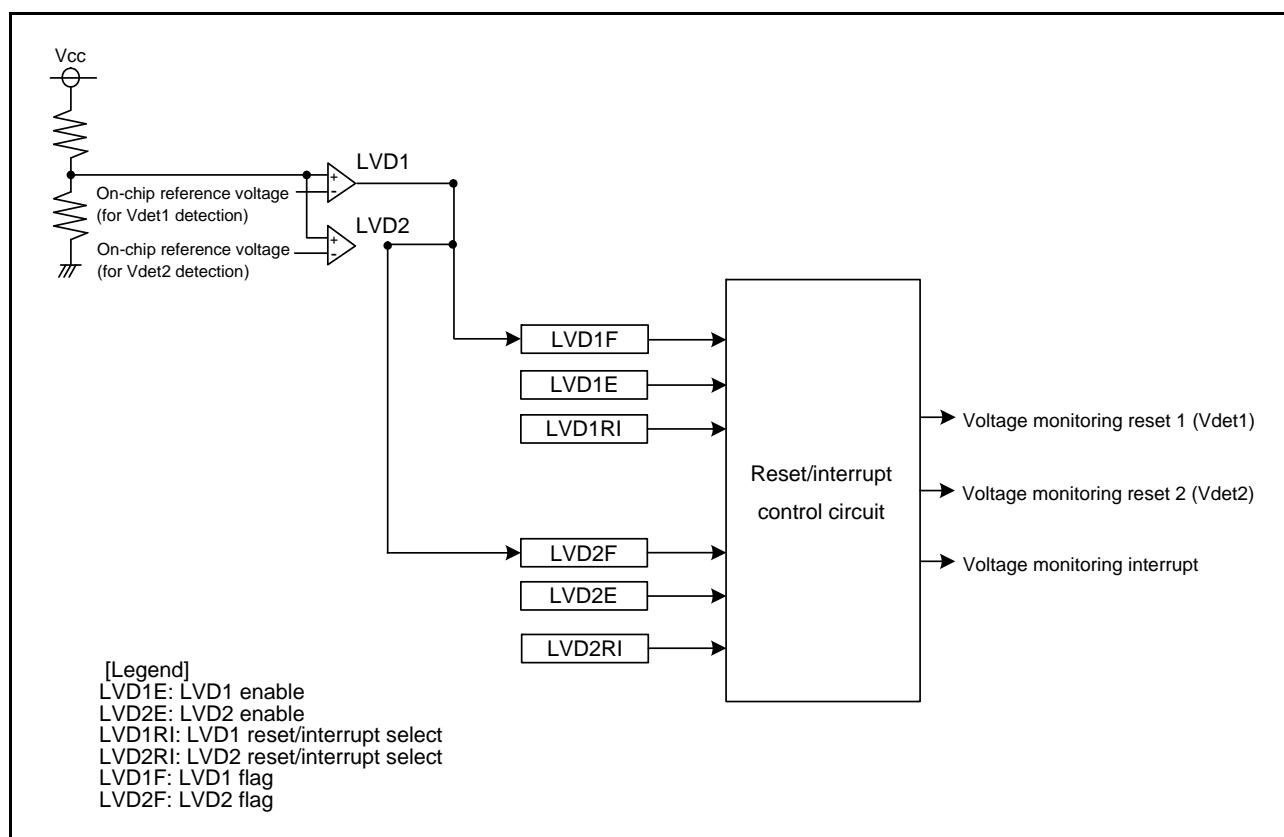


Figure 7.1 Block Diagram of Voltage Detection Circuit

7.2 Register Descriptions

Table 7.2 is the list of voltage detection circuit registers.

Table 7.2 List of Voltage Detection Circuit Registers

Register Name	Symbol	Value after Reset	Address	Access Size
Reset status register	RSTSR	x000 0xxxh	0008 C285h	8
Key code register for low-voltage detection control register	LVDKEYR	0000 0000h	0008 C28Ch	8
Low-voltage detection control register	LVDCCR	000x 000xh	0008 C28Dh	8

7.2.1 Reset Status Register (RSTSR)

For details of RSTSR, see section 9, Low Power Consumption.

7.2.2 Key Code Register for Low-Voltage Detection Control Register (LVDKEYR)

Address: 0008 C28Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	KEY[7:0]	LVDCCR Key Code	3Ch: Modifying LVDCCR is enabled Others: Modifying LVDCCR is disabled	R/W

LVDKEYR enables or disables writing to LVDCCR.

After a specified key code is written to LVDKEYR, LVDCCR can be written to.

KEY[7:0] Bits (LVDCCR Key Code)

Writing 3Ch to LVDKEYR enables writing to LVDCCR. While LVDKEYR holds a value other than 3Ch, writing to LVDCCR is ignored. After LVDCCR is written to, the KEY[7:0] bits are cleared to 00h.

7.2.3 Low-Voltage Detection Control Register (LVDCR)

Address: 0008 C28Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD2E	LVD2RI	—	—	LVD1E	LVD1RI	—	—
Value after reset:	0	0	0	x	0	0	0	x

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is always read as undefined. Writing to this bit has no effect.	R/W
b1	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W
b2	LVD1RI	LVD1 Reset/Interrupt Select	0: A reset is generated when a low voltage is detected 1: An interrupt is generated when a low voltage is detected	R/W
b3	LVD1E	LVD1 Enable	0: LVD1 is disabled 1: LVD1 is enabled	R/W
b4	—	Reserved	This bit is always read as undefined. Writing to this bit has no effect.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W
b6	LVD2RI	LVD2 Reset/Interrupt Select	0: A reset is generated when a low voltage is detected 1: An interrupt is generated when a low voltage is detected	R/W
b7	LVD2E	LVD2 Enable	0: LVD2 is disabled 1: LVD2 is enabled	R/W

LVDCR controls the voltage detection circuit.

The LVD1E and LVD1RI bits are initialized by a pin reset or power-on reset.

The LVD2E and LVD2RI bits are initialized by a pin reset, power-on reset, or voltage monitoring reset 1 (Vdet1).

Table 7.3 shows a relationship between the LVDCR register settings and the states of the voltage detection circuits.

LVD1RI Bit (LVD1 Reset/Interrupt Select)

This bit selects whether an internal reset or interrupt is generated when voltage detection circuit 1 detects a low voltage.

LVD1E Bit (LVD1 Enable)

This bit enables or disables issuing of a reset or an interrupt by voltage detection circuit 1.

LVD2RI Bit (LVD2 Reset/Interrupt Select)

This bit selects whether an internal reset or interrupt is generated when voltage detection circuit 2 detects a low voltage.

LVD2E Bit (LVD2 Enable)

This bit enables or disables issuing of a reset or an interrupt by voltage detection circuit 2.

Table 7.3 LVDCR Register Settings and Voltage Detection Circuit States

LVDCR Register				Voltage Detection Circuit 2 (LVD2)	Voltage Detection Circuit 1 (LVD1)
LVD2E Bit	LVD2RI Bit	LVD1E Bit	LVD1RI Bit		
0	0	0	0	LVD2 disabled	LVD1 disabled
0	0	1	0	LVD2 disabled	LVD1 enabled (reset)
0	0	1	1	LVD2 disabled	LVD1 enabled (interrupt)
1	0	0	0	LVD2 enabled (reset)	LVD1 disabled
1	1	0	0	LVD2 enabled (interrupt)	LVD1 disabled
1	1	1	0	LVD2 enabled (interrupt)	LVD1 enabled (reset)
Other than above: Setting prohibited					

7.3 Voltage Detection Circuit

7.3.1 Voltage Monitoring Reset

Figure 7.2 and Figure 7.3 show the timing of a voltage monitoring reset by the voltage detection circuit.

The LVD2F flag is set to 1 if V_{CC} falls below V_{det2} . If the setting of the LVD2E bit in LVDCR is 1 (enabling resets or interrupts from the voltage-detection circuit) and that of the LVD2RI bit in LVDCR is 0 (selecting generation of a reset in response to low-voltage detection) at this time, the voltage-detection circuit will generate a voltage-monitoring reset.

In the same way, the LVD1F flag in RSTSR is set to 1 if V_{CC} falls below V_{det1} . If the setting of the LVD1E bit in LVDCR is 1 (enabling resets or interrupts from the voltage-detection circuit) and that of the LVD1RI bit in LVDCR is 0 (selecting generation of a reset in response to low-voltage detection) at this time, the voltage-detection circuit will generate a voltage-monitoring reset.

Next, after V_{CC} has risen above V_{det1} or V_{det2} , the voltage monitoring reset is canceled. After the voltage monitoring reset is canceled, the LSI starts reset exception handling.

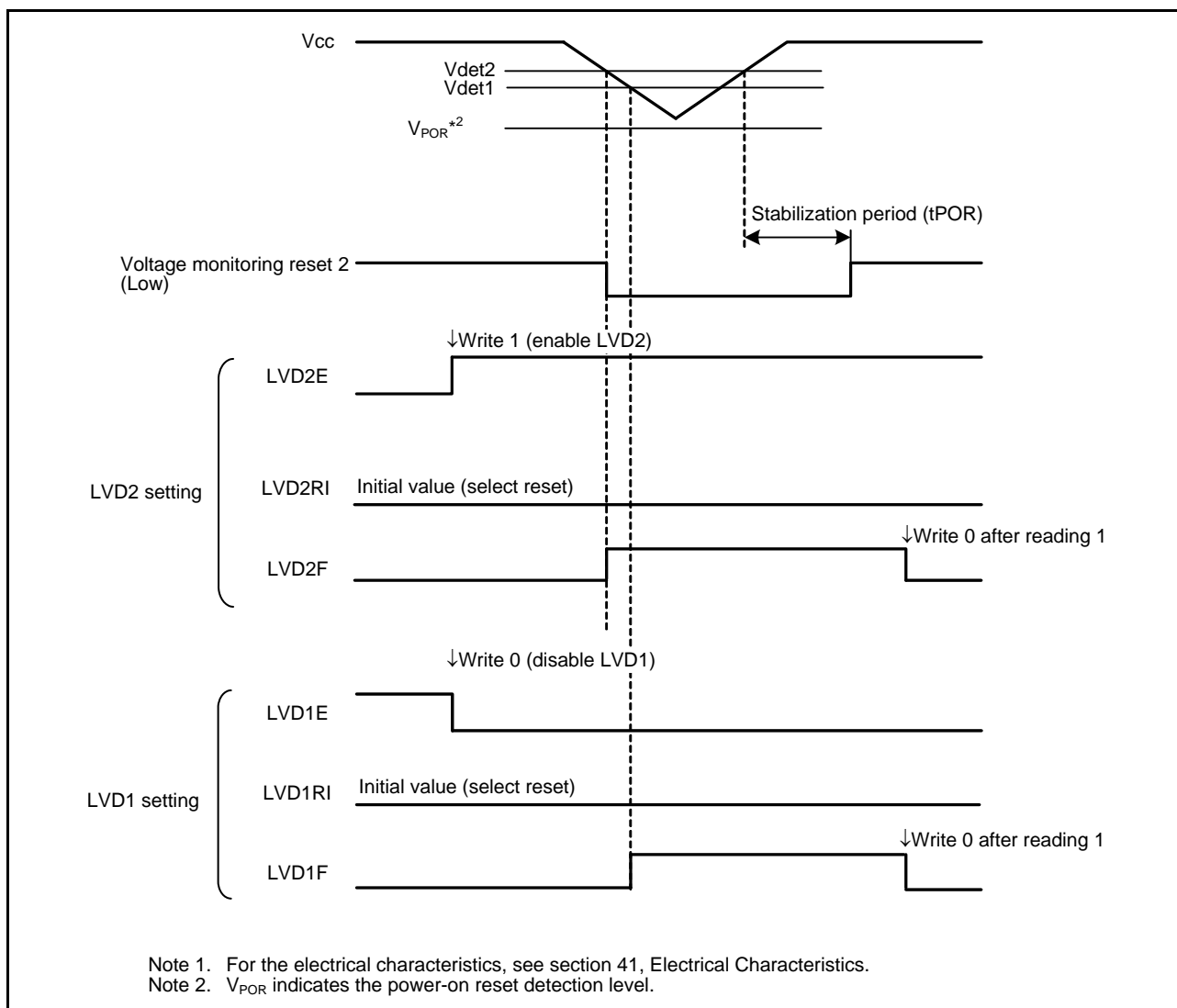


Figure 7.2 Timing Diagram1 of Voltage Monitoring Reset (Reset Selected in LVD2, LVD1 Disabled)

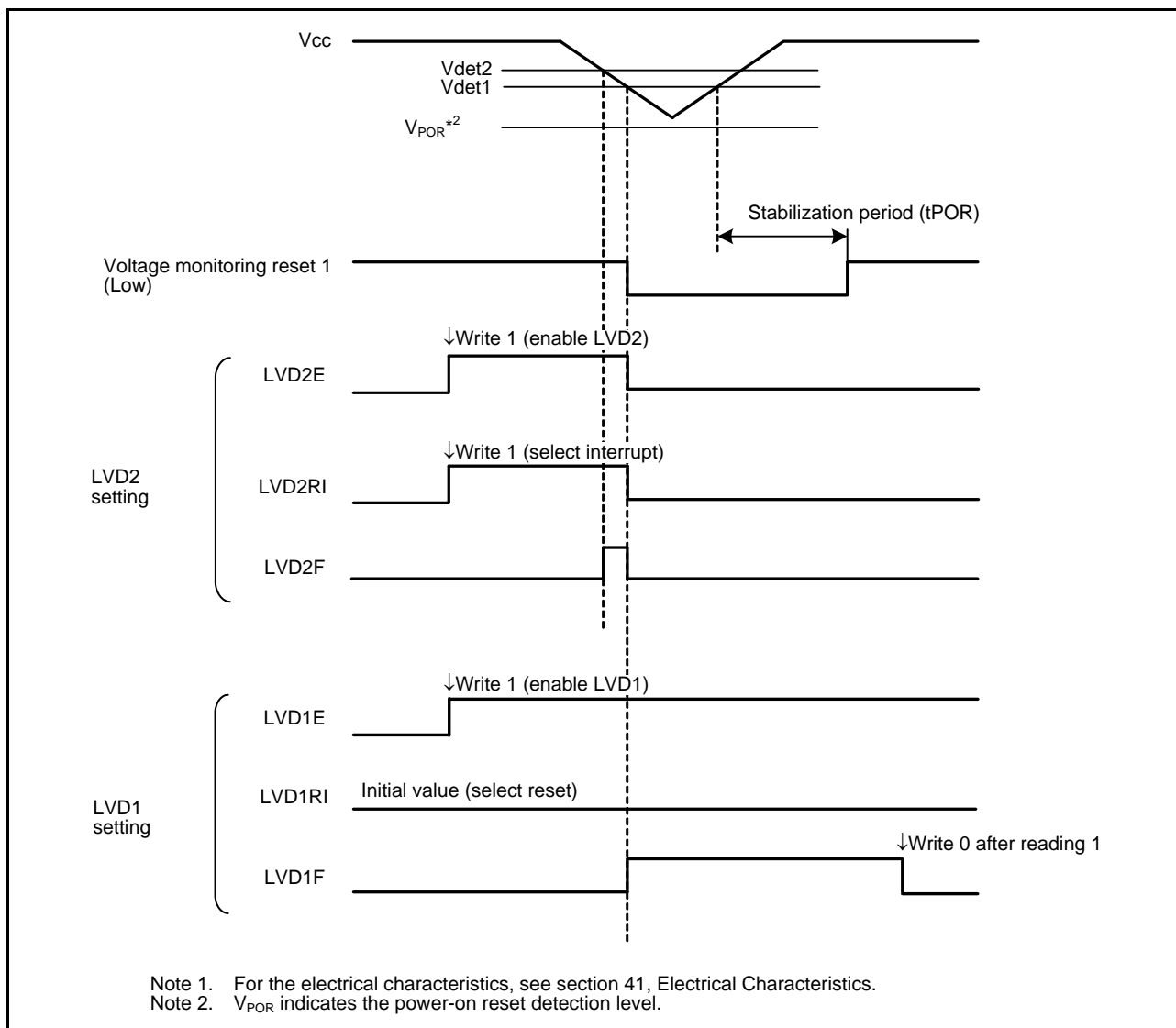


Figure 7.3 Timing Diagram2 of Voltage Monitoring Reset (Reset Selected in LVD2, Reset Selected in LVD1)

7.3.2 Voltage Monitoring Interrupt

Figure 7.4 shows the timing of a voltage monitoring interrupt by the voltage detection circuit.

The LVD1F flag in RSTSR is set to 1 if V_{cc} falls below V_{det1} . If both the LVD1E and LVD1RI bits in LVDCR have the setting 1 at this time, the voltage-detection circuit will generate a voltage-monitoring interrupt.

In the same way, the LVD2F flag is set to 1 if V_{cc} falls below V_{det2} . If both the LVD2E and LVD2RI bits in LVDCR have the setting 1 at this time, the voltage-detection circuit will generate a voltage-monitoring interrupt.

The RSTSR.LVD1F flag can be cleared by writing 0 to the flag after reading it as 1. If the V_{cc} voltage level is equal to or lower than the LVD1 detection level (V_{det1}) at this point, the RSTSR.LVD1F flag cannot be cleared. After 0 has been written to the RSTSR.LVD1F flag, reading 0 from the RSTSR.LVD1F flag will ensure that V_{cc} is higher than V_{det1} .

Likewise, the RSTSR.LVD2F flag can be cleared by writing 0 to the flag after reading it as 1. If the V_{cc} voltage level is equal to or lower than the LVD2 detection level (V_{det2}) at this point, the RSTSR.LVD2F flag cannot be cleared. After 0 has been written to the RSTSR.LVD2F flag, reading 0 from the RSTSR.LVD2F flag will ensure that V_{cc} is higher than V_{det2} .

The voltage monitoring interrupts are classified as non-maskable interrupts. For details of interrupt processing, see section 11, Interrupt Control Unit (ICUa).

Figure 7.5 shows the procedure for setting the voltage monitoring interrupt.

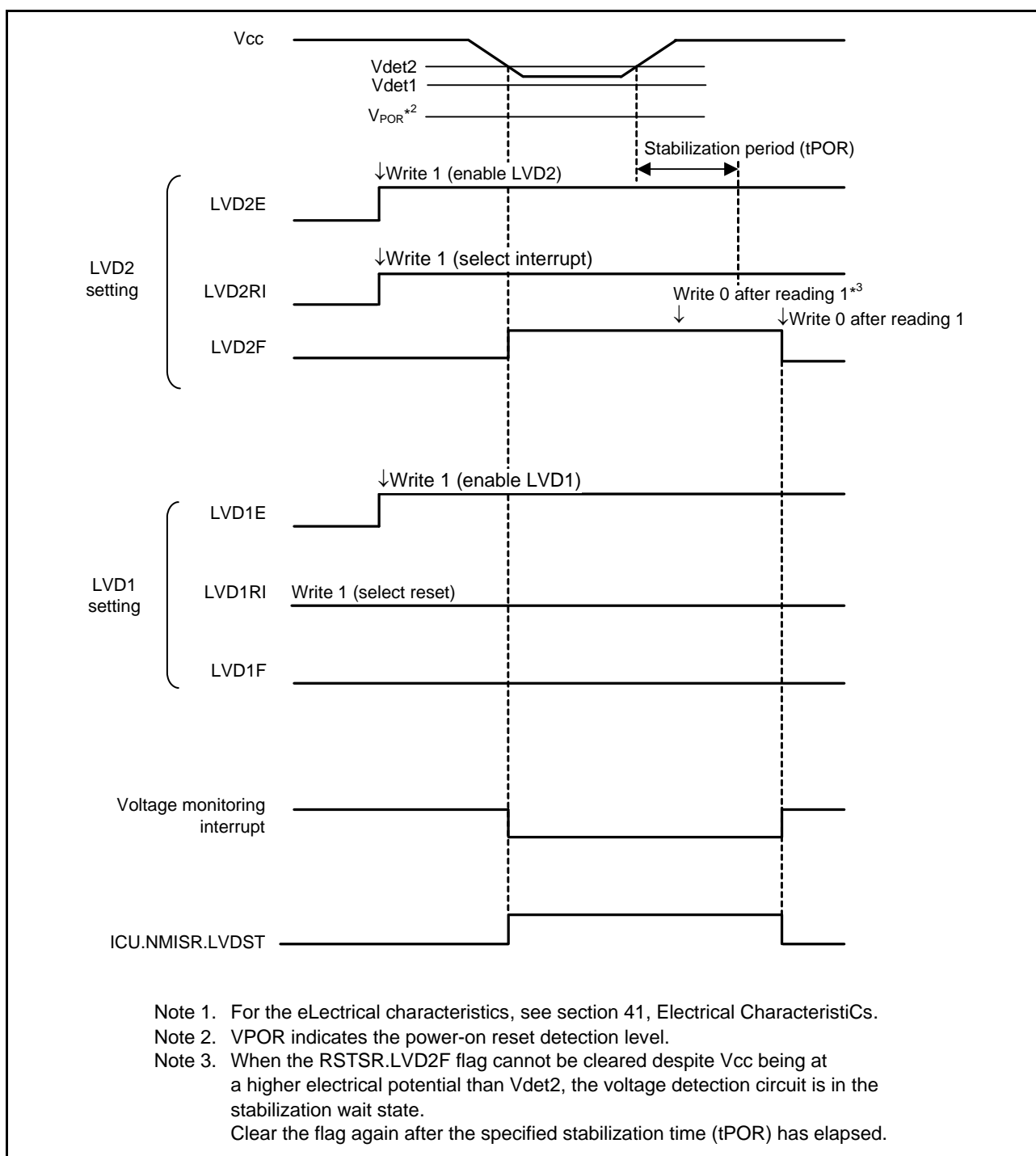


Figure 7.4 Timing of Voltage Monitoring Interrupt (Interrupt Selected in LVD2, Reset Selected in LVD1)

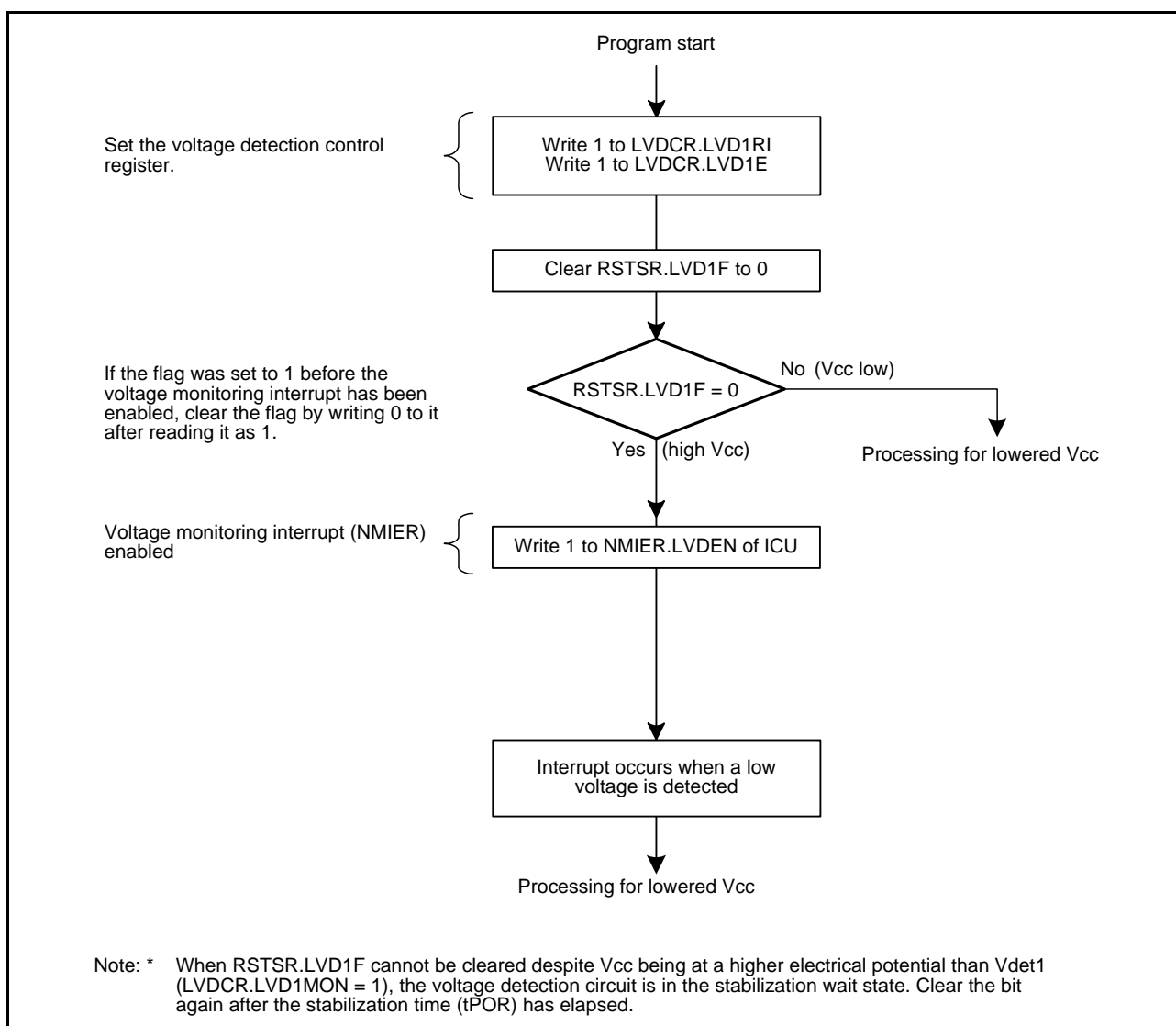


Figure 7.5 Example of Procedure for Setting Voltage Monitoring Interrupt

7.3.3 Cancellation of Deep Software Standby Mode by the Voltage Detection Circuit

The RSTSR.LVD1F flag is set to 1 if V_{cc} falls to or below V_{det1} in deep software standby mode. At this time, if the LVDCR.LVD1E, LVDCR.LVD1RI, and DPSIER.DLVDE bits are all set to 1, the DPSIFR.DLVDF flag is set to 1 and the voltage detection circuit requests release from deep software standby mode.

Likewise, the RSTSR.LVD2F flag is set to 1 if V_{cc} falls to or below V_{det2} in deep software standby mode. At this time, if the LVDCR.LVD2E, LVDCR.LVD2RI, and DPSIER.DLVDE bits are all set to 1, the DPSIFR.DLVDF flag is set to 1 and the voltage detection circuit requests release from deep software standby mode.

For the deep software standby mode, see section 9, Low Power Consumption.

8. Clock Generation Circuit

8.1 Overview

The RX62N/RX621 Group has a clock generation circuit that generates the system clock (ICLK), peripheral module clock (PCLK), external bus clock (BCLK), SDRAM clock (SDCLK), dedicated USB clock (UCLK), dedicated RTC clock (SUBCLK), and on-chip oscillator clock (IWDTCLK) signals.

The clock generation circuit consists of a main clock oscillator, sub-clock oscillator, on-chip oscillator, oscillation stop detection circuit, internal oscillation circuit, phase-locked loop (PLL) circuit, frequency divider, and selector.

Table 8.1 lists the specifications of the clock generation circuit. Figure 8.1 shows a block diagram of the clock generation circuit.

Table 8.1 Specifications of Clock Generation Circuit

Item	Specification
Use	<ul style="list-style-type: none"> Generates the system clock (ICLK) to be supplied to the CPU, DTC, DMACA, ETHERC, EDMAC, ROM, and RAM. Generates the peripheral module clock (PCLK) to be supplied to peripheral modules. Generates the external bus clock (BCLK) to be supplied to the external bus. Generates the SDRAM clock (SDCLK) to be supplied to SDRAM Generates the dedicated USB clock (UCLK) to be supplied to the USB Generates the dedicated RTC clock (SUBCLK) to be supplied to the RTC Generates the on-chip oscillator clock (IWDTCLK) to be supplied to the IWDT
Operating frequency	ICLK: 8 to 100 MHz PCLK: 8 to 50 MHz BCLK: 8 to 100 MHz*2 BCLK pin output: 8 to 50 MHz*3 SDCLK: 8 to 50 MHz SDCLK pin output: 8 to 50 MHz UCLK: 48 MHz (only when EXTAL = 12 MHz) SUBCLK: 32.768 kHz IWDTCLK: 125 kHz (Typ.)*1 Restrictions for setting clock frequencies: ICLK ≥ PCLK and ICLK ≥ BCLK
Connectable resonator or additional circuit	Crystal resonator
Pins for connection to the resonator or additional circuit	Main clock: EXTAL, XTAL Sub-clock: XCIN, XCOUT
Input clock (EXTAL) frequency	8 to 14 MHz
Input clock (XCIN) frequency	32.768 kHz
Selection of ICLK, PCLK, BCLK, SDCLK, or UCLK	<ul style="list-style-type: none"> The ICLK, PCLK, and BCLK are independently selectable as EXTAL × 8, × 4, × 2, and × 1 (the SDCLK and BCLK are set to the same frequency, the UCLK is fixed at EXTAL × 4).
Oscillation stop detection function	<ul style="list-style-type: none"> Switches to internal oscillation upon detection of main clock oscillator stop Sets the MTU pin to the high-impedance state
Control of output on the BCLK pin	<ul style="list-style-type: none"> BCLK clock output or high-level output is selectable BCLK or BCLK/2 is selectable (When EXTAL 1 is selected for BCLK, BCLK/2 cannot be selected.)
Control of output on the SDCLK pin	SDCLK output or a constant high-level output is selectable

Note 1. For details, see section 41, Electrical Characteristics.

Note 2. For products in the 100-pin LQFP and 85-pin TFLGA, the frequency of BCLK is from 8 to 50 MHz.

Note 3. For products in the 100-pin LQFP and 85-pin TFLGA, output on the BCLK pin is at 8 to 25 MHz.

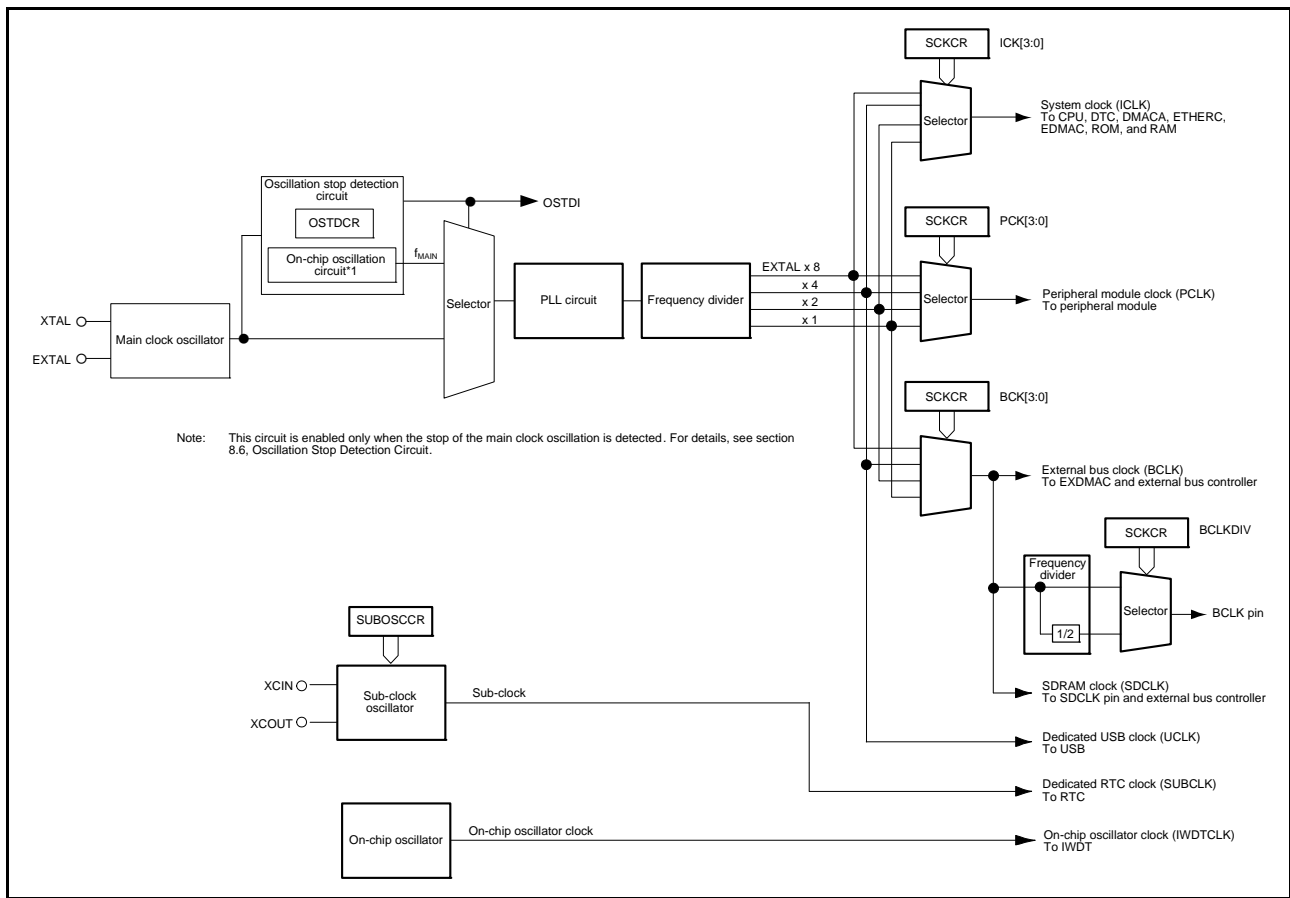


Figure 8.1 Block Diagram of Clock Generation Circuit

Table 8.2 lists the input/output pins of the clock generation circuit.

Table 8.2 Pin Configuration

Pin Name	I/O	Description
XTAL	Output	These pins are used to connect a crystal resonator. The EXTAL pin can also be used to input an external clock. For details, see section 8.3.2, External Clock Input.
EXTAL	Input	
XCIN	Input	These pins are for connection to a 32.768 kHz crystal resonator. An external clock signal is also connectable as the input to the XCIN pin.
XCOUT	Output	
BCLK	Output	This pin is used to supply external devices with the external bus clock (BCLK).
SDCLK	Output	This pin is used to supply external devices with SDRAM clock (SDCLK).

8.2 Register Descriptions

Table 8.3 shows the register of the clock generation circuit.

Table 8.3 Register of Clock Generation Circuit

Register Name	Symbol	Value after Reset	Address	Access Size
System clock control register	SCKCR	0202 0200h	0008 0020h	32
External bus clock control register	BCKCR	00h	0008 0030h	8
Oscillation stop detection control register	OSTDCR	0080h	0008 0040h	16
Sub-clock oscillator control register	SUBOSCCR	00h	0008 C28Ah	8

8.2.1 System Clock Control Register (SCKCR)

Address: 0008 0020h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	ICK[3:0]			
Value after reset:	0	0	0	0	0	0	1	0
	b23	b22	b21	b20	b19	b18	b17	b16
	PSTOP1	PSTOP0	—	—	BCK[3:0]			
Value after reset:	0	0	0	0	0	0	1	0
	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	PCK[3:0]			
Value after reset:	0	0	0	0	0	0	1	0
	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
b7 to b0	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b11 to b8	PCK[3:0]*1	Peripheral Module Clock Select	b11b8 0 0 0 0: x8 0 0 0 1: x4 0 0 1 0: x2 0 0 1 1: x1 Settings other than those listed above are prohibited.	R/W
b15 to b12	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b19 to b16	BCK[3:0]*1	External Bus Clock and SDRAM Clock Select	b19b16 0 0 0 0: x8 0 0 0 1: x4 0 0 1 0: x2 0 0 1 1: x1 Settings other than those listed above are prohibited.	R/W
b21, b20	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b22	PSTOP0	SDCLK Pin Output Control	0: SDCLK pin output is enabled. 1: SDCLK pin output is disabled. (Fixed high)	R/W
b23	PSTOP1	BCLK Pin Output Control	0: BCLK pin output is enabled. 1: BCLK pin output is disabled. (Fixed high)	R/W
b27 to b24	ICK[3:0]*2	System Clock Select	b27b24 0 0 0 0: x8 0 0 0 1: x4 0 0 1 0: x2 0 0 1 1: x1 Settings other than those listed above are prohibited.	R/W
b31 to b28	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note 1. Do not set a frequency higher than the system clock (ICKL). If such a frequency is set, the clock frequency will be the same as the ICKL.

Note 2. Do not set a frequency lower than the peripheral module clock (PCLK) and external bus clock (BCLK). If such a frequency is set, the frequency of the PCLK and BCLK will change to the system clock (ICKL) frequency.

SCKCR is used to control the BCLK and SDCLK outputs and select the frequencies of the system clock (ICLK), peripheral module clock (PCLK), external bus clock (BCLK), and SDRAM clock (SDCLK).

PCK[3:0] Bits (Peripheral Module Clock Select)

These bits select the PCLK frequency.

The value of these bits indicates a multiplication factor of the input clock (EXTAL).

BCK[3:0] Bits (External Bus Clock and SDRAM Clock Select)

These bits select the frequency of the external bus clock (BCLK) and SDRAM clock (SDCLK).

The value of these bits indicates a multiplication factor of the input clock (EXTAL).

PSTOP0 Bit (SDCLK Pin Output Control)

This bit controls stop/supply of SDCLK that is output from the P70 (SDCLK) pin.

When stop is selected, a high-level signal is output.

PSTOP1 Bit (BCLK Pin Output Control)

This bit controls stop/supply of BCLK that is output from the P53 (BCLK) pin.

When stop is selected, a high-level signal is output.

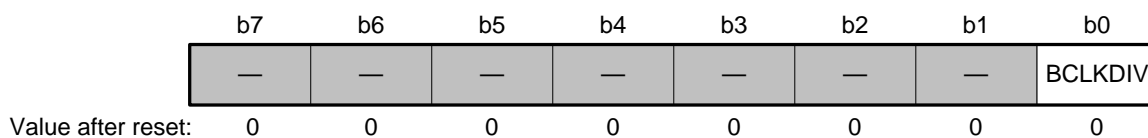
ICK[3:0] Bits (System Clock Select)

These bits select the frequency of the CPU, DMACA, DTC, and system clock (ICLK).

The value of these bits indicates a multiplication factor of the input clock (EXTAL).

8.2.2 External Bus Clock Control Register (BCKCR)

Address: 0008 0030h



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

BCKCR controls the external bus clock.

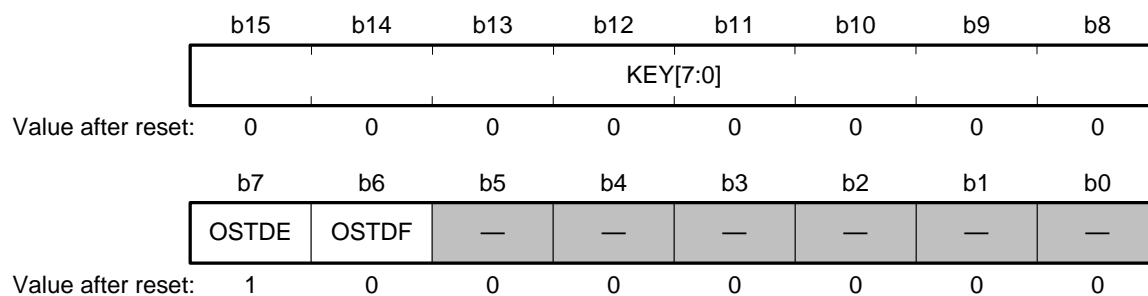
BCLKDIV (BCLK Pin Output Select)

This bit selects the clock signal for output from the P53 pin (BCLK).

Either the BCLK signal specified by the BCLK[3:0] bits in SCKCR or that signal frequency-divided by two is selectable.

8.2.3 Oscillation Stop Detection Control Register (OSTDCR)

Address: 0008 0040h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b6	OSTDF	Oscillation Stop Detection Flag	When OSTDE = 1: 0: The main clock oscillator is operating normally. 1: The stop of the main clock oscillator is detected. When OSTDE = 0: This bit is read as 0.	R
b7	OSTDE	Oscillation Stop Detection Function Enable	0: Oscillation stop detection function is disabled. 1: Oscillation stop detection function is enabled.	R/W
b15 to b8	KEY[7:0]	OSTDCR Key Code	ACh: Writing to OSTDCR is enabled. Other than above: Writing to OSTDCR is disabled. This bit is read as 0.	R/W

OSTDCR is used to control the oscillation stop detection function.

OSTDF Flag (Oscillation Stop Detection Flag)

This flag indicates the state of the main clock oscillator.

When the OSTDF flag is 1, this indicates that the stop of the main clock oscillator is detected.

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 is enabled), transitions to software standby mode or deep software standby mode are not possible. To make a transition to software standby mode or deep software standby mode, issue a WAIT instruction after setting the OSTDE bit to 0.

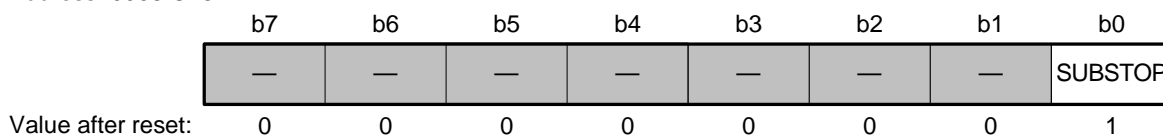
KEY[7:0] Bits (OSTDCR Key Code)

These bits are used to enable or disable writing to OSTDCR.

When writing a value to the OSTDE bit, set the KEY[7:0] bits to ACh before writing. If the KEY[7:0] bits are set to a value other than ACh, the OSTDE bit will not be changed even though OSTDCR is written to.

8.2.4 Sub-Clock Oscillator Control Register (SUBOSCCR)

Address: 0008 C28Ah



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

SUBOSCCR controls the sub-clock oscillator.

SUBSTOP Bit (Sub-Clock Oscillator Control)

This bit controls the operation or stop of the sub-clock oscillator.

8.3 Main Clock Oscillator

Clock pulses can be supplied by connecting a crystal resonator or by inputting an external clock.

8.3.1 Connecting a Crystal Resonator

Figure 8.2 shows an example of connecting a crystal resonator. Table 8.4 shows reference values of the damping resistance (Rd).

When supplying the clock from the crystal resonator, the frequency of the resonator should be in the range of 8 to 14 MHz.

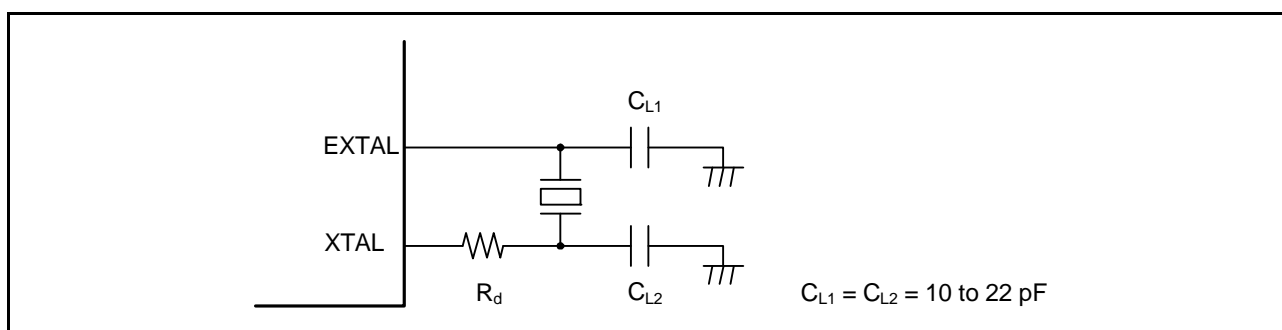


Figure 8.2 Example of Crystal Resonator Connection

Table 8.4 Damping Resistance (Reference Values)

Frequency (MHz)	8	10	12	14
Rd (Ω)	200	100	0	0

Figure 8.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 8.5.

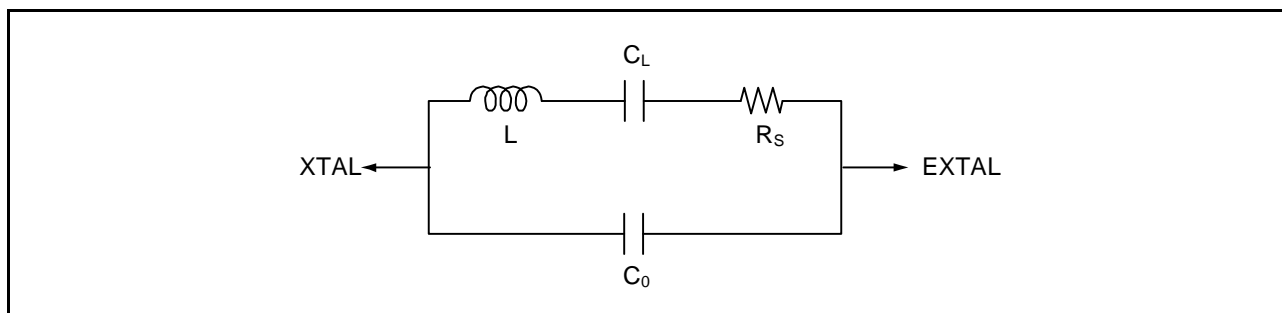


Figure 8.3 Equivalent Circuit of Crystal Resonator

Table 8.5 Crystal Resonator Characteristics (Reference Values)

Frequency (MHz)	8	10	12	14
RS max (Ω)	80	70	60	50
C0 max (pF)	7	7	7	7

8.3.2 External Clock Input

Figure 8.4 shows examples of external clock input. To leave the XTAL pin open, make the parasitic capacitance less than 10 pF. When the counter-phase clock is input to the XTAL pin, hold the external clock in high level during standby mode.

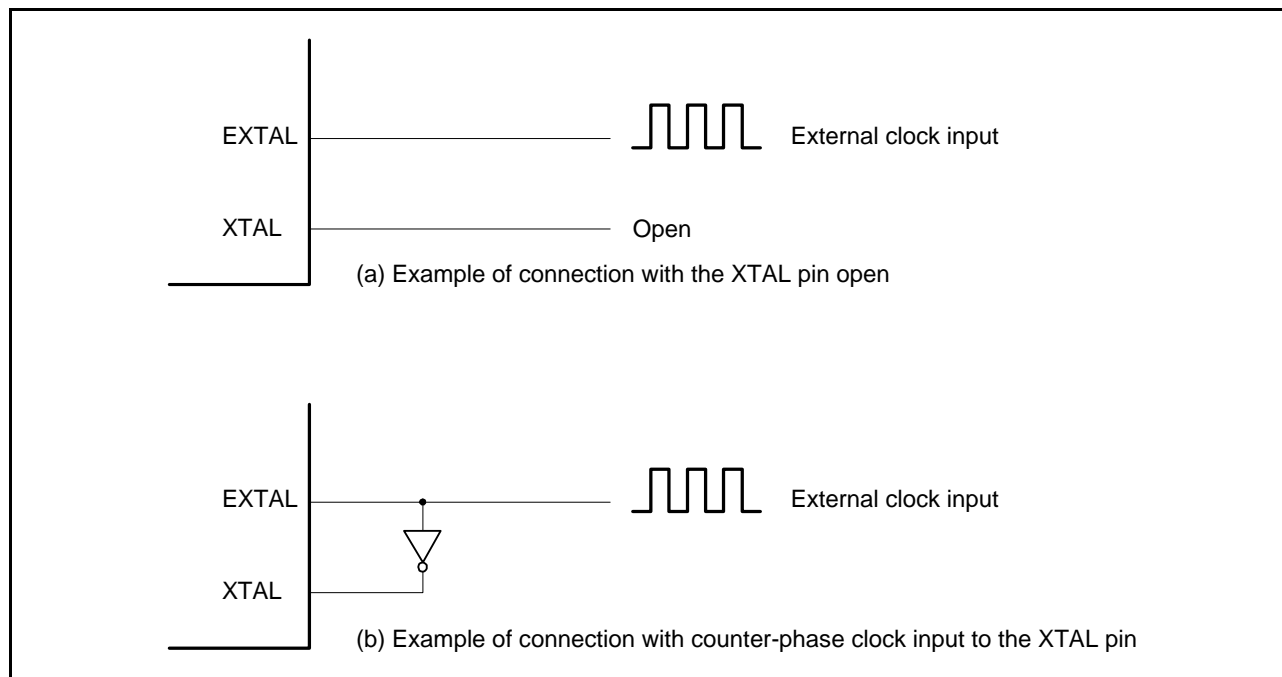


Figure 8.4 Examples of External Clock Input

8.4 Sub-Clock Oscillator

8.4.1 Connecting 32.768-kHz Crystal Resonator

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal resonator, as shown in figure 8.5.

For notes on resonator, see section 8.12.2, Notes on Resonator.

For board design, see section 8.12.3, Notes on Board Design.

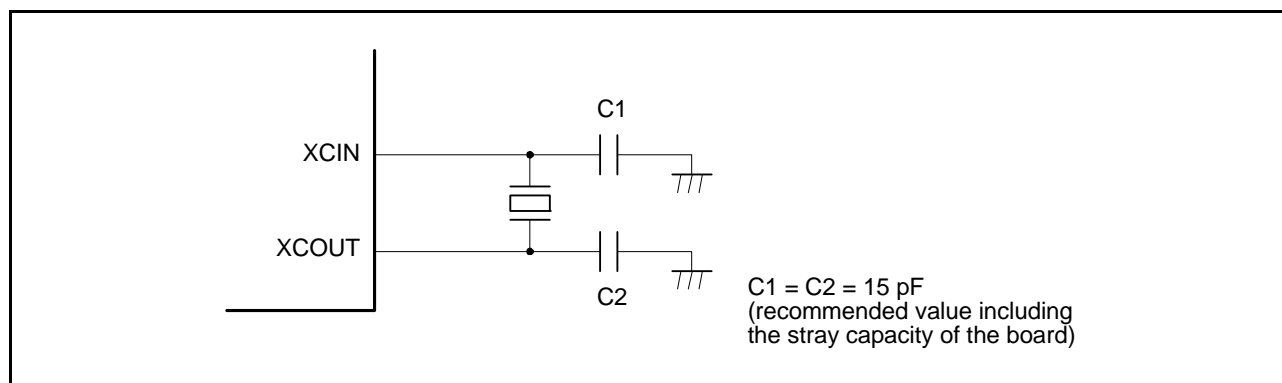


Figure 8.5 Connection Example of 32.768-kHz Crystal Resonator

Figure 8.6 shows an equivalent circuit for the 32.768-kHz crystal resonator.

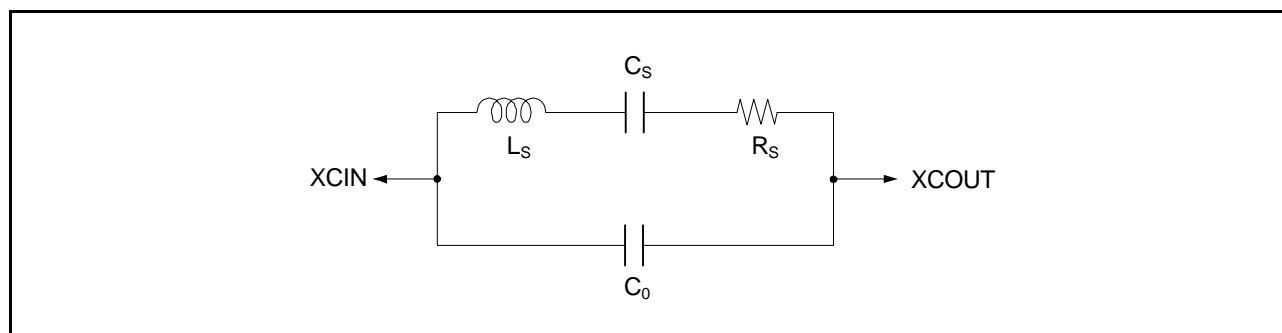


Figure 8.6 Equivalent Circuit for 32.768-kHz Crystal Resonator

8.4.2 Handling of Pins when Sub-Clock is Not Used

If the subclock is not required, connect the XCIN pin to VCC (pulled-up) or VSS (pulled-down) via a resistor and leave the XCOU pin open circuit, as shown in figure 8.7.

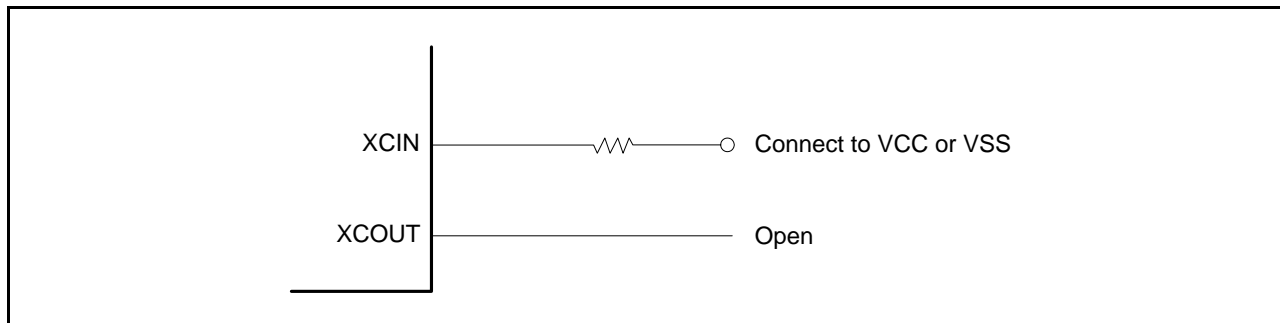


Figure 8.7 Pin Handling when Sub-Clock is not Used

8.5 On-Chip Oscillator

The on-chip oscillator generates the on-chip oscillator clock (IWDTCLK) by the internal oscillation.

8.6 Oscillation Stop Detection Circuit

The oscillation stop detection circuit is able to detect the stop of the main clock oscillator and supply the internal oscillation clock that is output by the internal oscillation circuit, instead of the main clock.

For details, see section 8.11, Oscillation Stop Detection Function.

8.7 Internal Oscillation Circuit

The internal oscillation circuit generates the internal oscillation clock by the internal oscillation.

8.8 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator by a factor of 8.

8.9 Frequency Divider

The frequency divider divides the PLL clock to generate 1/2, 1/4, or 1/8 clock. After the ICK[3:0], PCK[3:0], and BCK[3:0] bits in SCKCR are updated, the RX62N/RX621 operates with the updated frequencies.

The BCKCR.BCLKDIV bit selects either the BCLK signal or that signal frequency-divided by two, for output from the BCLK pin.

8.10 Internal Clock

An internal clock is generated by multiplying the external input clock (EXTAL) by 8 with the PLL circuit, and then dividing the multiplied clock by 1, 2, 4, or 8 with the frequency divider. Other internal clocks are an input clock (XCIN) from an external module and a clock generated with internal oscillation in the on-chip oscillator.

There are the following seven types of internal clock.

- Operating clock of the CPU, DMACA, DTC, ETHERC, EDMAC, ROM, and RAM: System clock (ICLK)
- Operating clock of peripheral modules: Peripheral module clock (PCLK)
- Clock for the external bus controller, EXDMAC, and external pin output: External bus clock (BCLK)
- Clock for the external bus controller and SDRAM external pin output: SDRAM clock (SDCLK)
- Operating clock for the USB: Dedicated USB clock (UCLK)
- Operating clock for the RTC: Dedicated RTC clock (SUBCLK)
- Operating clock for the IWDG: On-chip oscillator clock (IWDGCLK)

The frequencies of these clocks are set by the combinations of bits ICK[3:0], PCK[3:0], BCK[3:0], and BCLKDIV in BCKCR.

8.10.1 System Clock (ICLK)

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

The ICLK frequency is specified by the ICK[3:0] bits in SCKCR.

A frequency lower than the peripheral module clock (PCLK) and external bus clock (BCLK) should not be set for the ICLK. If such a frequency is set, the frequency of the PCLK and BCLK will change to the system clock (ICLK) frequency.

8.10.2 Peripheral Module Clock (PCLK)

The peripheral module clock (PCLK) is the operating clock for peripheral modules.

The PCLK frequency is specified by the PCK[3:0] bits in SCKCR.

A frequency higher than the system clock (ICLK) should not be set for the PCLK. If such a frequency is set, the clock frequency will be the same as the ICLK.

8.10.3 External Bus Clock (BCLK)

The external bus clock (BCLK) is an operating clock for the external bus controller and EXDMAC. It is also output externally from the BCLK pin for the external connection bus.

Setting the B3 bit in DDR for port 5 (PORT5.DDR) to 1 and the PSTOP1 bit in SCKCR to 0 selects output of the BCLK on the BCLK pin. When changing the value of the B3 bit in PORT5.DDR, make sure that the value of the PSTOP1 bit in SCKCR is 1 (high fixed).

Setting the BCKCR.BCLKDIV bit to 1 selects a clock with half the frequency of BCLK to be output from the BCLK pin. Make sure to change the SCKCR.BCK[3:0] bits to "0000b" (EXTAL x 8) while the BCKCR.BCLKDIV bit is 1, and the SCKCR.BCK[3:0] bits to "0011b" (EXTAL x 1) while the BCKCR.BCLKDIV bit is 0. Setting SCKCR.BCK[3:0] bits to 0011b (EXTAL x 1) disables BCKCR.BCLKDIV bit setting, resulting in output of BCLK.

The BCLK frequency is specified by the BCK[3:0] bits in SCKCR.

A frequency higher than the system clock (ICLK) should not be set for the BCLK. If such a frequency is set, the clock frequency will be the same as the ICLK.

8.10.4 SDRAM Clock (SDCLK)

The SDRAM clock (SDCLK) is an operating clock for the external bus controller. It is output externally from the SDCLK pin for the SDRAM, that is connected to the external bus.

When the SCKCR.PSTOP0 bit is cleared to 0 and the PF6BUS.MDSDE or PF6BUS.SDCLKE bit is set to 1, it selects output of SDCLK on the SDCLK pin. When changing the value of the PF6BUS.MDSDE or PF6BUS.SDCLKE bit, make sure that the value of the SCKCR.PSTOP0 bit is 1 (high fixed).

The SDRAM clock frequency is specified by the BCK[3:0] bits in SCKCR.

A frequency higher than the system clock (ICLK) should not be set for the SDCLK. If such a frequency is set, the clock frequency will be the same as the ICLK.

8.10.5 Dedicated USB Clock (UCLK)

The dedicated USB clock (UCLK) is the operating clock for the USB. A clock generated by multiplying the input clock (EXTAL) by four is always output.

48-MHz clock signals need to be supplied to the USB. When using the USB, set the main clock frequency to 12 MHz so that UCLK becomes 48 MHz.

8.10.6 Dedicated RTC Clock (SUBCLK)

The dedicated RTC clock (SUBCLK) is the operating clock for the RTC.

SUBCLK is an input clock (XCIN) from an external module.

8.10.7 On-Chip Oscillator Clock (IWDTCLK)

The on-chip oscillator clock (IWDTCLK) is the operating clock for the IWDT.

IWDTCLK is a clock generated by internal oscillation in the on-chip oscillator.

8.11 Oscillation Stop Detection Function

8.11.1 Detection of Oscillation Stop and Operation after the Detection

The oscillation stop detection function is used to detect termination of the main clock oscillator. After oscillation stop has been detected, the oscillation stop detection circuit is able to supply the internal oscillation clock that is output by the internal oscillation circuit, instead of the main clock. An interrupt request can be generated when oscillation stop has been detected. Also, the MTU output can be forcibly set to the high-impedance state when oscillation stop has been detected. For details on the MTU output, refer to section 18, Multi-Function Timer Pulse Unit 2 (MTU2), section 19, Port Output Enable 2 (POE2), and section 1, Port States in Each Processing Mode.

The RX62N/RX621 Group detects that the main clock has stopped when the input clock is kept at 0 or 1 for a certain period (see Table 41.24) due to an error in the main clock oscillator or another reason.

When oscillation stop has been detected, the RX62N/RX621 Group continues to operate on the internal oscillation clock that is output from the internal oscillation circuit.

For the frequency of the system clock (ICLK) during operation on the internal oscillation clock, see Table 8.6.

Table 8.6 System Clock (ICLK) Frequency when Internal Oscillation Circuit is Used

System Clock (ICLK)	Min.	Typ.	Max.
ICLK frequency (x 8)	$8 \times f_{\text{MAIN}}$	$8 \times f_{\text{MAIN}}$	$8 \times f_{\text{MAIN}}$
ICLK frequency (x 4)	$4 \times f_{\text{MAIN}}$	$4 \times f_{\text{MAIN}}$	$4 \times f_{\text{MAIN}}$
ICLK frequency (x 2)	$2 \times f_{\text{MAIN}}$	$2 \times f_{\text{MAIN}}$	$2 \times f_{\text{MAIN}}$
ICLK frequency (x 1)	f_{MAIN}	f_{MAIN}	f_{MAIN}

Note: For details on f_{MAIN} , see section 41, Electrical Characteristics.

The operating clock is automatically switched from the main clock to the internal oscillator when stoppage of oscillation is detected. Even if the main clock oscillator resumes operation after stoppage of oscillation has been detected, the internal oscillator will provide the operating clock for the LSI. However, the operating clock is switched from the internal oscillator to the main clock only during reset processing of pin resets, power-on resets, and voltage-monitoring. Accordingly, the operating clock for the LSI will be the main clock once oscillation of the main clock oscillator has started.

The oscillation stop detection function is enabled immediately after the LSI has been internally initialized by all resets. To disable this function, clear the OSTDCR.OSTDE bit to 0. In addition, when the oscillation stop is detected and the LSI is operating on the internal oscillation clock, the OSTDCR.OSTDE bit cannot be cleared to 0.

8.11.2 Oscillation Stop Detection Interrupt

When the oscillation stop detection function is enabled, an oscillation stop detection interrupt request (OSCERI) is generated when oscillation stop is detected. The oscillation stop detection interrupt is a non-maskable interrupt. Since non-maskable interrupts are disabled in the initial state after the reset state is canceled, enable non-maskable interrupts through software when using the oscillation stop detection interrupt. For details, refer to **section 11, Interrupt Control Unit (ICUa)**.

The state in which the internal oscillation clock is operating due to oscillation stop detection means that an error has occurred in the system. In this state, take only emergency measures to solve the error.

8.11.3 Note on Oscillation Stop Detection Function

If the main clock oscillator does not operate normally due to error occurrence while the chip is in deep software standby mode, interrupt-driven release from deep software standby mode might be impossible.

However, a pin reset will release the chip from deep software standby mode even if the main clock oscillator is not operating. Use a pin reset if release from deep software standby mode must be ensured.

8.12 Usage Notes

8.12.1 Notes on Clock Generation Circuit

1. The frequencies of the system clock (ICLK), peripheral module clock (PCLK), external bus clock (BCLK), and SDRAM clock (SDCLK) supplied to each module change according to the settings of SCKCR and BCKCR. Select each frequency that is within the operation guaranteed range of clock cycle time (tcyc) specified in AC characteristics of electrical characteristics. Each frequency should meet the following:
When the SDCLK is used, BCLK should be below 50 MHz.
In addition, select the frequency of the BCLK so that the BCLK pin output will be below 50 MHz.
For products in the 100-pin LQFP and 85-pin TFLGA, the BCLK pin output should be below 25 MHz.
ICLK = 8 MHz to 100 MHz, PCLK = 8 MHz to 50 MHz, BCLK = 8 MHz to 100 MHz,
SDCLK = 8 MHz to 50 MHz
When the ETHERC and EDMAC are used, ICLK min. should be 12.5 MHz.
When the USB is used, PCLK min should be 24 MHz.
2. All peripheral modules (except for the DMACA, DTC, EXDMAC, ETHERC, and EDMAC) operate on the PCLK. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.
In addition, the waiting time for canceling software standby mode varies with the PCLK frequency change. For details, see section 9.5.3.3, **Setting Oscillation Settling Time after Software Standby Mode is Canceled**.
3. The relationship among the system clock (ICLK), peripheral module clock (PCLK), and external bus clock (BCLK) is $ICLK \geq PCLK$ and $ICLK \geq BCLK$, and the ICLK has the highest priority. For this reason, if a setting that does not meet these conditions is made, the PCLK and BCLK may have the clock frequency set by the ICK[3:0] bits in SCKCR regardless of the settings of the PCK[3:0] and BCK[3:0] bits in SCKCR.
4. When making necessary settings to change a clock frequency, be careful that the frequency does not change during an access to the external bus.
5. After writing to the SCKCR or BCKCR, further writing to the same register before completion of the change in frequency is ignored. In the case of continued writing to the SCKCR or BCKCR, confirm that values read from the SCKCR or BCKCR are actually the most recently written values.
6. After writing to the SCKCR or BCKCR, transitions to software standby mode are prohibited until completion of the change in frequency. Subsequent operation is not guaranteed if a transition to software standby mode is attempted while the frequency is being changed. The interval between writing to the SCKCR or BCKCR and issuing of the WAIT instruction must take up at least 11 cycles of the system clock. For details, see section 5, I/O Registers.

8.12.2 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.

8.12.3 Notes on Board Design

When using a crystal resonator, place the resonator and its 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 8.8 to prevent electromagnetic induction from interfering with correct oscillation.

The XTAL pin and the reset pin are crossly arranged on the RX62N and RX621 Groups. Therefore, to avoid the reference from the clock signal, the reset signal should be guarded by GND.

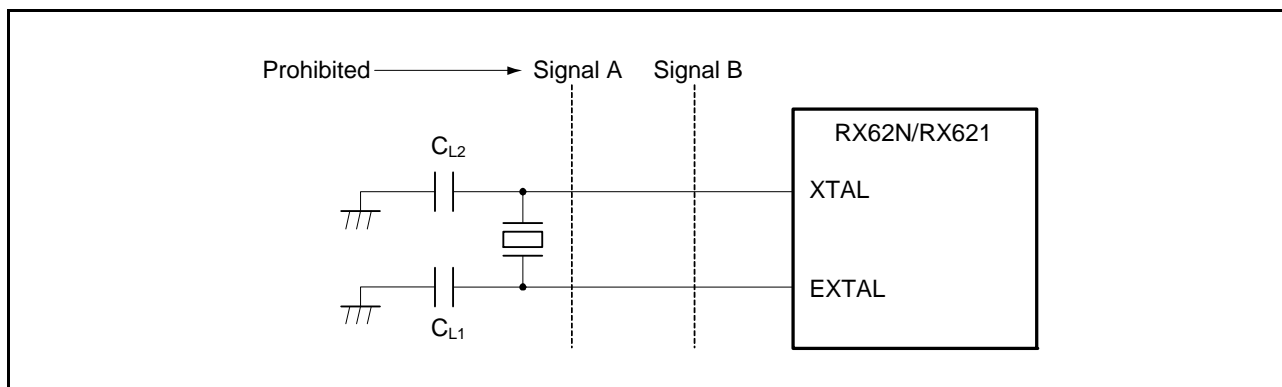


Figure 8.8 Notes on Board Design for Oscillation Circuit

Figure 8.9 shows a recommended external circuit for the PLL circuit. Separate pins PLLVcc and PLLVss from the other Vcc and Vss lines at the board power supply source, and be sure to insert bypass capacitors CPB and CB close to the pins. The pins PLLVcc and Vcc should be set to the same potential.

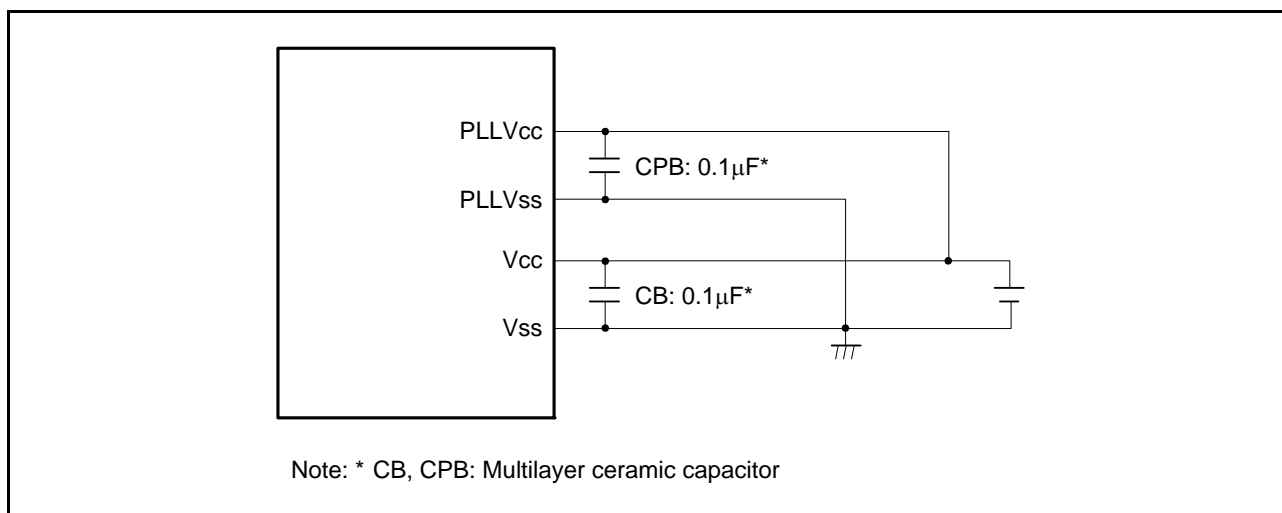


Figure 8.9 Recommended External Circuit for PLL Circuit

9. Low Power Consumption

9.1 Overview

The RX62N/RX621 Group has functions to reduce power consumption, including a multi-clock function, BCLK output control function, SDCLK output control function, module stop function, and a function for transition to low power consumption mode.

Table 9.1 shows the specifications of low power consumption function, and Table 9.2 shows the conditions to shift to low power consumption mode, states of the CPU and peripheral modules, and mode canceling method. After the reset state, this LSI enters the normal program execution state, but modules except the DTC, DMACA, and EXDMAC do not operate.

Table 9.1 Specifications of Low Power Consumption Function

Item	Specification
Multi-clock function	The frequency division ratio is settable independently for the system clock (ICLK), peripheral module clock (PCLK), and external bus clock (BCLK).
BCLK output control function	BCLK output or high-level output can be selected.
SDCLK output control function	SDCLK output or high-level output can be selected.
Module stop function	Functions can be stopped independently for each peripheral module.
Function for transition to low power consumption mode	Transition to low power consumption mode is enabled to stop the CPU, peripheral modules, and oscillator.
Four low power consumption modes	Sleep mode All-module clock stop mode Software standby mode Deep software standby mode

Table 9.2 Transition and Cancellation of the Mode and the State of Operation

Transition and Cancellation of the Mode and the State of Operation	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode	Deep Software Standby Mode
Transition method	Control register + instruction	Control register + instruction	Control register + instruction	Control register + instruction
Canceling method other than resets	Interrupt	Interrupt*1	Interrupt*2	Interrupt*3
State after cancellation*4	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (reset processing)
Oscillator	Operating	Operating	Stopped	Stopped
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
On-chip RAM 1 (0001 0000h to 0001 7FFFh)	Operating (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
On-chip RAM 0 (0000 0000h to 0000 FFFFh)	Operating (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained/Undefined)*5
USB2.0 host/function module (USB)	Operating	Stopped*6	Stopped*6	Stopped (Retained/Undefined)*7
Watchdog timer (WDT)	Operating	Operating	Stopped (Retained)	Stopped (Undefined)
Independent watchdog timer (IWDT)	Operating	Operating	Stopped (Retained)	Stopped (Undefined)
8-bit timer (unit 0, unit 1)	Operating	Operating*8	Stopped (Retained)	Stopped (Undefined)
Realtime clock (RTC)	Operating	Operating	Operating	Operating
Voltage detection circuit	Operating	Operating	Operating	Operating
Power-on reset circuit	Operating	Operating	Operating	Operating
Peripheral modules	Operating	Stopped*9	Stopped*9	Stopped (Undefined)
I/O pin state	Operating	Retained*11*12	Retained*10*12	Retained*10

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

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

Note 1. An external interrupt or some internal interrupts (8-bit timer, WDT, RTC alarm, oscillation stop detection, USB interrupt (USBR), and voltage monitoring).

Note 2. An external interrupt or some internal interrupts (voltage monitoring, RTC alarm, and USB interrupt (USBR)).

Note 3. NNMI, only side A of IRQ0 to IRQ3, or some internal interrupts (voltage monitoring, RTC alarm, and USB suspend/resume). These interrupts are enabled only when the corresponding bit in the deep standby interrupt enable register (DPSIER) is set to 1.

Note 4. Cancellation by the RES# pin, power-on reset, voltage monitoring reset, watchdog timer reset, or independent watchdog timer reset is excluded. When canceled by the RES# pin, power-on reset, voltage monitoring reset, watchdog timer reset, or independent watchdog timer reset, this LSI enters the reset state.

Note 5. "Retained" or "Undefined" can be selected by setting the on-chip RAM Off 2, on-chip RAM Off 1, and on-chip RAM Off 0 bits (RAMCUT2/RAMCUT1/RAMCUT0) in DPSBYCR.

Note 6. Resume detecting operation is valid.

Note 7. The USB resume detection function is enabled or disabled by setting the on-chip RAM Off 2, on-chip RAM Off 1, and on-chip RAM Off 0 bits (RAMCUT2/RAMCUT1/RAMCUT0) in DPSBYCR.

Note 8. "Operating" or "Stopped" can be selected by setting the 8-bit timer 3/2 (unit 1) module stop and 8-bit timer 1/0 (unit 0) module stop bits (MSTPA5/MSTPA4) in MSTPCRA.

Note 9. Peripheral modules retain the state.

Note 10. "Retained" or "High impedance" for the address bus and bus control signals (CS0# to CS7#, RD#, WR#, WR0# to WR3#, BC0# to BC3#, CKE, SDCS#, RAS#, CAS#, WE#, and DQM0 to DQM3) can be selected by the setting of the output port enable bit (OPE) in SBYCR.

Note 11. When pin P53 is being used as the output pin for the BCLK signal, operation as the BCLK output is maintained. When pin P70 is being used as the output pin for the SDCLK signal, operation as the SDCLK output is maintained. For details, see section 9.6, BCLK and SDCLK Output Control.

When an I/O pin for 8-bit timer output (TMO) is being used as the output pin for the TMO, operation as the TMO output is maintained. When the watchdog timer is used, operation of the WDTOVF# output is maintained.

Note 12. When pin P32 is being used as the output pin for the RTCOUT signal, operation as the RTCOUT output is maintained.

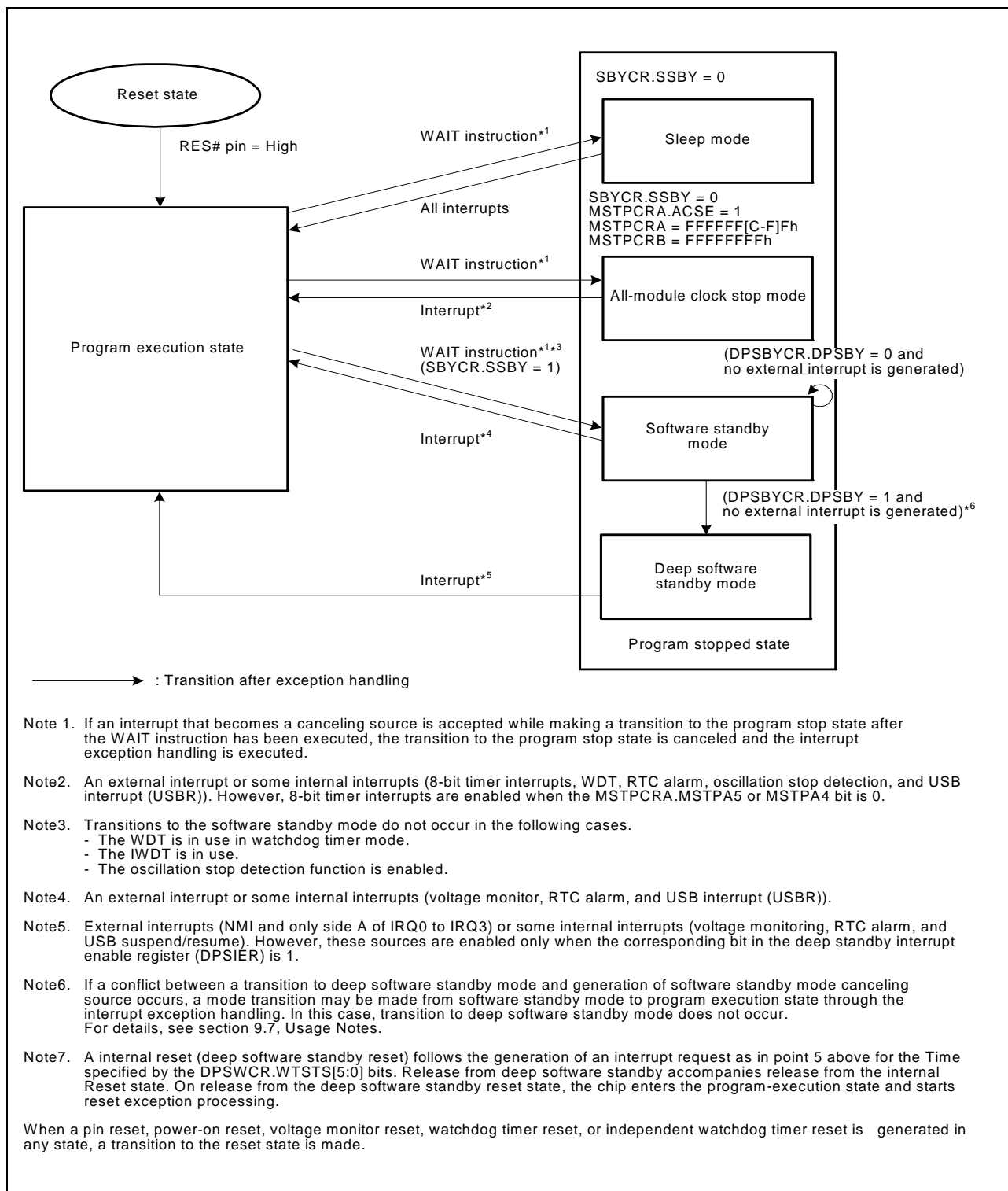


Figure 9.1 Mode Transitions

9.2 Register Descriptions

Table 9.3 is the list of low power consumption registers. For details on the system clock control register (SCKCR), see section 8.2.1, System Clock Control Register (SCKCR).

Table 9.3 List of Low Power Consumption Registers (1 / 2)

Register Name	Symbol	Value after Reset	Address	Access Size
Standby control register	SBYCR	4F00h	0008 000Ch	16
Module stop control register A	MSTPCRA	4xFF FFFFh	0008 0010h	32
Module stop control register B	MSTPCRB	FFFF FFFFh	0008 0014h	32
Module stop control register C	MSTPCRC	FFFF 0000h	0008 0018h	32
Deep standby control register	DPSBYCR	31h	0008 C280h	8
Deep standby wait control register	DPSWCR	0Fh	0008 C281h	8
Deep standby interrupt enable register	DPSIER	00h	0008 C282h	8
Deep standby interrupt flag register	DPSIFR	00h	0008 C283h	8
Deep standby interrupt edge register	DPSIEGR	00h	0008 C284h	8
Reset status register	RSTSR	x000 0xxxb	0008 C285h	8
Deep standby backup register 0	DPSBKR0	xxh*	0008 C290h	8
Deep standby backup register 1	DPSBKR1	xxh*	0008 C291h	8
Deep standby backup register 2	DPSBKR2	xxh*	0008 C292h	8
Deep standby backup register 3	DPSBKR3	xxh*	0008 C293h	8
Deep standby backup register 4	DPSBKR4	xxh*	0008 C294h	8
Deep standby backup register 5	DPSBKR5	xxh*	0008 C295h	8
Deep standby backup register 6	DPSBKR6	xxh*	0008 C296h	8
Deep standby backup register 7	DPSBKR7	xxh*	0008 C297h	8
Deep standby backup register 8	DPSBKR8	xxh*	0008 C298h	8
Deep standby backup register 9	DPSBKR9	xxh*	0008 C299h	8
Deep standby backup register 10	DPSBKR10	xxh*	0008 C29Ah	8
Deep standby backup register 11	DPSBKR11	xxh*	0008 C29Bh	8
Deep standby backup register 12	DPSBKR12	xxh*	0008 C29Ch	8
Deep standby backup register 13	DPSBKR13	xxh*	0008 C29Dh	8
Deep standby backup register 14	DPSBKR14	xxh*	0008 C29Eh	8
Deep standby backup register 15	DPSBKR15	xxh*	0008 C29Fh	8
Deep standby backup register 16	DPSBKR16	xxh*	0008 C2A0h	8
Deep standby backup register 17	DPSBKR17	xxh*	0008 C2A1h	8
Deep standby backup register 18	DPSBKR18	xxh*	0008 C2A2h	8
Deep standby backup register 19	DPSBKR19	xxh*	0008 C2A3h	8
Deep standby backup register 20	DPSBKR20	xxh*	0008 C2A4h	8
Deep standby backup register 21	DPSBKR21	xxh*	0008 C2A5h	8
Deep standby backup register 22	DPSBKR22	xxh*	0008 C2A6h	8
Deep standby backup register 23	DPSBKR23	xxh*	0008 C2A7h	8
Deep standby backup register 24	DPSBKR24	xxh*	0008 C2A8h	8
Deep standby backup register 25	DPSBKR25	xxh*	0008 C2A9h	8
Deep standby backup register 26	DPSBKR26	xxh*	0008 C2AAh	8
Deep standby backup register 27	DPSBKR27	xxh*	0008 C2ABh	8
Deep standby backup register 28	DPSBKR28	xxh*	0008 C2ACh	8
Deep standby backup register 29	DPSBKR29	xxh*	0008 C2ADh	8

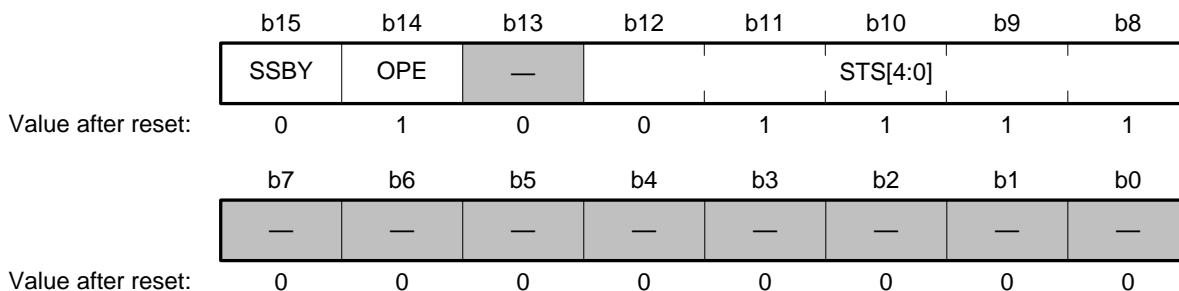
Table 9.3 List of Low Power Consumption Registers (2 / 2)

Register Name	Symbol	Value after Reset	Address	Access Size
Deep standby backup register 30	DPSBKR30	xxh*	0008 C2AEh	8
Deep standby backup register 31	DPSBKR31	xxh*	0008 C2AFh	8

Note: * DPSBKR0 to DPSBKR31 are not initialized and their values are undefined immediately after power-on.

9.2.1 Standby Control Register (SBYCR)

Address: 0008 000Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b12 to b8	STS[4:0]	Standby Timer Select	b12 b8 0 0 1 0 1: Waiting time = 64 states 0 0 1 1 0: Waiting time = 512 states 0 0 1 1 1: Waiting time = 1024 states 0 1 0 0 0: Waiting time = 2048 states 0 1 0 0 1: Waiting time = 4096 states 0 1 0 1 0: Waiting time = 16384 states 0 1 0 1 1: Waiting time = 32768 states 0 1 1 0 0: Waiting time = 65536 states 0 1 1 0 1: Waiting time = 131072 states 0 1 1 1 0: Waiting time = 262144 states 0 1 1 1 1: Waiting time = 524288 states Settings other than above are prohibited.	R/W
b13	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b14	OPE	Output Port Enable	0: In software standby mode or deep software standby mode, the address bus and bus control signals are set to the high-impedance state. 1: In software standby mode or deep software standby mode, the address bus and bus control signals retain the output state.	R/W
b15	SSBY	Software Standby	0: Shifts to sleep mode or all-module clock stop mode after the WAIT instruction is executed 1: Shifts to software standby mode after the WAIT instruction is executed	R/W

SBYCR is used to control software standby mode.

STS[4:0] Bits (Standby Timer Select)

These bits select the time for the RX62N/RX621 Group to wait until the clock is stabilized when software standby mode is canceled by an external interrupt.

In the case of crystal oscillation, see [Table 9.4](#) and make a selection according to the operating frequency so that the waiting time is no less than the oscillation settling time. When an external clock is used, the PLL circuit settling time is necessary. Select a waiting time referring to [Table 9.4](#).

During the oscillation settling time, the standby timer is counted on the peripheral module clock (PCLK) frequency. Note this in multi-clock mode.

OPE Bit (Output Port Enable)

The OPE bit specifies whether to retain the output of the address bus and bus control signals (CS0# to CS7#, RD#, WR#, WR0# to WR3#, BC0# to BC3#, CKE, SDCS#, RAS#, CAS#, WE#, and DQM0 to DQM3) in software standby mode or deep software standby mode, or to set the output to the high-impedance state.

SSBY Bit (Software Standby)

The SSBY bit specifies the transition destination after the WAIT instruction is executed.

When the SSBY bit is set to 0, the LSI enters either sleep mode or all-module clock stop mode after execution of the WAIT instruction, according to the setting of the MSTPCRA and MSTPCRB registers. When the SSBY bit is set to 1, the LSI enters software standby mode after execution of the WAIT instruction. In this case, when the DPSBY bit in DPSBYCR is 1, the LSI enters deep software standby mode after software standby mode. For details, see [section 9.5, Low Power Consumption Modes](#).

This bit is not cleared to 0 when software standby mode is canceled by an external interrupt and the LSI enters normal mode. Write 0 to this bit to clear.

When the WDT is used in watchdog timer mode, the IWDT is used, or the oscillation stop detection function is enabled, the setting of this bit is invalid and the LSI always enters sleep mode or all-module clock stop mode after the WAIT instruction is executed.

9.2.2 Module Stop Control Register A (MSTPCRA)

Address: 0008 0010h

	b31	b30	b29	b28	b27	b26	b25	b24
	ACSE	—	MSTPA29	MSTPA28	—	—	—	—
Value after reset:	0	1	0	0	x	1	1	1
	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPA23	MSTPA22	—	—	MSTPA19	—	MSTPA17	—
Value after reset:	1	1	1	1	1	1	1	1
	b15	b14	b13	b12	b11	b10	b9	b8
	MSTPA15	MSTPA14	—	—	MSTPA11	MSTPA10	MSTPA9	MSTPA8
Value after reset:	1	1	1	1	1	1	1	1
	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MSTPA5	MSTPA4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 1. The write value should always be 1.	R/W
b4	MSTPA4	8-Bit Timer 3/2 (Unit 1) Module Stop	Target module: TMR3/TMR2 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b5	MSTPA5	8-Bit Timer 1/0 (Unit 0) Module Stop	Target module: TMR1/TMR0 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b7, b6	—	Reserved	These bits are always read as 1. The write value should always be 1.	R/W
b8	MSTPA8	Multifunction Timer Pulse Unit (Unit 1) Module Stop	Target module: MTU unit 1 (MTU6 to MTU11) 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b9	MSTPA9	Multifunction Timer Pulse Unit (Unit 0) Module Stop	Target module: MTU unit 0 (MTU0 to MTU5) 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b10	MSTPA10	Programmable Pulse Generator (Unit 1) Module Stop	Target module: PPG1 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b11	MSTPA11	Programmable Pulse Generator (Unit 0) Module Stop	Target module: PPG0 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b13, b12	—	Reserved	These bits are always read as 1. The write value should always be 1.	R/W
b14	MSTPA14	Compare Match Timer (Unit 1) Module Stop	Target module: CMT unit 1 (CMT2, CMT3) 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b15	MSTPA15	Compare Match Timer (Unit 0) Module Stop	Target module: CMT unit 0 (CMT0, CMT1) 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b16	—	Reserved	This bit is always read as 1. The write value should always be 1.	R/W

Bit	Symbol	Bit Name	Description	R/W
b17	MSTPA17* ¹	12-bit A/D Converter Module Stop	Target module: S12AD 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b18	—	Reserved	This bit is always read as 1. The write value should always be 1.	R/W
b19	MSTPA19	D/A Converter Module Stop	Target module: DA 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b21, b20	—	Reserved	These bits are always read as 1. The write value should always be 1.	R/W
b22	MSTPA22* ¹	10-bit A/D Converter (Unit 1) Module Stop	Target module: AD1 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b23	MSTPA23* ¹	10-bit A/D Converter (Unit 0) Module Stop	Target module: AD0 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b26 to b24	—	Reserved	These bits are always read as 1. The write value should always be 1.	R/W
b27	—	Reserved	This bit is always read as undefined. The write value should always be 1.	R/W
b28	MSTPA28	DMA Controller/Data Transfer Controller Module Stop	Target module: DMACA and DTC 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b29	MSTPA29	EXDMA Controller Module Stop	Target module: EXDMAC 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b30	—	Reserved	This bit is always read as 1. The write value should always be 1.	R/W
b31	ACSE* ²	All-Module Clock Stop Mode Enable	0: All-module clock stop mode is disabled 1: All-module clock stop mode is enabled	R/W

Note 1. Simultaneously using a 10-bit A/D converter and the 12-bit A/D converter is not possible. Do not make the settings of the MSTPA17 and MSTPA22 bits or the MSTPA17 and MSTPA23 bits that release converters of both widths from the module-stop state at the same time.

Note 2. When the SBYCR.SSBY bit is 0 and the MSTPCRA.ACSE bit is 0, shift to sleep mode after the WAIT instruction is executed.

MSTPCRA is used to control the module stop state.

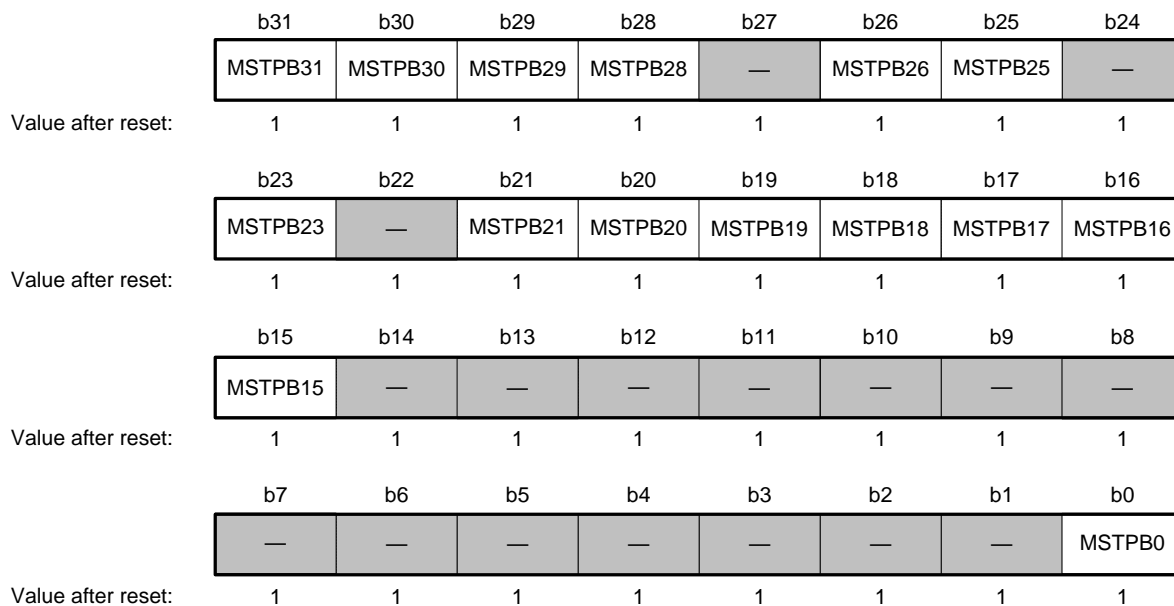
ACSE Bit (All-Module Clock Stop Mode Enable)

The ACSE bit enables or disables all-module clock stop mode for reducing supply current by stopping the bus controller and I/O ports when the CPU executes the WAIT instruction after the module stop state has been specified for all modules* controlled by MSTPCRA and MSTPCRB.

Note: * Whether to stop the 8-bit timers or not can be selected by the MSTPA5 and MSTPA 4 bits.

9.2.3 Module Stop Control Register B (MSTPCRB)

Address: 0008 0014h



Bit	Symbol	Bit Name	Description	R/W
b0	MSTPB0	CAN Module Stop	Target module: CAN 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b14 to b0	—	Reserved	These bits are always read as 1. The write value should always be 1.	R/W
b15	MSTPB15	Ethernet Controller DMAC Module Stop	Target module: EDMAC 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b16	MSTPB16	Serial Peripheral Interface 1 Module Stop	Target module: RSPI1 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b17	MSTPB17	Serial Peripheral Interface 0 Module Stop	Target module: RSPI0 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b18	MSTPB18	Universal Serial Bus Interface (Port 1) Module Stop	Target module: USB1 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b19	MSTPB19	Universal Serial Bus Interface (Port 0) Module Stop	Target module: USB0 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b20	MSTPB20	I ² C Bus Interface 1 Module Stop	Target module: RIIC1 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b21	MSTPB21	I ² C Bus Interface 0 Module Stop	Target module: RIIC0 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b22	—	Reserved	This bit is always read as 1. The write value should always be 1.	R/W
b23	MSTPB23	CRC Calculator Module Stop	Target module: CRC 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W

Bit	Symbol	Bit Name	Description	R/W
b24	—	Reserved	This bit is always read as 1. The write value should always be 1.	R/W
b25	MSTPB25	Serial Communication Interface 6 Module Stop	Target module: SCI6 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b27	—	Reserved	This bit is always read as 1. The write value should always be 1.	R/W
b28	MSTPB28	Serial Communication Interface 3 Module Stop	Target module: SCI3 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b29	MSTPB29	Serial Communication Interface 2 Module Stop	Target module: SCI2 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W
b31	MSTPB31	Serial Communication Interface 0 Module Stop	Target module: SCI0 0: The module stop state is canceled 1: Transition to the module stop state is made	R/W

MSTPCRB is used to control the module stop state.

9.2.4 Module Stop Control Register C (MSTPCRC)

Address: 0008 0018h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1
	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1
	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0
	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MSTPC1	MSTPC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPC0*	RAM0 Module Stop	Target module: RAM0 (0000 0000h to 0000 FFFFh) 0: RAM0 run 1: RAM0 stop	R/W
b1	MSTPC1*	RAM1 Module Stop	Target module: RAM1 (0001 0000h to 0001 7FFFh) 0: RAM1 run 1: RAM1 stop	R/W
b15 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b31 to b16	—	Reserved	These bits are always read as 1. The write value should always be 1.	R/W

Note: * The MSTPC0 or MSTPC1 bit should not be set to 1 during access to the on-chip RAM0 or RAM1. The on-chip RAM 0 or RAM 1 should not be accessed with the MSTPC0 or MSTPC1 bit set to 1.

MSTPCRC is used to control the module stop state.

9.2.5 Deep Standby Control Register (DPSBYCR)

Address: 0008 C280h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSBY	IOKEEP	RAMCUT2	RAMCUT1	—	—	—	RAMCUT0
Value after reset:	0	0	1	1	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RAMCUT0	On-Chip RAM Off 0	b5 b4 b0 0 0 0: Power is supplied to the on-chip RAM (RAM0*) and USB resume detecting unit in deep software standby mode 1 1 1: Power is not supplied to the on-chip RAM (RAM0*) and USB resume detecting unit in deep software standby mode Settings other than above are prohibited.	R/W
b3 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	RAMCUT1	On-Chip RAM Off 1	See the description of the RAMCUT0 bit	R/W
b5	RAMCUT2	On-Chip RAM Off 2	See the description of the RAMCUT0 bit	R/W
b6	IOKEEP	I/O Port Retention	0: Deep software standby mode and I/O port retention are canceled simultaneously 1: I/O port retention is canceled when 0 is written to the IOKEEP bit after deep software standby mode is canceled	R/W
b7	DPSBY	Deep Software Standby	SSBY b7 0 0: Transition to sleep mode or all-module clock stop mode is made after the WAIT instruction is executed 0 1: Transition to sleep mode or all-module clock stop mode is made after the WAIT instruction is executed 1 0: Transition to software standby mode is made after the WAIT instruction is executed 1 1: Transition to deep software standby mode is made after the WAIT instruction is executed	R/W

Note: * For the on-chip RAM address space, see Table 9.2.

DPSBYCR is used to control deep software standby mode.

DPSBYCR is initialized by the reset signal from the RES# pin, a power-on reset, or a voltage monitoring reset, but is not initialized by the internal reset signal that cancels deep software standby mode.

RAMCUTj Bits (On-Chip RAM Off j) (j = 2 to 0)

These bits control the internal power supply to the on-chip RAM and USB resume detecting unit in deep software standby mode.

The on-chip RAM address space is divided into the RAM 0 area and RAM 1 area. For the on-chip RAM address space, see Table 9.2.

Only the internal power supply of RAM0 and USB resume detecting unit can be controlled by the setting of bits RAMCUT0, RAMCUT1, and RAMCUT2.

When a USB suspend/resume interrupt is used as a deep software standby mode canceling source, bits RAMCUT0, RAMCUT1, and RAMCUT2 must all be set to 0.

The internal power supply of RAM1 is stopped in deep software standby mode regardless of the setting of bits RAMCUT0, RAMCUT1, and RAMCUT2.

IOKEEP Bit (I/O Port Retention)

In deep software standby mode, I/O ports keep retaining the same states from software standby mode. The IOKEEP bit specifies whether to keep retaining the I/O port states from deep software standby mode even after deep software standby mode is canceled, or to cancel retaining the I/O port states.

DPSBY Bit (Deep Software Standby)

The DPSBY bit controls transitions to deep software standby mode.

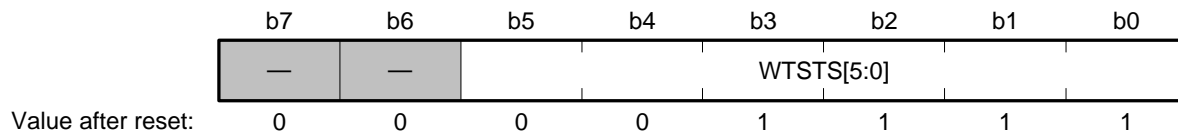
When the WAIT instruction is executed while the SSBY and DPSBY bits in SBYCR are 1, the LSI enters deep software standby mode through software standby mode.

This bit is not cleared to 0 when deep software standby mode is canceled by an external interrupt or some internal interrupts (voltage monitoring, RTC alarm, or USB suspend/resume). Write 0 to this bit to clear.

The setting of this bit is invalid when the WDT is used in watchdog timer mode, the IWDT is used, or the oscillation stop detection function is enabled. In this case, even when the SSBY and DPSBY bits in SBYCR are set to 1, the LSI always enters sleep mode or all-module clock stop mode after the WAIT instruction is executed.

9.2.6 Deep Standby Wait Control Register (DPSWCR)

Address: 0008 C281h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	WTSTS[5:0]	Deep Software Standby Waiting Time	b5 b0 0 0 0 1 0 1: Waiting time = 64 states 0 0 0 1 1 0: Waiting time = 512 states 0 0 0 1 1 1: Waiting time = 1024 states 0 0 1 0 0 0: Waiting time = 2048 states 0 0 1 0 0 1: Waiting time = 4096 states 0 0 1 0 1 0: Waiting time = 16384 states 0 0 1 0 1 1: Waiting time = 32768 states 0 0 1 1 0 0: Waiting time = 65536 states 0 0 1 1 0 1: Waiting time = 131072 states 0 0 1 1 1 0: Waiting time = 262144 states 0 0 1 1 1 1: Waiting time = 524288 states	R/W
b7, b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

DPSWCR is used to select the time for the LSI Group to wait until the clock is stabilized when deep software standby mode is canceled by an external interrupt or some internal interrupts (voltage monitoring, RTC alarm, or USB suspend/resume).

DPSWCR is initialized by the reset signal from the RES# pin, a power-on reset, or a voltage monitoring reset, but is not initialized by the internal reset signal that cancels deep software standby mode.

WTSTS[5:0] Bits (Deep Software Standby Waiting Time)

These bits select the time for the LSI Group to wait until the clock is stabilized when deep software standby mode is canceled by an external interrupt or some internal interrupts (voltage monitoring, RTC alarm, or USB suspend/resume). When using deep software standby mode, set the WTSTS[5:0] bits before a transition to deep software standby mode is made.

In the case of crystal oscillation, see Table 9.4 and make a selection according to the operating frequency so that the waiting time is no less than the oscillation settling time. When an external clock is used, the PLL circuit settling time is necessary. Select a waiting time referring to Table 9.4.

During the oscillation settling time, the counter is counted on the EXTAL input clock frequency.

9.2.7 Deep Standby Interrupt Enable Register (DPSIER)

Address: 0008 C282h

	b7	b6	b5	b4	b3	b2	b1	b0
	DNMIE	DUSBE	DRTCE	DLVDE	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0E	IRQ0 Pin Enable	0: Canceling deep software standby mode by the IRQ0 pin is disabled 1: Canceling deep software standby mode by the IRQ0 pin is enabled	R/W
b1	DIRQ1E	IRQ1 Pin Enable	0: Canceling deep software standby mode by the IRQ1 pin is disabled 1: Canceling deep software standby mode by the IRQ1 pin is enabled	R/W
b2	DIRQ2E	IRQ2 Pin Enable	0: Canceling deep software standby mode by the IRQ2 pin is disabled 1: Canceling deep software standby mode by the IRQ2 pin is enabled	R/W
b3	DIRQ3E	IRQ3 Pin Enable	0: Canceling deep software standby mode by the IRQ3 pin is disabled 1: Canceling deep software standby mode by the IRQ3 pin is enabled	R/W
b4	DLVDE	LVD Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the voltage detection interrupt is disabled 1: Canceling deep software standby mode by the voltage detection interrupt is enabled	R/W
b5	DRTCE	RTC Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the RTC alarm interrupt is disabled 1: Canceling deep software standby mode by the RTC alarm interrupt is enabled	R/W
b6	DUSBE	USB Suspend/Resume Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the USB suspend/resume is disabled 1: Canceling deep software standby mode by the USB suspend/resume is enabled	R/W
b7	DNMIE	NMI Pin Enable	0: Canceling deep software standby mode by the NMI pin is disabled 1: Canceling deep software standby mode by the NMI pin is enabled	R/(W)*

Note: * A 1 can be written only once. Once 1 is written to the DNMIE bit, subsequent write accesses are disabled.

DPSIER is used to enable or disable the external interrupt pin and internal cancellation signals that cancel deep software standby mode.

DPSIER is initialized by the reset signal from the RES# pin, a power-on reset, or a voltage monitoring reset, but is not initialized by the internal reset signal that cancels deep software standby mode.

Note that modifying the setting of the DPSIER register changes the internal state of control over the input buffer of the corresponding pin. In such cases, the change to the pin state may lead to the internal generation of a spurious edge and setting of the corresponding flag in the DPSIFR register to 1. Accordingly, ensure that the value of the DPSIFR register is 0 before placing the chip in deep software standby mode.

Also note that a transition to deep software standby mode disables the input buffer of the pin for which the corresponding bit of the DPSIER register is 0. The transition may lead to the internal generation of a rising edge depending on the pin state, and to setting of the corresponding flag in the DPSIFR register to 1. If the value of the corresponding bit in the DPSIEGR register is 0, however, a rising edge will not be detected and the value of the flag in the DPSIFR register will not become 1.

9.2.8 Deep Standby Interrupt Flag Register (DPSIFR)

Address: 0008 C283h

	b7	b6	b5	b4	b3	b2	b1	b0
	DNMIF	DUSBF	DRTCF	DLVDF	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0F	IRQ0 Deep Standby Cancel Flag	0: No cancel request by the IRQ0 pin is generated 1: A cancel request by the IRQ0 pin is generated	R/(W)*
b1	DIRQ1F	IRQ1 Deep Standby Cancel Flag	0: No cancel request by the IRQ1 pin is generated 1: A cancel request by the IRQ1 pin is generated	R/(W)*
b2	DIRQ2F	IRQ2 Deep Standby Cancel Flag	0: No cancel request by the IRQ2 pin is generated 1: A cancel request by the IRQ2 pin is generated	R/(W)*
b3	DIRQ3F	IRQ3 Deep Standby Cancel Flag	0: No cancel request by the IRQ3 pin is generated 1: A cancel request by the IRQ3 pin is generated	R/(W)*
b4	DLVDF	LVD Deep Standby Cancel Flag	0: No cancel request by the voltage monitor signal is generated 1: A cancel request by the voltage monitor signal is generated	R/(W)*
b5	DRTCF	RTC Deep Standby Cancel Flag	0: No cancel request by the RTC alarm interrupt is generated 1: A cancel request by the RTC alarm interrupt is generated	R/(W)*
b6	DUSBF	USB Suspend/Resume Deep Standby Cancel Flag	0: No cancel request by the USB suspend/resume is generated 1: A cancel request by the USB suspend/resume is generated	R/(W)*
b7	DNMIF	NMI Deep Standby Cancel Flag	0: No cancel request by the NMI pin is generated 1: A cancel request by the NMI pin is generated	R/(W)*

Note: * Only 0 can be written to clear the flag.

DPSIFR is used to hold the request for canceling deep software standby mode.

Each flag is set to 1 when a cancel request specified by the deep standby interrupt edge register (DPSIEGR) is generated. Since each flag is set to 1 when a cancel request is generated in any mode, a transition to deep software standby mode should be made after DPSIFR is cleared to 0.

DPSIFR is initialized by the reset signal from the RES# pin, a power-on reset, or a voltage monitoring reset, but is not initialized by the internal reset signal that cancels deep software standby mode.

DIRQnF Flags (IRQn Deep Standby Cancel Flag) (n = 0 to 3)

These flags indicate that a cancel request by the IRQn pin has been generated.

[Setting condition]

- A cancel request by the IRQn pin specified by DPSIEGR is generated

[Clearing condition]

- Each bit is read as 1 and then written by 0

DLVDF Flag (LVD Deep Standby Cancel Flag)

This flag indicates that a cancel request by the voltage monitor signal has been generated.

[Setting condition]

- A cancel request by the voltage monitor signal is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

DRTCF Flag (RTC Deep Standby Cancel Flag)

This flag indicates that a cancel request by the RTC alarm interrupt has been generated.

[Setting condition]

- A cancel request by the RTC alarm interrupt is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

DUSBF Flag (USB Suspend/Resume Deep Standby Cancel Flag)

This flag indicates that a cancel request by the USB suspend/resume has been generated.

[Setting condition]

- A cancel request by the USB suspend/resume is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

DNMIF Flag (NMI Deep Standby Cancel Flag)

This flag indicates that a cancel request by the NMI pin has been generated.

[Setting condition]

- A cancel request by the NMI pin specified by DPSIEGR is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

9.2.9 Deep Standby Interrupt Edge Register (DPSIEGR)

Address: 0008 C284h

	b7	b6	b5	b4	b3	b2	b1	b0
	DNMIEG	—	—	—	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0EG	IRQ0 Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b1	DIRQ1EG	IRQ1 Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b2	DIRQ2EG	IRQ2 Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b3	DIRQ3EG	IRQ3 Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b6 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	DNMIEG	NMI Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

DPSIEGR is used to select an edge of the cancel signal for canceling deep software standby mode.

DPSIEGR is initialized by the reset signal from the RES# pin, a power-on reset, or a voltage monitoring reset, but is not initialized by the internal reset signal that cancels deep software standby mode.

9.2.10 Reset Status Register (RSTSR)

Address: 0008 C285h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSRSTF	—	—	—	—	LVD2F	LVD1F	PORF
Value after reset:	0/1*	0	0	0	0	0/1*	0/1*	0/1*

Note: * The initial value depends on reset generation sources.

Bit	Symbol	Bit Name	Description	R/W
b0	PORF	Power-On Reset Flag	0: A power-on reset is not generated. 1: A power-on reset is generated.	R
b1	LVD1F	LVD1 Detection Flag	0: LVD1 is not detected. 1: LVD1 is detected.	R/(W)*
b2	LVD2F	LVD2 Detection Flag	0: LVD2 is not detected. 1: LVD2 is detected.	R/(W)*
b6 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	DPSRSTF	Deep Software Standby Reset Flag	0: A deep software standby mode canceling source is not generated by an external interrupt or some internal interrupts (voltage monitoring, RTC alarm, or USB suspend/resume). 1: A deep software standby mode canceling source is generated by an external interrupt or some internal interrupts (voltage monitoring, RTC alarm, or USB suspend/resume).	R/(W)*

Note: * Only 0 can be written to clear the flag.

RSTSR indicates an internal reset generation source.

PORF Flag (Power-On Reset Flag)

The PORF flag indicates that a power-on reset has been generated.

This flag is initialized by the reset signal from the RES# pin, but is not initialized by the internal reset signal that cancels deep software standby mode.

[Setting condition]

- A power-on reset is generated.

[Clearing condition]

- This bit is reset by the RES# pin.

LVD1F Flag (LVD1 Detection Flag)

The LVD1F flag indicates that a Vcc voltage lower than the Vdet1 level has been detected.

This flag is initialized by the reset signal from the RES# pin or a power-on reset, but is not initialized by the internal reset signal that cancels deep software standby mode.

[Setting condition]

- Vcc voltage equal to or lower than the Vdet1 level is detected.

[Clearing conditions]

- This bit is read as 1 and then written by 0 in a state where Vcc has exceeded Vdet1 and the settling time has passed, while the LVDCR.LVD1E bit is 1.
- A reset by the RES# pin is generated.
- A power-on reset is generated.

LVD2F Flag (LVD2 Detection Flag)

The LVD2F flag indicates that a Vcc voltage lower than the Vdet2 level has been detected.

This flag is initialized by the reset signal from the RES# pin or a power-on reset, but is not initialized by the internal reset signal that cancels deep software standby mode.

[Setting condition]

- Vcc voltage equal to or lower than the Vdet2 level is detected.

[Clearing conditions]

- This bit is read as 1 and then written by 0 in a state where Vcc has exceeded Vdet2 and the settling time has passed, while the LVDCR.LVD2E bit is 1.
- A reset by the RES# pin is generated.
- A power-on reset is generated.
- A reset by LVD1 is generated.

DPSRSTF Flag (Deep Software Standby Reset Flag)

The DPSRSTF flag indicates that deep software standby mode has been canceled by an external interrupt source specified by DPSIER and DPSIEGR or some internal interrupts (voltage monitoring, RTC alarm, or USB suspend/resume), and an internal reset has been generated.

This flag is initialized by the reset signal from the RES# pin, a power-on reset, or a voltage monitoring reset, but is not initialized by the internal reset signal that cancels deep software standby mode.

[Setting condition]

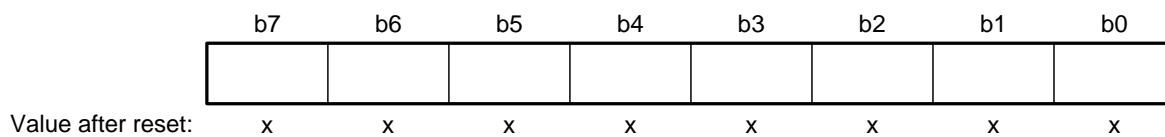
- Deep software standby mode is canceled by an external interrupt source or some internal interrupts (voltage monitoring, RTC alarm, or USB suspend/resume).

[Clearing conditions]

- This bit is read as 1 and then written by 0.
- A reset by the RES# pin is generated.
- A power-on reset is generated.
- A reset by LVD1 is generated.
- A reset by LVD2 is generated.

9.2.11 Deep Standby Backup Register (DPSBK_{Ry}) (y = 0 to 31)

Addresses: 0008 C290h to 0008 C2AFh



DPSBK_{Ry} is a 32-byte readable/writable register to store data during deep software standby mode. The value of this register is retained even in deep software standby mode where on-chip RAM data is not retained. DPSBK_{Ry} is not initialized and the register value is undefined immediately after power-on.

9.3 Multi-Clock Function

When the ICK[3:0], BCK[3:0], and PCK[3:0] bits in SCKCR are set, the clock frequency changes.

The CPU and bus masters operate on the operating clock specified by the ICK[3:0] bits. Peripheral modules operate on the operating clock specified by the PCK[3:0] bits. The external bus operates on the operating clock specified by the BCK[3:0] bits. For details, see section 8, Clock Generation Circuit.

9.4 Module Stop Function

The module stop function can be set for each on-chip peripheral module.

When the MSTPyj bit (y = A to C, j = 31 to 0) in MSTPCRA to MSTPCRC is set to 1, the specified module stops operating and enters the module stop state, but the CPU continues to operate independently. Clearing the MSTPyj bit to 0 cancels the module stop state, allowing the module to restart operating at the end of the bus cycle.

The internal states of modules are retained in the module stop state.

After a reset, all modules other than the DMACA, DTC, EXDMAC, and on-chip RAM are placed in the module stop state. No read/write access can be made to the registers of the module that are in the module stop state.

9.5 Low Power Consumption Modes

9.5.1 Sleep Mode

9.5.1.1 Transition to Sleep Mode

When the WAIT instruction is executed while the SSBY bit in SBYCR 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.

9.5.1.2 Canceling Sleep Mode

Sleep mode is canceled by any interrupt, the reset signal from the RES# pin, a power-on reset, a voltage monitoring reset, a reset caused by a WDT overflow, or a reset caused by an IWDT underflow.

- **Canceling by an interrupt**
When an interrupt occurs, sleep mode is canceled and the interrupt exception handling starts. If a maskable interrupt has been masked by the CPU (the priority level*¹ of the interrupt has been set to a value lower than that of the IPL[3:0] bits*² in PSW of the CPU), sleep mode is not canceled.
- **Canceling by the RES# pin**
When the RES# pin is driven low, the LSI enters the reset state. When the RES# pin is driven high after the reset signal is input for a predetermined time period, the CPU starts the reset exception handling.
- **Canceling by a WDT overflow reset**
Sleep mode is canceled by an internal reset generated by a WDT overflow.
- **Canceling by an IWDT underflow reset**
Sleep mode is canceled by an internal reset generated by an IWDT underflow.
- **Canceling by a voltage monitoring reset**
Sleep mode is canceled by a voltage monitoring reset from the voltage detection circuit.
- **Canceling by a power-on reset**
Sleep mode is canceled by a power-on reset.

Note 1. For details, see section 11, Interrupt Control Unit (ICUa).

Note 2. For details, see section 2, CPU.

9.5.2 All-Module Clock Stop Mode

9.5.2.1 Transition to All-Module Clock Stop Mode

When the following two conditions are satisfied, executing the WAIT instruction with the SSBY bit in SBYCR cleared to 0 will cause the transition to all-module clock stop mode at the end of the bus cycle.*¹

- The ACSE bit in MSTPCRA is set to 1.
- All the modules controlled by the MSTPCRA and MSTPCRB registers except for the 8-bit timers (units 0 and 1) are set in the module stop state (MSTPCRA = FFFFFFF[C to F]Fh, MSTPCRB = FFFFFFFFh).

In all-module clock stop mode, all modules (other than the 8-bit timers*², WDT, IWDT, RTC, power-on reset circuit, and voltage detection circuit), the CPU, the bus controller, and the I/O ports are stopped.

If a further reduction in supply current is required beyond that in all-module clock stop mode, stop the target modules for which operation or stopping is controlled by MSTPCRC.

When all-module clock stop mode is in use, issue a WAIT instruction after making the following settings.

1. Clear the I bit*³ in PSW of the CPU to 0.
2. Set the interrupt destination to be used for recovery from all-module clock stop mode to the CPU.
3. Set the priority*⁴ of the interrupt to be used for recovery from all-module clock stop mode to a level higher than the setting of the IPL[3:0] bits*³ in PSW.
4. Set the IENn bit*⁴ in IERm for the interrupt to be used for recovery from all-module clock stop mode to 1.
5. Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the I bit*³ in PSW of the CPU to 1).

Note 1. Transitions to all-module clock stop mode are not be made in some states of DTC, DMACA, EXDMAC, or EDMAC operations. Before setting the MSTPA29 or MSTPA28 bit in MSTPCRA and the MSTPB15 bit in MSTPCRB to 1, clear the DMST bit in DMAST of the DMACA, the DTCST bit in DTCST of the DTC, and the DMST bit in EDMAST of the EXDMAC to 0 so that the DTC, DMACA, EXDMAC, and EDMAC are not initiated.

Note 2. The MSTPA4 and MSTPA5 bits of MSTPCRA select operation or stopping of these modules.

Note 3. For details, see section 2, CPU.

Note 4. For details, see section 11, Interrupt Control Unit (ICUa).

9.5.2.2 Canceling All-Module Clock Stop Mode

All-module clock stop mode is canceled by an external interrupt (the NMI pin or any pin from among IRQ0 to IRQ15), the reset signal from the RES# pin, a voltage monitoring reset, a power-on reset, or some internal interrupts (from the 8-bit timer*1, WDT, RTC alarm, voltage monitoring, oscillation stop detection, or USB interrupt (USBR)), and normal program execution resumes once handling of the given exception is complete. However, note that in cases where a maskable interrupt has been masked by the CPU (the priority level*2 of the interrupt has been set to a value lower than that of the IPL[3:0] bits*3 in PSW of the CPU) or a maskable interrupt has been set up as a trigger for transfer by the DTC, DMACA, or EXDMAC, the interrupt will not trigger release from all-module clock stop mode.

Note 1. The MSTPA4 and MSTPA5 bits of MSTPCRA select operation or stopping of these modules.

Note 2. For details, see section 11, Interrupt Control Unit (ICUa).

Note 3. For details, see section 2, CPU.

9.5.3 Software Standby Mode

9.5.3.1 Transition to Software Standby Mode

When the WAIT instruction is executed with the SSBY bit in SBYCR set to 1 and the DPSBY bit in DPSBYCR cleared to 0, a transition to software standby mode is made. In this mode, the CPU, on-chip peripheral functions, and all the oscillator functions stop. However, the contents of the CPU internal registers, on-chip RAM data, on-chip peripheral functions, and the states of the I/O ports are retained. Whether the address bus and bus control signals are placed in the high-impedance state or the output state is retained can be specified by the OPE bit in SBYCR. This mode allows significant reduction in power consumption because the oscillator stops in this mode.

Clear the DMST bit in DMAST of the DMACA, the DTCST bit in DTCST of the DTC, the DMST bit in EXDMAST of the EXDMAC, and the TR bit in EDTRR and the RR bit in EDRRR of the EDMAC to 0 before executing the WAIT instruction.

When the WDT is used in watchdog timer mode or the IWDG is used, no transition to software standby mode is made. Stop the WDT before executing the WAIT instruction.

When the oscillation stop detection function is enabled*¹, software standby mode cannot be entered. To make a transition to software standby mode, issue a WAIT instruction after disabling the oscillation stop detection function.

When software standby mode is in use, issue a WAIT instruction after making the following settings.

1. Clear the I bit*² in PSW of the CPU to 0.
2. Set the interrupt destination to be used for recovery from software standby mode to the CPU.
3. Set the priority*³ of the interrupt to be used for recovery from software standby mode to a level higher than the setting of the IPL[3:0] bits*² in PSW of the CPU.
4. Set the IENn bit*³ in IERm for the interrupt to be used for recovery from software standby mode to 1.
5. Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the I bit*² in PSW of the CPU to 1).

Note 1. The oscillation stop detection function (the OSTDCR.OSTED bit) is valid after a reset is canceled.

Note 2. For details, see section 2, CPU.

Note 3. For details, see section 11, Interrupt Control Unit (ICUa).

9.5.3.2 Canceling Software Standby Mode

Software standby mode is canceled by an external interrupt (NMI or IRQ0 to IRQ15*), some internal interrupts (voltage monitoring, RTC alarm, or USB interrupt (USBR)), the reset signal from the RES# pin, a power-on reset, or a voltage monitoring reset.

1. Canceling by an interrupt

When an interrupt request signal is input by an NMI, IRQ0 to IRQ15*, voltage monitoring, RTC alarm, or USB interrupt (USBR), clock oscillation starts thus supplying stable clocks to the entire LSI chip after the time specified by the SBYCR.STS[4:0] bits has passed. Software standby mode is then canceled and the interrupt exception handling is started.

2. Canceling by the RES# pin

When the RES# pin is driven low, clock oscillation starts and at the same time the clocks are supplied to the entire LSI chip. Be sure to hold the RES# pin low until the clock oscillation settles. When the RES# pin is driven high, the CPU begins the reset exception handling.

3. Canceling by a power-on reset

When a power-on reset is generated by a power-supply voltage drop, software standby mode is canceled.

4. Canceling by a voltage monitoring reset

When a voltage monitoring reset is generated by a power-supply voltage drop, software standby mode is canceled and clock oscillation starts.

Note: * For details, see section 11, Interrupt Control Unit (ICUa).

9.5.3.3 Setting Oscillation Settling Time after Software Standby Mode is Canceled

Set the STS[4:0] bits in SBYCR as follows:

1. When using a crystal resonator

Set the STS[4:0] bits so that the waiting time is no less than the oscillation settling time.

Table 9.4 shows operating frequencies and waiting time corresponding to each setting of the STS[4:0] bits.

2. When using an external clock

The PLL circuit settling time is necessary. Set the waiting time referring to Table 9.4.

Table 9.4 Oscillation Settling Time Setting

STS4	STS3	STS2	STS1	STS0	Waiting Time (States)	PCLK* (MHz)			Unit		
						50	25	8			
0	0	0	0	0	Reserved	—	—	—	μs		
				1	Reserved	—	—	—			
				1	0	Reserved	—	—		—	
					1	Reserved	—	—		—	
				1	0	0	Reserved	—		—	—
						1	64	1.3		2.6	8.0
	1	0	0	0	2048	40.95	81.9	256.0			
				1	4096	0.08	0.16	0.51			
				1	16384	0.33	0.66	2.05			
		1	0	0	65536	1.31	2.62	8.19			
				1	131072	2.62	5.24	16.38			
				1	262144	5.25	10.49	32.77			
1	x	x	x	x	Reserved	—	—	—			
					1	524288	10.49	20.97	65.54		
					1	Reserved	—	—	—		

Light gray shaded cells: Recommended time setting when an external clock is used

Dark gray shaded cells: Recommended time setting when a crystal resonator is used

Note: * The PCLK is the output of the peripheral module frequency divider.

The oscillation settling time (including oscillator's unstable oscillation time) depends on the resonator characteristics.

The PCLK values in this table are reference values.

9.5.3.4 Example of Software Standby Mode Application

Figure 9.2 shows an example where a transition to software standby mode is made at the falling edge on the IRQ pin, and software standby mode is canceled at the rising edge on the IRQ pin.

In this example, an IRQ interrupt is accepted with the IRQMD[1:0] bits in IRQCRi of the ICU set to "01b" (falling edge), and then the IRQMD[1:0] bits are set to "10b" (rising edge). After that, the SSBY bit in SBYCR are set to 1, and then the WAIT instruction is executed. Thus a transition to software standby mode is made.

After that, software standby mode is canceled at the rising edge on the IRQ pin.

To return from software standby mode, settings of the interrupt control unit (ICU) are also necessary. For details, see section 11, Interrupt Control Unit (ICUa).

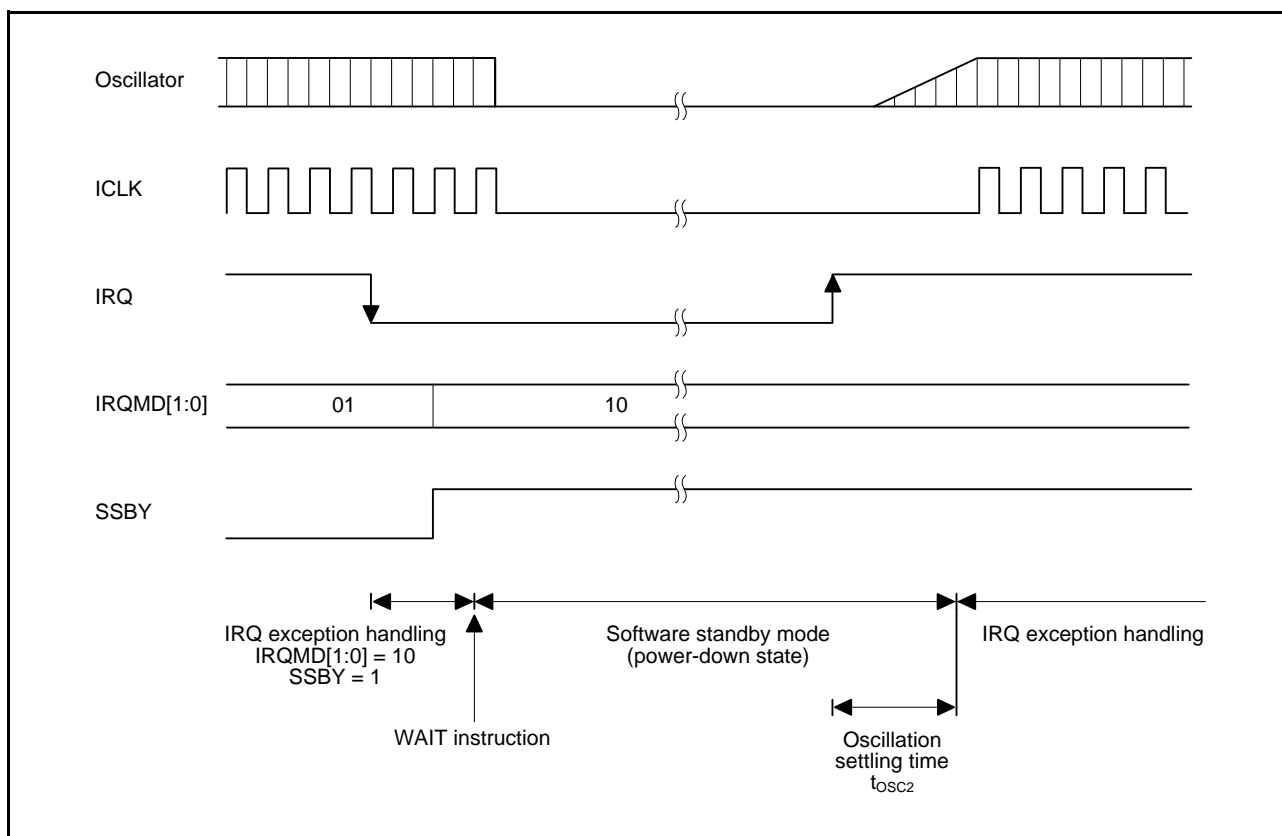


Figure 9.2 Example of Software Standby Mode Application

9.5.4 Deep Software Standby Mode

9.5.4.1 Transition to Deep Software Standby Mode

When the WAIT instruction is executed with the SSBY bit in SBYCR set to 1, a transition to software standby mode*1 is made. At this time, when the DPSBY bit in DPSBYCR is set to 1, a transition to deep software standby mode is made. However, if a software standby mode canceling source (NMI, IRQ0 to IRQ15 interrupt requests, or some internal interrupts of voltage monitoring, RTC alarm, or USB interrupt (USBR)) is generated concurrently when a transition to software standby mode is made, software standby mode is canceled regardless of the DPSBY setting, and the interrupt exception handling starts after the oscillation settling time for software standby mode specified by the STS[4:0] bits in SBYCR has passed.

When the SSBY and DPSBY bits are set to 1 and no software standby mode canceling source is generated, a transition to deep software standby mode is made immediately after transition to software standby mode.

In deep software standby mode, the CPU, on-chip peripheral functions (other than the USB resume detecting unit and RTC), on-chip RAM 1*2, and all the oscillator functions stop. Furthermore, the internal power supply to these modules stops, allowing significant reduction in power consumption. At this time, the contents of all the registers of the CPU and on-chip peripheral functions (other than the USB resume detecting unit and RTC) become undefined. All the on-chip RAM 1*2 data becomes undefined regardless of the setting of the RAMCUT2 to RAMCUT0 bits in DPSBYCR.

The on-chip RAM 0*2 data can be retained by clearing the RAMCUT2 to RAMCUT0 bits to all 0 beforehand. Setting all of these bits to 1 will stop the internal power supply to the on-chip RAM 0*2 and USB resume detecting unit, allowing further reduction in power consumption. At this time, the on-chip RAM 0*2 data is undefined.

The voltage detection circuit and power-on reset circuit are enabled even in deep software standby mode.

The I/O port states remain unchanged from software standby mode.

Note 1. Conditions on the DTC, DMACA, EXDMAC, EDMAC, WDT, IWDT, and the oscillation stop detection function for transition to software standby mode should be met before the WAIT instruction is executed. For details, see section 9.5.3, **Software Standby Mode**.

Note 2. The on-chip RAM address space is divided into the RAM 0 area and RAM 1 area. For the on-chip RAM address space, see Table 9.2.

9.5.4.2 Canceling Deep Software Standby Mode

Deep software standby mode is canceled by an external interrupt (NMI or IRQ0-A to IRQ3-A pins), some internal interrupts (voltage monitoring, RTC alarm, or USB suspend/resume), the reset signal from the RES# pin, a power-on reset, or a voltage monitoring reset.

1. Canceling by an external or internal interrupt

The DPSIFR holds the cancelation cause of deep software standby mode and the bits in this register are set to 1 when the corresponding cancelation requests are generated. When a bit is set to 1 and the corresponding cancelation cause is enabled in the DPSIER register, deep software standby mode is canceled.

The DNMIF or DIRQnF flag in DPSIFR is set to 1 when an edge is generated on the NMI pin or IRQ0-A to IRQ3-A pins enabled by the DNMIE bit in DPSIER or the DIRQnE bit (n = 3 to 0) in DPSIER. Rising edge or falling edge is selectable with DPSIEGR for each pin.

The DRTCF flag is set to 1 when an RTC alarm interrupt enabled by the DPSIER.DRTCE bit is generated. The DRTCIF flag is set to 1 when an RTC alarm interrupt enabled by the DRTCIE bit in DPSIER is generated. The DLVDIF flag is set to 1 when a voltage monitoring interrupt enabled by the DLDVIE bit in DPSIER is generated. When a deep software standby mode canceling source is generated, clock oscillation starts and the internal power supply begins at the same time, and then the internal reset signal is generated throughout the LSI. After the time specified by the WTSTS[5:0] bits in DPSWCR has passed, stable clocks are supplied to the entire LSI and the internal reset is released. At the same time, deep software standby mode is canceled and the reset exception handling starts.

When deep software standby mode is canceled by an external or internal interrupt, the DPSRSTF flag in RSTSR is set to 1.

2. Canceling by the RES# pin

When the RES# pin is driven low, clock oscillation starts and the internal power supply begins at the same time. Clocks are supplied to this LSI simultaneously with the start of clock oscillation. Be sure to hold the RES# pin low until the clock oscillation settles. When the RES# pin is driven high, the CPU begins the reset exception handling.

3. Canceling by a power-on reset

When a power-on reset is generated by a power-supply voltage drop, deep software standby mode is canceled.

4. Canceling by a voltage monitoring reset

When a voltage monitoring reset is generated by a power-supply voltage drop, deep software standby mode is canceled.

9.5.4.3 Pin States when Deep Software Standby Mode is Canceled

In deep software standby mode, I/O ports retain the same states from software standby mode. The inside of the LSI is initialized by an internal reset generated when deep software standby mode is canceled. Upon cancellation of deep software standby mode, the reset exception handling starts. The following shows the states of I/O ports at this time.

Whether to initialize the I/O ports or to keep retaining the I/O port states at the time of software standby mode can be selected by the IOKEEP bit in DPSBYCR.

- When IOKEEP = 0
I/O ports are initialized by an internal reset generated when deep software standby mode is canceled.
- When IOKEEP = 1
Although the inside of the LSI is initialized by an internal reset generated when deep software standby mode is canceled, I/O ports keep retaining their states from software standby mode regardless of the LSI internal state. At this time, the I/O port states remain unchanged from software standby mode even if settings of I/O ports or peripheral modules are made. Then the retained I/O port states are released by clearing the IOKEEP bit to 0, and the LSI operates according to the internal state.

The IOKEEP bit is not initialized by an internal reset generated when deep software standby mode is canceled.

9.5.4.4 Setting Oscillation Settling Time after Deep Software Standby Mode is Canceled

Set the WTSTS[5:0] bits in DPSWCR as follows:

1. When using a crystal resonator

Set the WTSTS[5:0] bits so that the waiting time is no less than the oscillation settling time.

Table 9.5 shows EXTAL input clock frequencies and waiting time corresponding to each setting of the WTSTS[5:0] bits.

2. When using an external clock

The PLL circuit settling time is necessary. Set the waiting time referring to Table 9.5.

Table 9.5 Oscillation Settling Time Setting

WTSTS5	WTSTS4	WTSTS3	WTSTS2	WTSTS1	WTSTS0	Waiting Time (States)	EXTAL Input Clock Frequency* (MHz)				Unit										
							14	12	10	8											
0	0	0	0	0	0	Reserved	—	—	—	—	μs										
						1	Reserved	—	—	—		—									
						1	0	Reserved	—	—		—	—								
							1	Reserved	—	—		—	—								
						1	0	0	0	0		0	Reserved	—	—	—	—				
												1	64	4.6	5.3	6.4	8.0				
												1	0	512	36.6	42.7	51.2	64.0			
													1	1024	73.1	85.3	102.4	128.0			
												1	0	0	0	0	2048	146.3	170.7	204.8	256.0
																	1	4096	0.29	0.34	0.41
						1	0	0	0	0		0	16384	1.17	1.37	1.64	2.05				
													1	32768	2.34	2.73	3.26	4.10			
1	0	0	0	0	0						65536		4.68	5.46	6.55	8.19					
											1		131072	9.36	10.92	13.11	16.38				
1	0	0	0	0	0						262144		18.72	21.85	26.21	32.77					
											1		524288	37.45	43.69	52.43	65.54				
1	x	x	x	x	x	Reserved	—	—	—	—											
						Reserved	—	—	—	—											

 : Recommended time setting when an external clock is used

 : Recommended time setting when a crystal resonator is used

Note: * The oscillation settling time (including oscillator's unstable oscillation time) depends on the resonator characteristics. The values of the EXTAL input clock frequency in this table are reference values.

9.5.4.5 Canceling Deep Software Standby Mode by USB

Deep software standby mode can be canceled by a USB suspend/resume interrupt.

A USB suspend/resume interrupt is detected at the USB resume detecting unit. For details, refer to section 28.3.1.4, Canceling Deep Software Standby Mode by a USB Suspend/Resume Interrupt.

9.5.4.6 Example of Deep Software Standby Mode Application

Figure 9.3 shows an example where a transition to deep software standby mode is made at the falling edge on the IRQ pin, and deep software standby mode is canceled at the rising edge on the IRQ pin.

In this example, an IRQ interrupt is accepted with the IRQMD[1:0] bits in IRQCRn of the ICU set to "01b" (falling edge), and then the DIRQnEG bit (n = 3 to 0) in DPSIEGR is set to 1 (rising edge). After that, the SSBY bit in SBYCR and the DPSBY bit in DPSBYCR are set to 1, and then the WAIT instruction is executed. Thus a transition to deep software standby mode is made.

After that, deep software standby mode is canceled at the rising edge on the IRQ pin.

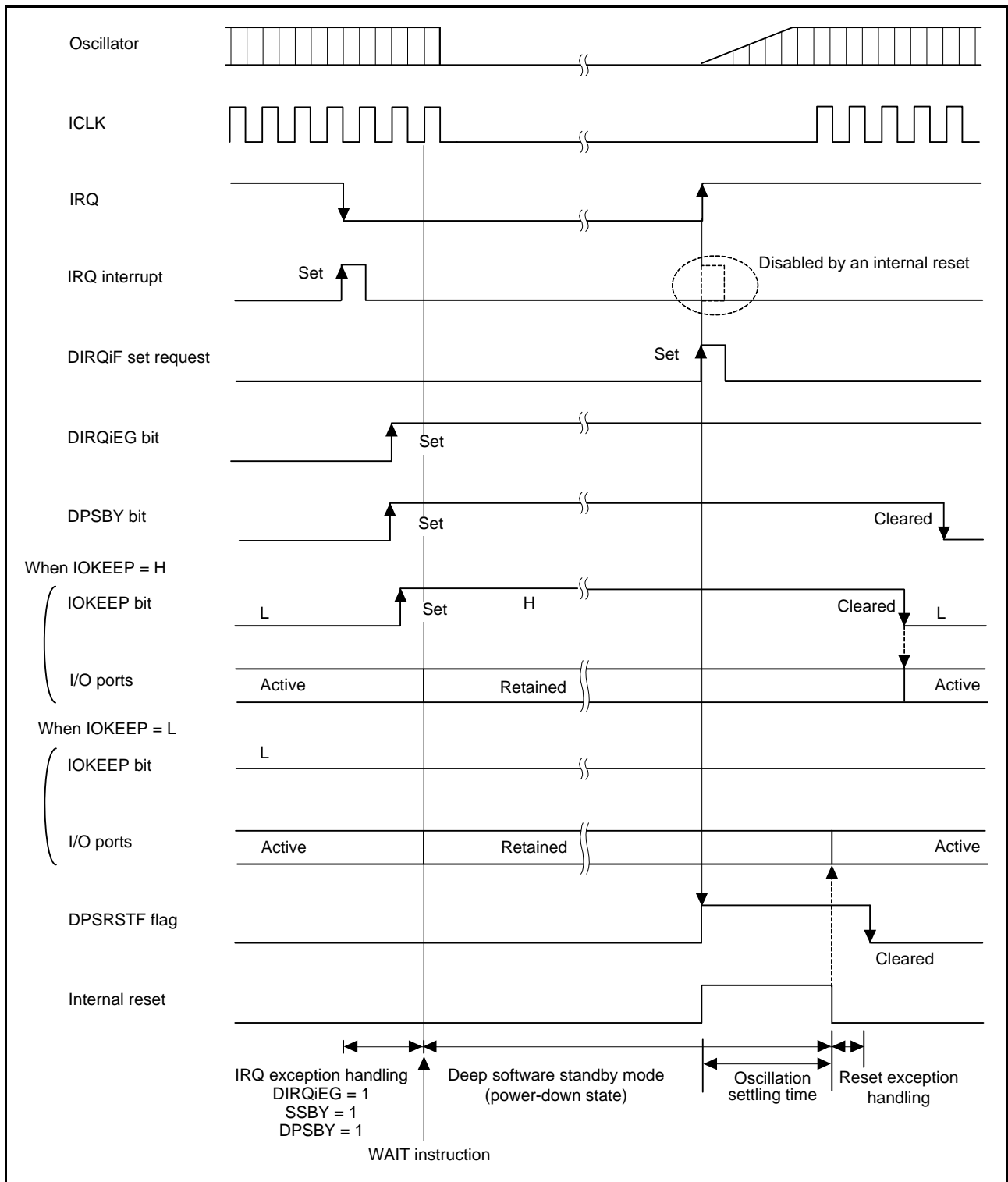


Figure 9.3 Example of Deep Software Standby Mode Application

9.5.4.7 Flowchart to Use Deep Software Standby Mode

Figure 9.4 shows an example of a flowchart to use deep software standby mode.

In this example, the DPSRSTF flag in RSTSR of the reset function is read after the reset exception handling to determine whether a reset was generated by the RES# pin or by the cancellation of deep software standby mode.

In the case of a reset by the RES# pin, a transition to deep software standby mode is made after required register settings.

In the case of a reset by the cancellation of deep software standby mode, the IOKEEP bit in DPSBYCR is cleared to 0 after the I/O port settings.

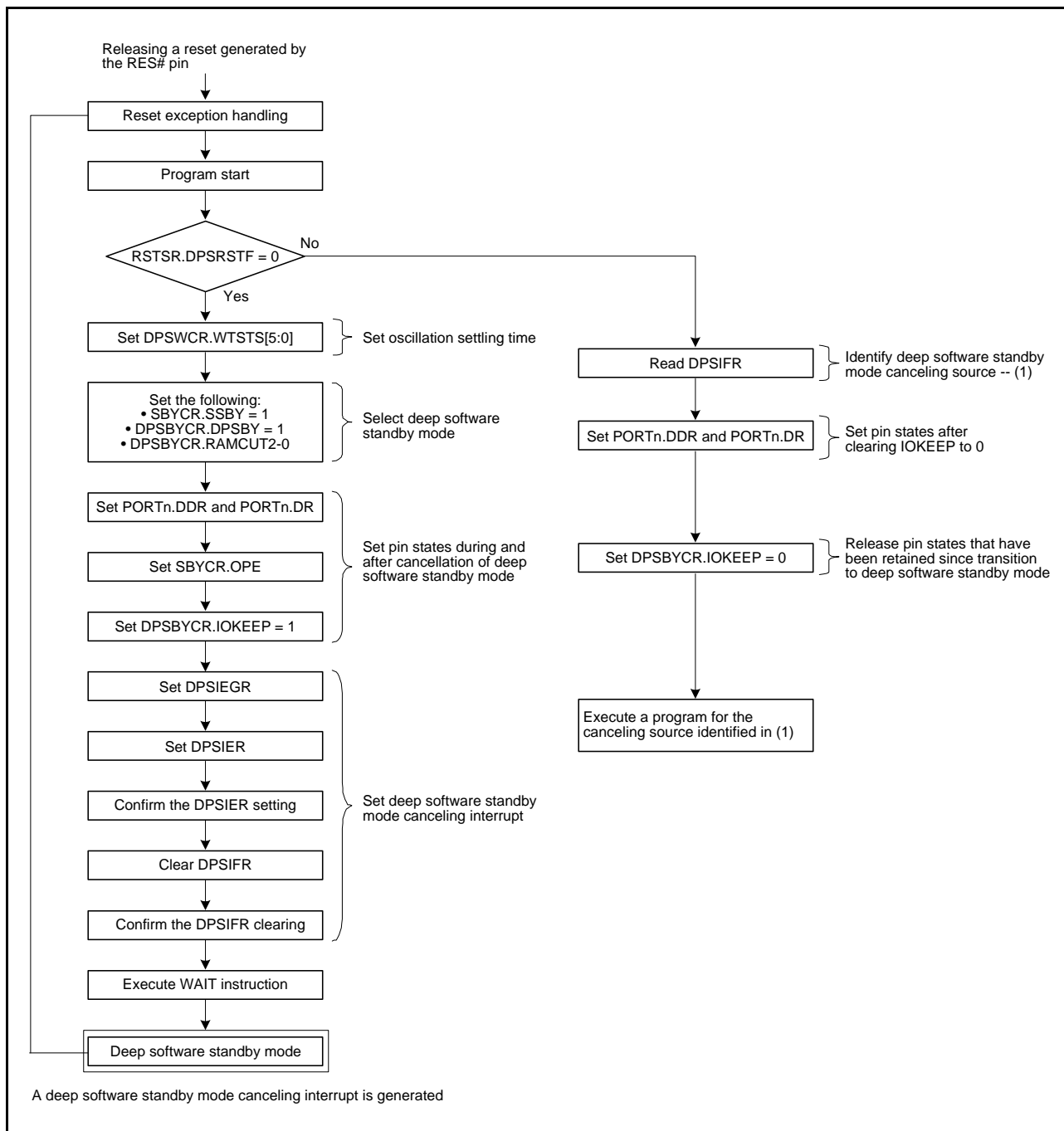


Figure 9.4 Example of Flowchart to Use Deep Software Standby Mode

9.6 BCLK and SDCLK Output Control

The BCLK output can be controlled with the PSTOP1 bit in SCKCR and the B3 bit in PORT5.DDR of corresponding P53. When the PSTOP1 bit is cleared to 0, P53 functions as the BCLK output. When the PSTOP1 bit is set to 1, the BCLK output stops and goes high. When the B3 bit in PORT5.DDR of P53 is cleared to 0, the BCLK output is disabled and the pin functions as an input port.

Table 9.6 shows the BCLK pin state in each low power consumption mode.

Table 9.6 BCLK Pin (P53) State in Each Low Power Consumption Mode

Register Settings		Normal Operating Mode	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode		Deep Software Standby Mode	
DDR	PSTOP1				OPE = 0	OPE = 1	IOKEEP = 0	IOKEEP = 1
0	x	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
1	0	BCLK output	BCLK output	BCLK output	High	High	High	High
1	1	High	High	High	High	High	High	High

The SDCLK output is controllable by the PSTOP0 bit in the SCKCR and whichever of the PORT7.DDR.B0 bit for P70 and PF6BUS.SDCLKE bit is applicable. When the SDCLKE bit is 1 and the PSTOP0 bit is 0, pin P70 becomes the SDCLK output. When the SDCLKE bit is 1 and the PSTOP0 bit is 1, output of the SDCLK signal stops and the SDCLK output is fixed to the high level. Furthermore, when the SDCLKE bit is 0 and the PORT7.DDR.B0 bit is 0, the pin function changes to an input port pin, and when the SDCLKE bit is 0 and the PORT7.DDR.B0 bit is 1, the pin function changes to an output port pin.

Table 9.7 shows the SDCLK pin state in each low power consumption mode.

Table 9.7 SDCLK Pin (P70) State in Each Low Power Consumption Mode

Register Settings			Normal Operating Mode	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode		Deep Software Standby Mode	
SDCLKE	DDR	PSTOP0				OPE = 0	OPE = 1	IOKEEP = 0	IOKEEP = 1
0	0	x	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
	1		PORT output	PORT output	PORT output	PORT output	PORT output	PORT output	PORT output
1	x	0	SDCLK output	SDCLK output	SDCLK output	High	High	High	High
		1	High	High	High	High	High	High	High

9.7 Usage Notes

9.7.1 I/O Port States

I/O port states are retained in software standby mode and deep software standby mode. Therefore, supply current is not reduced while output signals are held high.

9.7.2 Module Stop State of the DMACA, DTC, EXDMAC, and EDMAC

Before setting the MSTPA29 or MSTPA28 bit in MSTPCRA and the MSTPB15 bit in MSTPCRB to 1, clear the DMST bit in EDMAST of the EXDMAC, the DMST bit in DMAST of the DMACA, the DTCST bit in DTCST of the DTC, and the TR bit in EDTRR and the RR bit in EDRRR of the EDMAC to 0 so that initiating transfer by the DTC, DMACA, EXDMAC, or EDMAC is not possible.

For details, see section 14, DMA Controller (DMACA), section 15, EXDMA Controller (EXDMAC), section 16, Data Transfer Controller (DTCa), and section 27, Ethernet Controller Direct Memory Access Controller (EDMAC).

9.7.3 On-Chip Peripheral Module Interrupts

Operation of relevant interrupt is disabled in the module stop state. Therefore, if the module stop state is made with an interrupt request pending, a CPU interrupt source or a DMACA, DTC, EXDMAC, or EDMAC startup source cannot be cleared. For this reason, disable interrupts before entering the module stop state.

9.7.4 Write-Access to MSTPCRA, MSTPCRB, and MSTPCRC

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

9.7.5 Input Buffer Control by DIRQnE Bit (n = 3 to 0)

When the input buffers for the P30/IRQ0-A to P33/IRQ3-A pins are enabled by setting the DIRQnE bit (n = 3 to 0) in DPSIER to 1. Therefore, note that, although inputs to these pins are sent to the DIRQnF (n = 3 to 0) bit in DPSIER, they are not sent to the interrupt controller, peripheral modules, and I/O ports. Use PORTn.ICR for inputs to the interrupt controller, peripheral modules, and I/O ports.

9.7.6 Conflict between Transition to Deep Software Standby Mode and Interrupt

If a conflict occurs between a transition to deep software standby mode and generation of a software standby mode canceling source, the transition to deep software standby mode is not realized but the software standby mode canceling sequence is started. After the oscillation settling time specified by the STS[4:0] bits in SBYCR for software standby mode has passed, the interrupt exception handling is started.

Note that if a conflict occurs between a transition to deep software standby mode and generation of a software standby mode canceling source, the interrupt exception handling routine is required.

9.7.7 Timing of Wait Instructions

The WAIT instruction is executed before completion of the preceding register write; it may be executed before the register modification is reflected, causing unintended operation. To avoid this, always execute the WAIT instruction after confirming that the last write to the register has been completed.

10. Exceptions

10.1 Types of Exceptions

During the execution of a program by the CPU, the occurrence of certain events may necessitate suspending execution of the main flow of the program and starting the execution of another flow. Such events are called exceptions.

The RX CPU supports the seven types of exceptions listed in table 10.1.

The occurrence of an exception causes the processor mode to switch to supervisor mode.

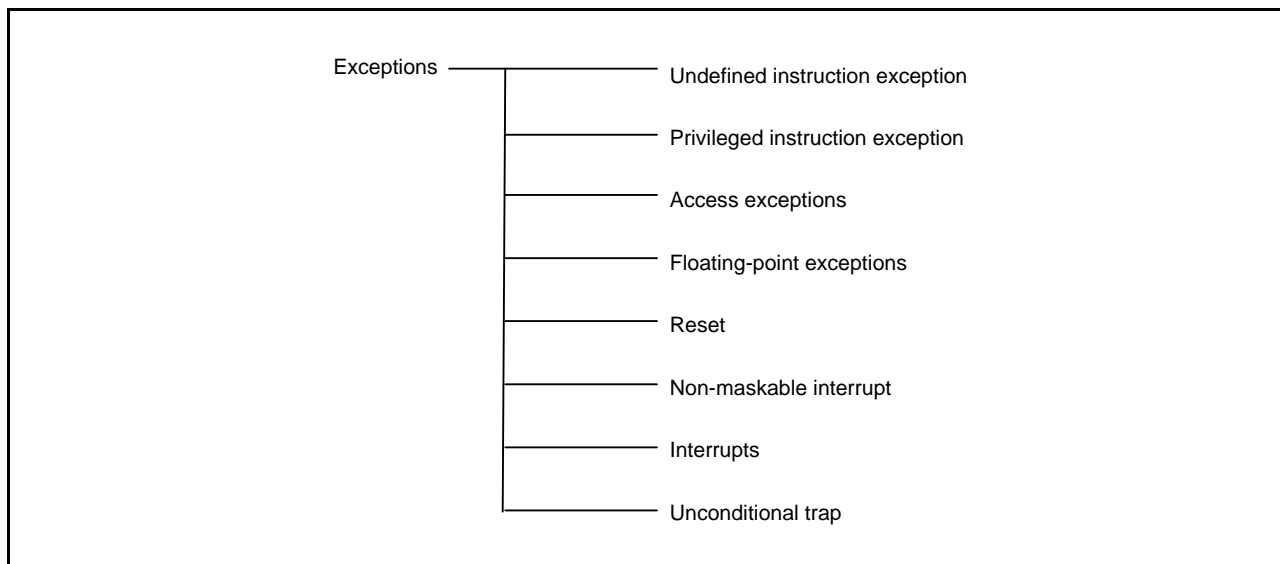


Figure 10.1 Types of Exception

10.1.1 Undefined Instruction Exception

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

10.1.2 Privileged Instruction Exception

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

10.1.3 Access Exceptions

An access exception is generated when access to memory by the CPU leads to an error. An instruction-access exception is generated when the memory-protection unit detects an instruction memory-protection error and an operand-access exception is generated when it detects a data memory-protection error.

10.1.4 Floating-Point Exceptions

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

10.1.5 Reset

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

10.1.6 Non-Maskable Interrupt

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

10.1.7 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 fifteen (the highest). The exception handling of interrupts is masked when the I bit in PSW is 0.

10.1.8 Unconditional Trap

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

10.2 Exception Handling Procedure

For exception handling, part of the processing is handled automatically by hardware and part is handled by a program (the exception handling routine) that has been written by the user. figure 10.2 shows the handling procedure when an exception other than a reset is accepted.

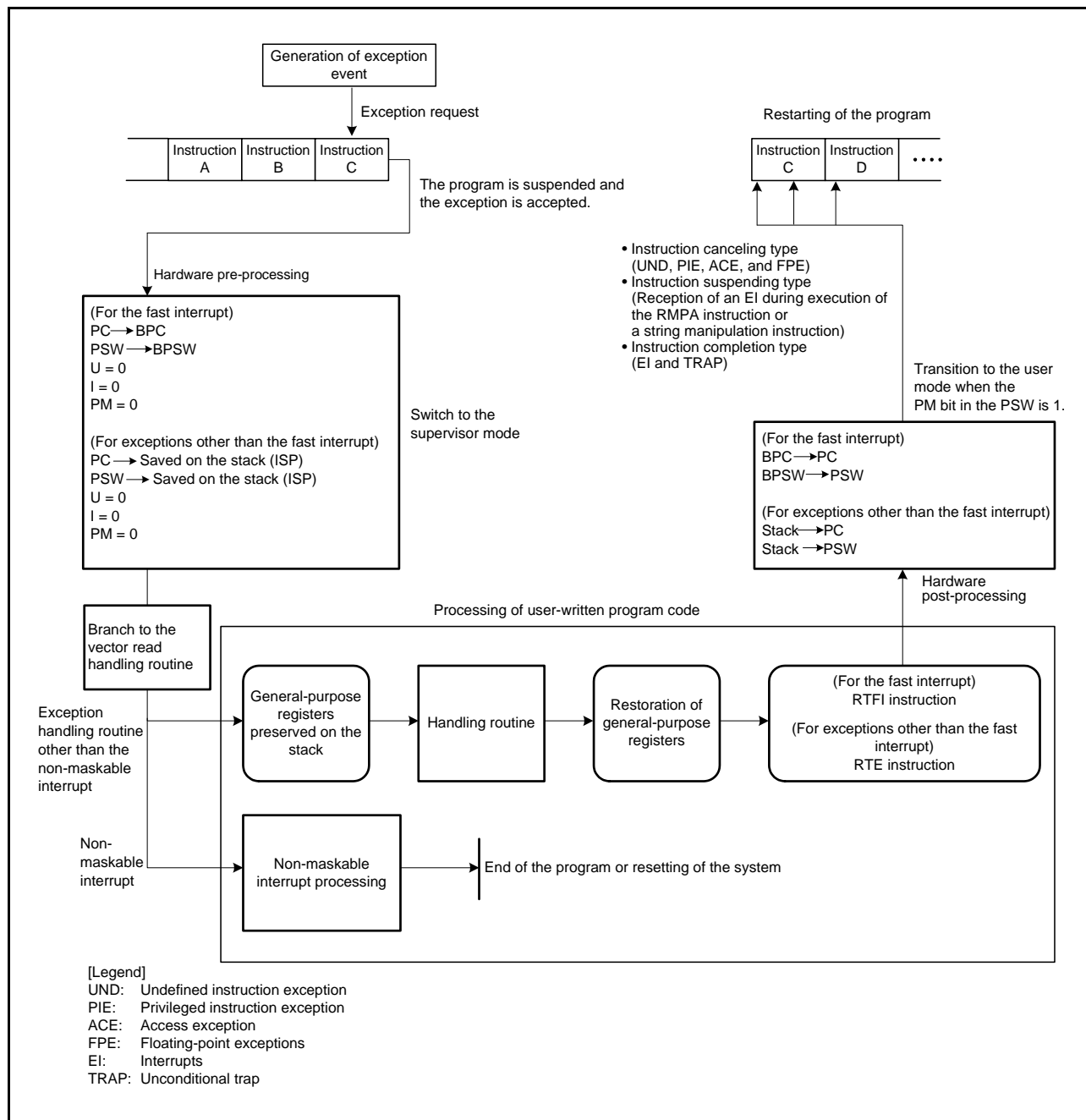


Figure 10.2 Outline of the Exception Handling Procedure

When an exception is accepted, hardware processing by the RX CPU is followed by vector access to acquire the address of the branch destination. A vector address is allocated to each exception. The branch destination address of the exception handling routine for the given exception is written to each vector address. The combination is referred to as a vector.

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 the fast interrupt, the contents are saved in the backup program counter (BPC) and the backup processor status word (BPSW), respectively. In the case of other exceptions, the contents are preserved 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 preserved on the stack by user program code at the start of the exception handling routine.

On completion of processing by most exception processing handlers, registers preserved under program control are restored and the RTE instruction is executed to restore execution from the exception handling routine to the original program. For return from the fast interrupt, the RTFI instruction is used instead. In the case of the non-maskable interrupt, however, end the program or reset the system without returning to the original program.

Hardware post-processing by the RX CPU handles restoration of the pre-exception contents of the PC and PSW. In the case of the fast interrupt, the contents of the BPC and BPSW are restored to the PC and PSW, respectively. In the case of other exceptions, the contents are restored from the stack area to the PC and PSW.

10.3 Acceptance of Exceptions

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

10.3.1 Timing of Acceptance and Saved PC Values

Table 10.1 lists the timing of acceptance and program counter (PC) value to be saved for each type of exception event.

Table 10.1 Timing of Acceptance and Saved PC Value

Exception	Type of Handling	Timing of Acceptance	Value Saved in the BPC/ on the Stack	
Undefined instruction exception	Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception	
Privileged instruction exception	Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception	
Floating-point exceptions	Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception	
Access exceptions	Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception	
Reset	Program abandonment type	At each 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 the above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Interrupts	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 the 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	

10.3.2 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 10.2.

Table 10.2 Vector and Site for Saving the Values in the PC and PSW

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

10.4 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 the values in the PSW

- For the fast interrupt
PSW → BPSW
- For other exceptions
PSW → Stack area

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

(b) Updating of the PM, U, and I bits in the PSW

I: Cleared
I: Cleared
PM: Cleared

(c) Saving the values in the PC

- For the fast interrupt
PC → BPC
- For other exceptions
PC register → Stack area

(d) Set the branch-destination address of the exception handling routine in the PC register

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

(2) Hardware post-processing for execution of RTE and RTFI instructions

(a) Restoring the values in the PSW

- For the fast interrupt
BPSW → PSW
- For other exceptions
Stack area → PSW

(b) Restoring the values in the PC

- For the fast interrupt
BPC → PC
- For other exceptions
Stack area → PC

10.5 Hardware Pre-Processing

The sequences of Hardware Pre-Processing from reception of each exception request to execution of the associated exception processing routine are explained below.

10.5.1 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 the PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The address of the processing routine is fetched from the vector address, FFFFFFFDCh.
5. The PC is set to the fetched address and processing branches to the start of the exception handling routine.

10.5.2 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 the PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The address of the processing routine is fetched from the vector address, FFFFFFFD0h.
5. The PC is set to the fetched address and processing branches to the start of the exception handling routine.

10.5.3 Floating-Point Exceptions

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in the PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The address of the processing routine is fetched from the vector address, FFFFFFFE4h.
5. The PC is set to the fetched address and processing branches to the start of the exception handling routine.

10.5.4 Access Exceptions

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in the PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The address of the processing routine is fetched from the vector address, FFFFFFFD4h.
5. The PC is set to the fetched address and processing branches to the start of the exception handling routine.

10.5.5 Reset

1. The control registers are initialized.
2. The address of the processing routine is fetched from the vector address, FFFFFFFFCh.
3. The PC is set to the fetched address.

10.5.6 Non-Maskable Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in the PSW are cleared to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved on the stack (ISP). For other instructions, the PC value of the next instruction is saved.

4. The processor interrupt priority level bits (IPL[3:0]) in the PSW are set to "Fh".
5. The address of the processing routine is fetched from the vector address, FFFFFFFF8h.
6. The PC is set to the fetched address and the processing branches to the start of the exception handling routine.

10.5.7 Interrupts

1. The value of the processor status word (PSW) is saved on the stack (ISP) or, for the fast interrupt, in the backup processor status word (BPSW).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in the PSW are cleared to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved. For other instructions, the PC value of the next instruction is saved. Saving of the PC is in the backup program counter (BPC) for fast interrupts.
4. The processor interrupt priority level bits (IPL[3:0]) in the PSW indicate the interrupt priority level of the interrupt.
5. The address of the processing routine 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 PC is set to the fetched address and processing branches to the start of the exception handling routine.

10.5.8 Unconditional Trap

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in the PSW register are cleared to 0.
3. The value of the program counter (PC) for the next instruction is saved on the stack (ISP).
4. For the INT instruction, the value at the vector corresponding to the INT instruction number is fetched from the relocatable vector table.
For the BRK instruction, the value at the vector from the start address is fetched from the relocatable vector table.
5. The PC is set to the fetched address and the processing branches to the start of the exception handling routine.

10.6 Return from Exception Processing Routines

Executing the instructions listed in table 10.3 at the end of the corresponding exception processing routines 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 processing sequence.


Table 10.3 Return from Exception Processing Routines

Exception	Instruction for Return	
Undefined instruction exception	RTE	
Privileged instruction exception	RTE	
Access exceptions	RTE	
Floating-point exceptions	RTE	
Reset	Return is impossible	
Non-maskable interrupt	Return is impossible	
Interrupts	Fast interrupt	RTFI
	Other than the above	RTE
Unconditional trap	RTE	

10.7 Order of Priority for Exceptions

The order of priority for exceptions is given in table . When multiple exceptions are generated at the same time, the exception with the highest priority is accepted first.

Table 10.4 Priority of Exception Events

Order of Priority	Exception
High  Low	1 Reset
	2 Non-maskable interrupt
	3 Interrupts
	4 INstruction access exceptions
	5 Undefined instruction exception Privileged instruction exception
	6 Unconditional trap
	7 Operand access exception
	8 Floating-point exceptions

11. Interrupt Control Unit (ICUa)

11.1 Overview

The interrupt control unit (ICU) receives interrupt signals from the peripheral modules and external pins, sends interrupts to the CPU, and activates the DTC and DMACA.

Table 11.1 lists the specifications of the interrupt control unit, and Figure 11.1 shows a block diagram of the interrupt control unit.

Table 11.1 Specifications of the Interrupt Control Unit

Item	Description	
Interrupts	Peripheral function interrupts <ul style="list-style-type: none"> • Interrupts from peripheral modules • Number of sources: 146 • Interrupt detection: Edge detection/level detection • Edge detection or level detection is determined for each source of connected peripheral modules. 	
	External pin interrupts <ul style="list-style-type: none"> • Interrupts from pins IRQ15 to IRQ0 • Number of sources: 16 • Interrupt detection: Low level/falling edge/rising edge/rising and falling edges • One of these detection methods can be set for each source. 	
	Software interrupt <ul style="list-style-type: none"> • Interrupt generated by writing to a register • One interrupt source 	
	Interrupt priority	Specified by registers.
	Fast interrupt function	Faster interrupt processing of the CPU can be set only for a single interrupt source.
	DTC/DMACA control	<ul style="list-style-type: none"> • The DTC and DMACA can be activated by interrupt sources. • Number of DTC activating sources: 102 (85 peripheral function interrupts + 16 external pin interrupts + 1 software interrupt) • Number of DMACA activating sources: 45 (41 peripheral function interrupts + 4 external pin interrupts)
Non-maskable interrupts	NMI pin interrupts <ul style="list-style-type: none"> • Interrupt from the NMI pin • Interrupt detection: Falling edge/rising edge 	
	Voltage monitoring interrupt	Interrupt during power-voltage fall detection
	Oscillation stop detection interrupt	Interrupt during oscillation stop detection
Return from power-down modes	<ul style="list-style-type: none"> • Sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source. • All-module clock stop mode: Return is initiated by non-maskable interrupts, IRQ15 to IRQ0 interrupts, WDT interrupts, TMR interrupts, USB interrupts (USBR), or RTC alarm interrupts. • Software standby mode: Return is initiated by non-maskable interrupts, IRQ15 to IRQ0 interrupts, USB interrupts (USBR), or RTC alarm interrupts. 	

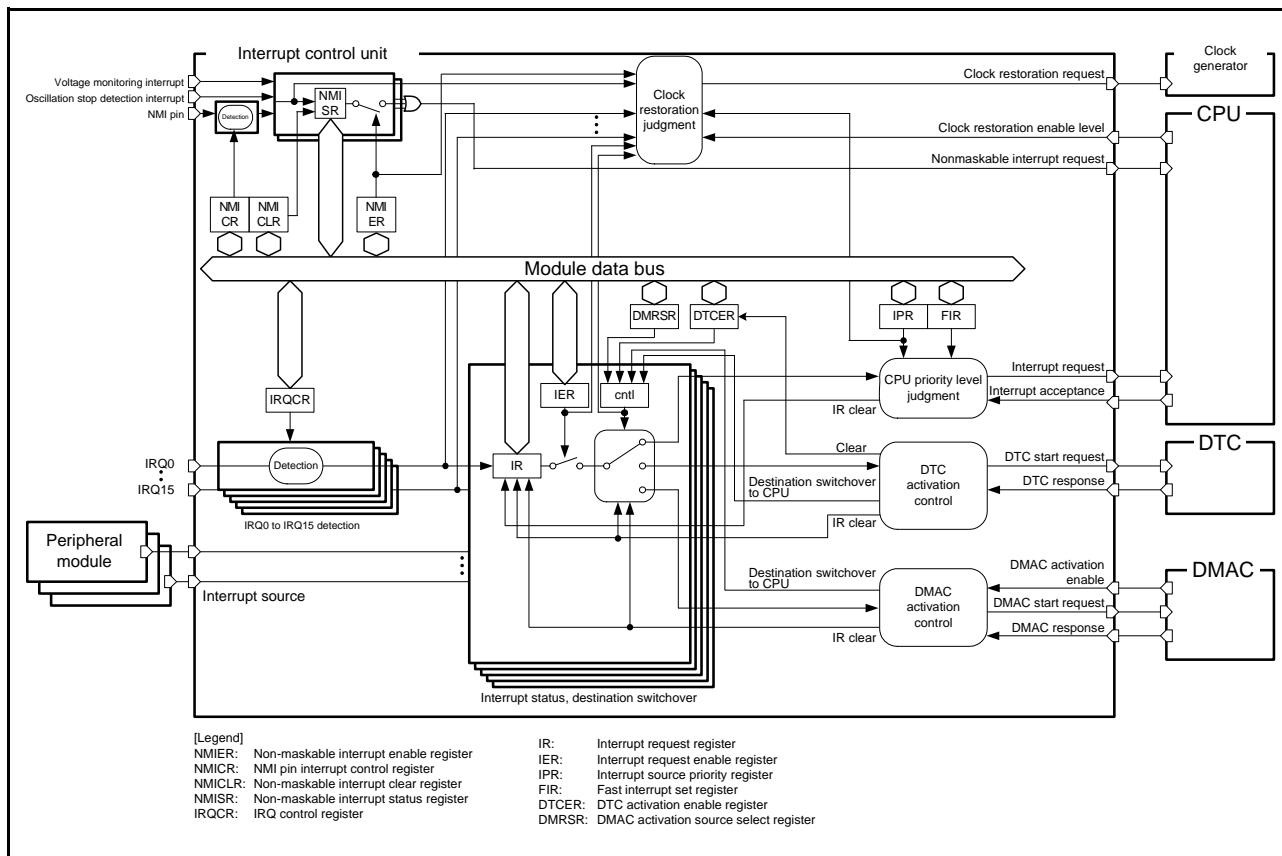


Figure 11.1 Block Diagram of Interrupt Control Unit

Table 11.2 shows the input/output pins of the interrupt control unit.

Table 11.2 Pin Configuration of ICU

Pin Name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ15 to IRQ0	Input	External interrupt request pins

11.2 Register Descriptions

Table 11.3 lists the registers of the interrupt control unit.

Table 11.3 Registers of the Interrupt Control Unit (1 / 10)

Register Name	Symbol	Value after Reset	Address	Access Size
Interrupt request register 016	IR016	00h	0008 7010h	8
Interrupt request register 021	IR021	00h	0008 7015h	8
Interrupt request register 023	IR023	00h	0008 7017h	8
Interrupt request register 027	IR027	00h	0008 701Bh	8
Interrupt request register 028	IR028	00h	0008 701Ch	8
Interrupt request register 029	IR029	00h	0008 701Dh	8
Interrupt request register 030	IR030	00h	0008 701Eh	8
Interrupt request register 031	IR031	00h	0008 701Fh	8
Interrupt request register 032	IR032	00h	0008 7020h	8
Interrupt request register 036	IR036	00h	0008 7024h	8
Interrupt request register 037	IR037	00h	0008 7025h	8
Interrupt request register 038	IR038	00h	0008 7026h	8
Interrupt request register 040	IR040	00h	0008 7028h	8
Interrupt request register 041	IR041	00h	0008 7029h	8
Interrupt request register 042	IR042	00h	0008 702Ah	8
Interrupt request register 044	IR044	00h	0008 702Ch	8
Interrupt request register 045	IR045	00h	0008 702Dh	8
Interrupt request register 046	IR046	00h	0008 702Eh	8
Interrupt request register 047	IR047	00h	0008 702Fh	8
Interrupt request register 048	IR048	00h	0008 7030h	8
Interrupt request register 049	IR049	00h	0008 7031h	8
Interrupt request register 050	IR050	00h	0008 7032h	8
Interrupt request register 051	IR051	00h	0008 7 033h	8
Interrupt request register 056	IR056	00h	0008 7 038h	8
Interrupt request register 057	IR057	00h	0008 7039h	8
Interrupt request register 058	IR058	00h	0008 703Ah	8
Interrupt request register 059	IR059	00h	0008 703Bh	8
Interrupt request register 060	IR060	00h	0008 703Ch	8
Interrupt request register 062	IR062	00h	0008 703Eh	8
Interrupt request register 063	IR063	00h	0008 703Fh	8
Interrupt request register 064	IR064	00h	0008 7040h	8
Interrupt request register 065	IR065	00h	0008 7041h	8
Interrupt request register 066	IR066	00h	0008 7042h	8
Interrupt request register 067	IR067	00h	0008 7043h	8
Interrupt request register 068	IR068	00h	0008 7044h	8
Interrupt request register 069	IR069	00h	0008 7045h	8
Interrupt request register 070	IR070	00h	0008 7046h	8
Interrupt request register 071	IR071	00h	0008 7047h	8
Interrupt request register 072	IR072	00h	0008 7048h	8
Interrupt request register 073	IR073	00h	0008 7049h	8
Interrupt request register 074	IR074	00h	0008 704Ah	8
Interrupt request register 075	IR075	00h	0008 704Bh	8

Table 11.3 Registers of the Interrupt Control Unit (2 / 10)

Register Name	Symbol	Value after Reset	Address	Access Size
Interrupt request register 076	IR076	00h	0008 704Ch	8
Interrupt request register 077	IR077	00h	0008 704Dh	8
Interrupt request register 078	IR078	00h	0008 704Eh	8
Interrupt request register 079	IR079	00h	0008 704Fh	8
Interrupt request register 090	IR090	00h	0008 705Ah	8
Interrupt request register 091	IR091	00h	0008 705Bh	8
Interrupt request register 092	IR092	00h	0008 705Ch	8
Interrupt request register 096	IR096	00h	0008 7060h	8
Interrupt request register 098	IR098	00h	0008 7062h	8
Interrupt request register 099	IR099	00h	0008 7063h	8
Interrupt request register 102	IR102	00h	0008 7066h	8
Interrupt request register 114	IR114	00h	0008 7072h	8
Interrupt request register 115	IR115	00h	0008 7073h	8
Interrupt request register 116	IR116	00h	0008 7074h	8
Interrupt request register 117	IR117	00h	0008 7075h	8
Interrupt request register 118	IR118	00h	0008 7076h	8
Interrupt request register 119	IR119	00h	0008 7077h	8
Interrupt request register 120	IR120	00h	0008 7078h	8
Interrupt request register 121	IR121	00h	0008 7079h	8
Interrupt request register 122	IR122	00h	0008 707Ah	8
Interrupt request register 123	IR123	00h	0008 707Bh	8
Interrupt request register 124	IR124	00h	0008 707Ch	8
Interrupt request register 125	IR125	00h	0008 707Dh	8
Interrupt request register 126	IR126	00h	0008 707Eh	8
Interrupt request register 127	IR127	00h	0008 707Fh	8
Interrupt request register 128	IR128	00h	0008 7080h	8
Interrupt request register 129	IR129	00h	0008 7081h	8
Interrupt request register 130	IR130	00h	0008 7082h	8
Interrupt request register 131	IR131	00h	0008 7083h	8
Interrupt request register 132	IR132	00h	0008 7084h	8
Interrupt request register 133	IR133	00h	0008 7085h	8
Interrupt request register 134	IR134	00h	0008 7086h	8
Interrupt request register 135	IR135	00h	0008 7087h	8
Interrupt request register 136	IR136	00h	0008 7088h	8
Interrupt request register 137	IR137	00h	0008 7089h	8
Interrupt request register 138	IR138	00h	0008 708Ah	8
Interrupt request register 139	IR139	00h	0008 708Bh	8
Interrupt request register 140	IR140	00h	0008 708Ch	8
Interrupt request register 141	IR141	00h	0008 708Dh	8
Interrupt request register 142	IR142	00h	0008 708Eh	8
Interrupt request register 143	IR143	00h	0008 708Fh	8
Interrupt request register 144	IR144	00h	0008 7090h	8
Interrupt request register 145	IR145	00h	0008 7091h	8
Interrupt request register 146	IR146	00h	0008 7092h	8
Interrupt request register 147	IR147	00h	0008 7093h	8

Table 11.3 Registers of the Interrupt Control Unit (3 / 10)

Register Name	Symbol	Value after Reset	Address	Access Size
Interrupt request register 148	IR148	00h	0008 7094h	8
Interrupt request register 149	IR149	00h	0008 7095h	8
Interrupt request register 150	IR150	00h	0008 7096h	8
Interrupt request register 151	IR151	00h	0008 7097h	8
Interrupt request register 152	IR152	00h	0008 7098h	8
Interrupt request register 153	IR153	00h	0008 7099h	8
Interrupt request register 154	IR154	00h	0008 709Ah	8
Interrupt request register 155	IR155	00h	0008 709Bh	8
Interrupt request register 156	IR156	00h	0008 709Ch	8
Interrupt request register 157	IR157	00h	0008 709Dh	8
Interrupt request register 158	IR158	00h	0008 709Eh	8
Interrupt request register 159	IR159	00h	0008 709Fh	8
Interrupt request register 160	IR160	00h	0008 70A0h	8
Interrupt request register 161	IR161	00h	0008 70A1h	8
Interrupt request register 162	IR162	00h	0008 70A2h	8
Interrupt request register 163	IR163	00h	0008 70A3h	8
Interrupt request register 164	IR164	00h	0008 70A4h	8
Interrupt request register 165	IR165	00h	0008 70A5h	8
Interrupt request register 166	IR166	00h	0008 70A6h	8
Interrupt request register 167	IR167	00h	0008 70A7h	8
Interrupt request register 168	IR168	00h	0008 70A8h	8
Interrupt request register 169	IR169	00h	0008 70A9h	8
Interrupt request register 170	IR170	00h	0008 70AAh	8
Interrupt request register 171	IR171	00h	0008 70ABh	8
Interrupt request register 172	IR172	00h	0008 70ACh	8
Interrupt request register 173	IR173	00h	0008 70ADh	8
Interrupt request register 174	IR174	00h	0008 70AEh	8
Interrupt request register 175	IR175	00h	0008 70AFh	8
Interrupt request register 176	IR176	00h	0008 70B0h	8
Interrupt request register 177	IR177	00h	0008 70B1h	8
Interrupt request register 178	IR178	00h	0008 70B2h	8
Interrupt request register 179	IR179	00h	0008 70B3h	8
Interrupt request register 180	IR180	00h	0008 70B4h	8
Interrupt request register 181	IR181	00h	0008 70B5h	8
Interrupt request register 182	IR182	00h	0008 70B6h	8
Interrupt request register 183	IR183	00h	0008 70B7h	8
Interrupt request register 184	IR184	00h	0008 70B8h	8
Interrupt request register 185	IR185	00h	0008 70B9h	8
Interrupt request register 198	IR198	00h	0008 70C6h	8
Interrupt request register 199	IR199	00h	0008 70C7h	8
Interrupt request register 200	IR200	00h	0008 70C8h	8
Interrupt request register 201	IR201	00h	0008 70C9h	8
Interrupt request register 202	IR202	00h	0008 70CAh	8
Interrupt request register 203	IR203	00h	0008 70CBh	8
Interrupt request register 214	IR214	00h	0008 70D6h	8

Table 11.3 Registers of the Interrupt Control Unit (4 / 10)

Register Name	Symbol	Value after Reset	Address	Access Size
Interrupt request register 215	IR215	00h	0008 70D7h	8
Interrupt request register 216	IR216	00h	0008 70D8h	8
Interrupt request register 217	IR217	00h	0008 70D9h	8
Interrupt request register 218	IR218	00h	0008 70DAh	8
Interrupt request register 219	IR219	00h	0008 70DBh	8
Interrupt request register 220	IR220	00h	0008 70DCh	8
Interrupt request register 221	IR221	00h	0008 70DDh	8
Interrupt request register 222	IR222	00h	0008 70DEh	8
Interrupt request register 223	IR223	00h	0008 70DFh	8
Interrupt request register 224	IR224	00h	0008 70E0h	8
Interrupt request register 225	IR225	00h	0008 70E1h	8
Interrupt request register 226	IR226	00h	0008 70E2h	8
Interrupt request register 227	IR227	00h	0008 70E3h	8
Interrupt request register 228	IR228	00h	0008 70E4h	8
Interrupt request register 229	IR229	00h	0008 70E5h	8
Interrupt request register 234	IR234	00h	0008 70EAh	8
Interrupt request register 235	IR235	00h	0008 70EBh	8
Interrupt request register 236	IR236	00h	0008 70ECh	8
Interrupt request register 237	IR237	00h	0008 70EDh	8
Interrupt request register 238	IR238	00h	0008 70EEh	8
Interrupt request register 239	IR239	00h	0008 70EFh	8
Interrupt request register 240	IR240	00h	0008 70F0h	8
Interrupt request register 241	IR241	00h	0008 70F1h	8
Interrupt request register 246	IR246	00h	0008 70F6h	8
Interrupt request register 247	IR247	00h	0008 70F7h	8
Interrupt request register 248	IR248	00h	0008 70F8h	8
Interrupt request register 249	IR249	00h	0008 70F9h	8
Interrupt request register 250	IR250	00h	0008 70FAh	8
Interrupt request register 251	IR251	00h	0008 70FBh	8
Interrupt request register 252	IR252	00h	0008 70FCh	8
Interrupt request register 253	IR253	00h	0008 70FDh	8
DTC activation enable register 027	DTCER027	00h	0008 711Bh	8
DTC activation enable register 028	DTCER028	00h	0008 711Ch	8
DTC activation enable register 029	DTCER029	00h	0008 711Dh	8
DTC activation enable register 030	DTCER030	00h	0008 711Eh	8
DTC activation enable register 031	DTCER031	00h	0008 711Fh	8
DTC activation enable register 036	DTCER036	00h	0008 7124h	8
DTC activation enable register 037	DTCER037	00h	0008 7125h	8
DTC activation enable register 040	DTCER040	00h	0008 7128h	8
DTC activation enable register 041	DTCER041	00h	0008 7129h	8
DTC activation enable register 045	DTCER045	00h	0008 712Dh	8
DTC activation enable register 046	DTCER046	00h	0008 712Eh	8
DTC activation enable register 049	DTCER049	00h	0008 7131h	8
DTC activation enable register 050	DTCER050	00h	0008 7132h	8
DTC activation enable register 064	DTCER064	00h	0008 7140h	8

Table 11.3 Registers of the Interrupt Control Unit (5 / 10)

Register Name	Symbol	Value after Reset	Address	Access Size
DTC activation enable register 065	DTCER065	00h	0008 7141h	8
DTC activation enable register 066	DTCER066	00h	0008 7142h	8
DTC activation enable register 067	DTCER067	00h	0008 7143h	8
DTC activation enable register 068	DTCER068	00h	0008 7144h	8
DTC activation enable register 069	DTCER069	00h	0008 7145h	8
DTC activation enable register 070	DTCER070	00h	0008 7146h	8
DTC activation enable register 071	DTCER071	00h	0008 7147h	8
DTC activation enable register 072	DTCER072	00h	0008 7148h	8
DTC activation enable register 073	DTCER073	00h	0008 7149h	8
DTC activation enable register 074	DTCER074	00h	0008 714Ah	8
DTC activation enable register 075	DTCER075	00h	0008 714Bh	8
DTC activation enable register 076	DTCER076	00h	0008 714Ch	8
DTC activation enable register 077	DTCER077	00h	0008 714Dh	8
DTC activation enable register 078	DTCER078	00h	0008 714Eh	8
DTC activation enable register 079	DTCER079	00h	0008 714Fh	8
DTC activation enable register 098	DTCER098	00h	0008 7162h	8
DTC activation enable register 099	DTCER099	00h	0008 7163h	8
DTC activation enable register 102	DTCER102	00h	0008 7166h	8
DTC activation enable register 114	DTCER114	00h	0008 7172h	8
DTC activation enable register 115	DTCER115	00h	0008 7173h	8
DTC activation enable register 116	DTCER116	00h	0008 7174h	8
DTC activation enable register 117	DTCER117	00h	0008 7175h	8
DTC activation enable register 121	DTCER121	00h	0008 7179h	8
DTC activation enable register 122	DTCER122	00h	0008 717Ah	8
DTC activation enable register 125	DTCER125	00h	0008 717Dh	8
DTC activation enable register 126	DTCER126	00h	0008 717Eh	8
DTC activation enable register 129	DTCER129	00h	0008 7181h	8
DTC activation enable register 130	DTCER130	00h	0008 7182h	8
DTC activation enable register 131	DTCER131	00h	0008 7183h	8
DTC activation enable register 132	DTCER132	00h	0008 7184h	8
DTC activation enable register 134	DTCER134	00h	0008 7186h	8
DTC activation enable register 135	DTCER135	00h	0008 7187h	8
DTC activation enable register 136	DTCER136	00h	0008 7188h	8
DTC activation enable register 137	DTCER137	00h	0008 7189h	8
DTC activation enable register 138	DTCER138	00h	0008 718Ah	8
DTC activation enable register 139	DTCER139	00h	0008 718Bh	8
DTC activation enable register 140	DTCER140	00h	0008 718Ch	8
DTC activation enable register 141	DTCER141	00h	0008 718Dh	8
DTC activation enable register 142	DTCER142	00h	0008 718Eh	8
DTC activation enable register 143	DTCER143	00h	0008 718Fh	8
DTC activation enable register 144	DTCER144	00h	0008 7190h	8
DTC activation enable register 145	DTCER145	00h	0008 7191h	8
DTC activation enable register 149	DTCER149	00h	0008 7195h	8
DTC activation enable register 150	DTCER150	00h	0008 7196h	8
DTC activation enable register 153	DTCER153	00h	0008 7199h	8

Table 11.3 Registers of the Interrupt Control Unit (6 / 10)

Register Name	Symbol	Value after Reset	Address	Access Size
DTC activation enable register 154	DTCER154	00h	0008 719Ah	8
DTC activation enable register 157	DTCER157	00h	0008 719Dh	8
DTC activation enable register 158	DTCER158	00h	0008 719Eh	8
DTC activation enable register 159	DTCER159	00h	0008 719Fh	8
DTC activation enable register 160	DTCER160	00h	0008 71A0h	8
DTC activation enable register 162	DTCER162	00h	0008 71A2h	8
DTC activation enable register 163	DTCER163	00h	0008 71A3h	8
DTC activation enable register 164	DTCER164	00h	0008 71A4h	8
DTC activation enable register 165	DTCER165	00h	0008 71A5h	8
DTC activation enable register 166	DTCER166	00h	0008 71A6h	8
DTC activation enable register 167	DTCER167	00h	0008 71A7h	8
DTC activation enable register 168	DTCER168	00h	0008 71A8h	8
DTC activation enable register 169	DTCER169	00h	0008 71A9h	8
DTC activation enable register 174	DTCER174	00h	0008 71AEh	8
DTC activation enable register 175	DTCER175	00h	0008 71AFh	8
DTC activation enable register 177	DTCER177	00h	0008 71B1h	8
DTC activation enable register 178	DTCER178	00h	0008 71B2h	8
DTC activation enable register 180	DTCER180	00h	0008 71B4h	8
DTC activation enable register 181	DTCER181	00h	0008 71B5h	8
DTC activation enable register 183	DTCER183	00h	0008 71B7h	8
DTC activation enable register 184	DTCER184	00h	0008 71B8h	8
DTC activation enable register 198	DTCER198	00h	0008 71C6h	8
DTC activation enable register 199	DTCER199	00h	0008 71C7h	8
DTC activation enable register 200	DTCER200	00h	0008 71C8h	8
DTC activation enable register 201	DTCER201	00h	0008 71C9h	8
DTC activation enable register 202	DTCER202	00h	0008 71CAh	8
DTC activation enable register 203	DTCER203	00h	0008 71CBh	8
DTC activation enable register 215	DTCER215	00h	0008 71D7h	8
DTC activation enable register 216	DTCER216	00h	0008 71D8h	8
DTC activation enable register 219	DTCER219	00h	0008 71DBh	8
DTC activation enable register 220	DTCER220	00h	0008 71DCh	8
DTC activation enable register 223	DTCER223	00h	0008 71DFh	8
DTC activation enable register 224	DTCER224	00h	0008 71E0h	8
DTC activation enable register 227	DTCER227	00h	0008 71E3h	8
DTC activation enable register 228	DTCER228	00h	0008 71E4h	8
DTC activation enable register 235	DTCER235	00h	0008 71EBh	8
DTC activation enable register 236	DTCER236	00h	0008 71ECh	8
DTC activation enable register 239	DTCER239	00h	0008 71EFh	8
DTC activation enable register 240	DTCER240	00h	0008 71F0h	8
DTC activation enable register 247	DTCER247	00h	0008 71F7h	8
DTC activation enable register 248	DTCER248	00h	0008 71F8h	8
DTC activation enable register 251	DTCER251	00h	0008 71FBh	8
DTC activation enable register 252	DTCER252	00h	0008 71FCh	8
Interrupt request enable register 02	IER02	00h	0008 7202h	8
Interrupt request enable register 03	IER03	00h	0008 7203h	8

Table 11.3 Registers of the Interrupt Control Unit (7 / 10)

Register Name	Symbol	Value after Reset	Address	Access Size
Interrupt request enable register 04	IER04	00h	0008 7204h	8
Interrupt request enable register 05	IER05	00h	0008 7205h	8
Interrupt request enable register 06	IER06	00h	0008 7206h	8
Interrupt request enable register 07	IER07	00h	0008 7207h	8
Interrupt request enable register 08	IER08	00h	0008 7208h	8
Interrupt request enable register 09	IER09	00h	0008 7209h	8
Interrupt request enable register 0B	IER0B	00h	0008 720Bh	8
Interrupt request enable register 0C	IER0C	00h	0008 720Ch	8
Interrupt request enable register 0E	IER0E	00h	0008 720Eh	8
Interrupt request enable register 0F	IER0F	00h	0008 720Fh	8
Interrupt request enable register 10	IER10	00h	0008 7210h	8
Interrupt request enable register 11	IER11	00h	0008 7211h	8
Interrupt request enable register 12	IER12	00h	0008 7212h	8
Interrupt request enable register 13	IER13	00h	0008 7213h	8
Interrupt request enable register 14	IER14	00h	0008 7214h	8
Interrupt request enable register 15	IER15	00h	0008 7215h	8
Interrupt request enable register 16	IER16	00h	0008 7216h	8
Interrupt request enable register 17	IER17	00h	0008 7217h	8
Interrupt request enable register 18	IER18	00h	0008 7218h	8
Interrupt request enable register 19	IER19	00h	0008 7219h	8
Interrupt request enable register 1A	IER1A	00h	0008 721Ah	8
Interrupt request enable register 1B	IER1B	00h	0008 721Bh	8
Interrupt request enable register 1C	IER1C	00h	0008 721Ch	8
Interrupt request enable register 1D	IER1D	00h	0008 721Dh	8
Interrupt request enable register 1E	IER1E	00h	0008 721Eh	8
Interrupt request enable register 1F	IER1F	00h	0008 721Fh	8
Software interrupt activation register	SWINTR	00h	0008 72E0h	8
Fast interrupt set register	FIR	0000h	0008 72F0h	16
Interrupt source priority register 00	IPR00	00h	0008 7300h	8
Interrupt source priority register 01	IPR01	00h	0008 7301h	8
Interrupt source priority register 02	IPR02	00h	0008 7302h	8
Interrupt source priority register 03	IPR03	00h	0008 7303h	8
Interrupt source priority register 04	IPR04	00h	0008 7304h	8
Interrupt source priority register 05	IPR05	00h	0008 7305h	8
Interrupt source priority register 06	IPR06	00h	0008 7306h	8
Interrupt source priority register 07	IPR07	00h	0008 7307h	8
Interrupt source priority register 08	IPR08	00h	0008 7308h	8
Interrupt source priority register 0C	IPR0C	00h	0008 730Ch	8
Interrupt source priority register 0D	IPR0D	00h	0008 730Dh	8
Interrupt source priority register 0E	IPR0E	00h	0008 730Eh	8
Interrupt source priority register 10	IPR10	00h	0008 7310h	8
Interrupt source priority register 11	IPR11	00h	0008 7311h	8
Interrupt source priority register 12	IPR12	00h	0008 7312h	8
Interrupt source priority register 14	IPR14	00h	0008 7314h	8
Interrupt source priority register 15	IPR15	00h	0008 7315h	8

Table 11.3 Registers of the Interrupt Control Unit (8 / 10)

Register Name	Symbol	Value after Reset	Address	Access Size
Interrupt source priority register 18	IPR18	00h	0008 7318h	8
Interrupt source priority register 1E	IPR1E	00h	0008 731Eh	8
Interrupt source priority register 1F	IPR1F	00h	0008 731Fh	8
Interrupt source priority register 20	IPR20	00h	0008 7320h	8
Interrupt source priority register 21	IPR21	00h	0008 7321h	8
Interrupt source priority register 22	IPR22	00h	0008 7322h	8
Interrupt source priority register 23	IPR23	00h	0008 7323h	8
Interrupt source priority register 24	IPR24	00h	0008 7324h	8
Interrupt source priority register 25	IPR25	00h	0008 7325h	8
Interrupt source priority register 26	IPR26	00h	0008 7326h	8
Interrupt source priority register 27	IPR27	00h	0008 7327h	8
Interrupt source priority register 28	IPR28	00h	0008 7328h	8
Interrupt source priority register 29	IPR29	00h	0008 7329h	8
Interrupt source priority register 2A	IPR2A	00h	0008 732Ah	8
Interrupt source priority register 2B	IPR2B	00h	0008 732Bh	8
Interrupt source priority register 2C	IPR2C	00h	0008 732Ch	8
Interrupt source priority register 2D	IPR2D	00h	0008 732Dh	8
Interrupt source priority register 2E	IPR2E	00h	0008 732Eh	8
Interrupt source priority register 2F	IPR2F	00h	0008 732Fh	8
Interrupt source priority register 3A	IPR3A	00h	0008 733Ah	8
Interrupt source priority register 3B	IPR3B	00h	0008 733Bh	8
Interrupt source priority register 3C	IPR3C	00h	0008 733Ch	8
Interrupt source priority register 40	IPR40	00h	0008 7340h	8
Interrupt source priority register 44	IPR44	00h	0008 7344h	8
Interrupt source priority register 45	IPR45	00h	0008 7345h	8
Interrupt source priority register 48	IPR48	00h	0008 7348h	8
Interrupt source priority register 51	IPR51	00h	0008 7351h	8
Interrupt source priority register 52	IPR52	00h	0008 7352h	8
Interrupt source priority register 53	IPR53	00h	0008 7353h	8
Interrupt source priority register 54	IPR54	00h	0008 7354h	8
Interrupt source priority register 55	IPR55	00h	0008 7355h	8
Interrupt source priority register 56	IPR56	00h	0008 7356h	8
Interrupt source priority register 57	IPR57	00h	0008 7357h	8
Interrupt source priority register 58	IPR58	00h	0008 7358h	8
Interrupt source priority register 59	IPR59	00h	0008 7359h	8
Interrupt source priority register 5A	IPR5A	00h	0008 735Ah	8
Interrupt source priority register 5B	IPR5B	00h	0008 735Bh	8
Interrupt source priority register 5C	IPR5C	00h	0008 735Ch	8
Interrupt source priority register 5D	IPR5D	00h	0008 735Dh	8
Interrupt source priority register 5E	IPR5E	00h	0008 735Eh	8
Interrupt source priority register 5F	IPR5F	00h	0008 735Fh	8
Interrupt source priority register 60	IPR60	00h	0008 7360h	8
Interrupt source priority register 61	IPR61	00h	0008 7361h	8
Interrupt source priority register 62	IPR62	00h	0008 7362h	8
Interrupt source priority register 63	IPR63	00h	0008 7363h	8

Table 11.3 Registers of the Interrupt Control Unit (9 / 10)

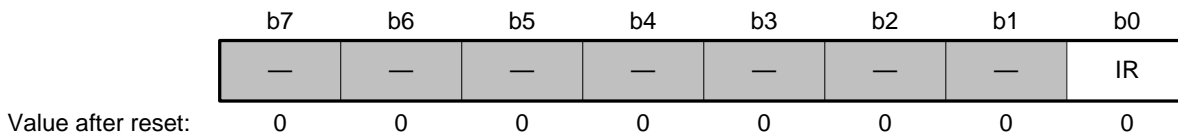
Register Name	Symbol	Value after Reset	Address	Access Size
Interrupt source priority register 64	IPR64	00h	0008 7364h	8
Interrupt source priority register 65	IPR65	00h	0008 7365h	8
Interrupt source priority register 66	IPR66	00h	0008 7366h	8
Interrupt source priority register 67	IPR67	00h	0008 7367h	8
Interrupt source priority register 68	IPR68	00h	0008 7368h	8
Interrupt source priority register 69	IPR69	00h	0008 7369h	8
Interrupt source priority register 6A	IPR6A	00h	0008 736Ah	8
Interrupt source priority register 6B	IPR6B	00h	0008 736Bh	8
Interrupt source priority register 70	IPR70	00h	0008 7370h	8
Interrupt source priority register 71	IPR71	00h	0008 7371h	8
Interrupt source priority register 72	IPR72	00h	0008 7372h	8
Interrupt source priority register 73	IPR73	00h	0008 7373h	8
Interrupt source priority register 74	IPR74	00h	0008 7374h	8
Interrupt source priority register 75	IPR75	00h	0008 7375h	8
Interrupt source priority register 80	IPR80	00h	0008 7380h	8
Interrupt source priority register 81	IPR81	00h	0008 7381h	8
Interrupt source priority register 82	IPR82	00h	0008 7382h	8
Interrupt source priority register 83	IPR83	00h	0008 7383h	8
Interrupt source priority register 85	IPR85	00h	0008 7385h	8
Interrupt source priority register 86	IPR86	00h	0008 7386h	8
Interrupt source priority register 88	IPR88	00h	0008 7388h	8
Interrupt source priority register 89	IPR89	00h	0008 7389h	8
Interrupt source priority register 8A	IPR8A	00h	0008 738Ah	8
Interrupt source priority register 8B	IPR8B	00h	0008 738Bh	8
Interrupt source priority register 8C	IPR8C	00h	0008 738Ch	8
Interrupt source priority register 8D	IPR8D	00h	0008 738Dh	8
Interrupt source priority register 8E	IPR8E	00h	0008 738Eh	8
Interrupt source priority register 8F	IPR8F	00h	0008 738Fh	8
DMACA activation source select register 0	DMRSR0	00h	0008 7400h	8
DMACA activation source select register 1	DMRSR1	00h	0008 7404h	8
DMACA activation source select register 2	DMRSR2	00h	0008 7408h	8
DMACA activation source select register 3	DMRSR3	00h	0008 740Ch	8
IRQ control register 0	IRQCR0	00h	0008 7500h	8
IRQ control register 1	IRQCR1	00h	0008 7501h	8
IRQ control register 2	IRQCR2	00h	0008 7502h	8
IRQ control register 3	IRQCR3	00h	0008 7503h	8
IRQ control register 4	IRQCR4	00h	0008 7504h	8
IRQ control register 5	IRQCR5	00h	0008 7505h	8
IRQ control register 6	IRQCR6	00h	0008 7506h	8
IRQ control register 7	IRQCR7	00h	0008 7507h	8
IRQ control register 8	IRQCR8	00h	0008 7508h	8
IRQ control register 9	IRQCR9	00h	0008 7509h	8
IRQ control register 10	IRQCR10	00h	0008 750Ah	8
IRQ control register 11	IRQCR11	00h	0008 750Bh	8
IRQ control register 12	IRQCR12	00h	0008 750Ch	8

Table 11.3 Registers of the Interrupt Control Unit (10 / 10)

Register Name	Symbol	Value after Reset	Address	Access Size
IRQ control register 13	IRQCR13	00h	0008 750Dh	8
IRQ control register 14	IRQCR14	00h	0008 750Eh	8
IRQ control register 15	IRQCR15	00h	0008 750Fh	8
Non-maskable interrupt status register	NMISR	00h	0008 7580h	8
Non-maskable interrupt enable register	NMIER	00h	0008 7581h	8
Non-maskable interrupt clear register	NMICLR	00h	0008 7582h	8
NMI pin interrupt control register	NMICR	00h	0008 7583h	8

11.2.1 Interrupt Request Register i (IRi) (i = interrupt vector number)

Addresses: 0008 7010h to 0008 70FDh



Bit	Symbol	Bit Name	Description	R/W
b0	IR	Interrupt Status Flag	0: No interrupt request is generated 1: An interrupt request is generated	R/W*
b7 to b1	—	Reserved	These bits are read as 0. The write value should always be 0.	R/W

Note 1. For edge-detected sources, only 0 can be written to this bit, which clears the flag, and writing 1 to the bit is enabled under the condition described in section 11.7, Usage Notes.

The IRi register indicates interrupt request status.

IRi is provided for each interrupt source, where "i" indicates the interrupt vector number.

For the correspondence between interrupt sources and interrupt vector numbers, see Table 11.4, 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. For the actual interrupt to be generated, interrupt output by the corresponding peripheral module should be enabled in the case of peripheral-module interrupts.

The form of detection for an interrupt request is either edge detection or level detection. For interrupts from peripheral modules, either edge detection or level detection is determined per interrupt source. For interrupts from IRQn pins, edge detection or level detection is selected by the setting of the IRQMD[1:0] bits in the corresponding IRQCRn (n = 0 to 15). Regarding detection of the various interrupt sources, see Table 11.4, 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 IRQn pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.
- Though writing 1 to the IR flag is prohibited, it is enabled under the condition described in section 11.7, Usage Notes.

[Clearing conditions]

- The flag is cleared to 0 when the interrupt request destination accepts the interrupt request.
- The IR flag is cleared to 0 by writing 0 to it. Note, however, that writing 0 to the IR flag is prohibited if the destination of the interrupt request is the DTC or DMACA.

(2) Level detection

[Setting condition]

- The flag remains set to 1 while an interrupt request is being sent from the corresponding peripheral module or IRQn pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.

[Clearing conditions]

- The flag is only cleared to 0 when the source of the interrupt request is cleared (it is not cleared when the interrupt request destination accepts the interrupt request). For clearing interrupts from the various peripheral modules, refer to the sections describing the modules.

When level detection has been selected for an IRQn pin, the interrupt request is withdrawn by driving the IRQn pin high. Do not write 0 or 1 to the IR flag while level detection is selected.

11.2.2 Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh)

Addresses: 0008 7202h to 0008 721Fh

b7	b6	b5	b4	b3	b2	b1	b0
IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0

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

Bit	Symbol	Bit Name	Description	R/W
b0	IEN0	Interrupt Request Enable 0	0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
b1	IEN1	Interrupt Request Enable 1		R/W
b2	IEN2	Interrupt Request Enable 2		R/W
b3	IEN3	Interrupt Request Enable 3		R/W
b4	IEN4	Interrupt Request Enable 4		R/W
b5	IEN5	Interrupt Request Enable 5		R/W
b6	IEN6	Interrupt Request Enable 6		R/W
b7	IEN7	Interrupt Request Enable 7		R/W

Note: * Write 0 to the bit that corresponds to the vector number for reservation. These bits are read as 0.

The IERm register is used to enable or disable interrupt requests to the CPU or DMACA/DTC activation requests.

IENj Bits (Interrupt Request Enable) (j = 7 to 0)

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 11.2.1, Interrupt Request Register i (IRi) (i = interrupt vector number).

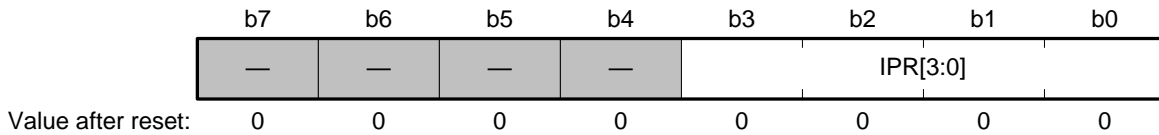
The IERm.IENj bit is set for each request source (vector number).

For the correspondence between interrupt sources and IERm.IENj bits, see Table 11.4, Interrupt Vector Table.

For the procedure for setting IERm.IENj bits during the selection of destinations for interrupt requests, refer to section 11.4.3, Selecting Interrupt Request Destinations.

11.2.3 Interrupt Priority Register m (IPRm) (m = 00h to 8Fh)

Addresses: 0008 7300h to 0008 738Fh



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>b2</td> <td>b1</td> <td>b0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>: Level 0 (interrupt prohibited)</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>: Level 1</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>: Level 2</td> </tr> <tr> <td></td> <td></td> <td></td> <td>:</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>: Level 15 (highest)</td> </tr> </table>	b3	b2	b1	b0		0	0	0	0	: Level 0 (interrupt prohibited)	0	0	0	1	: Level 1	0	0	1	0	: Level 2				:		1	1	1	1	: Level 15 (highest)	R/W
b3	b2	b1	b0																															
0	0	0	0	: Level 0 (interrupt prohibited)																														
0	0	0	1	: Level 1																														
0	0	1	0	: Level 2																														
			:																															
1	1	1	1	: Level 15 (highest)																														
b7 to b4	—	Reserved	This bit is read as 0. The write value should always be 0.	R/W																														

The IPRm register is used to set the priority level of an interrupt source.

IPRm exists for each interrupt source group, where m is the serial number from 00h to 8Fh.

For the correspondence between interrupt sources and groups, see Table 11.4, Interrupt Vector Table.

IPR[3:0] Bits (Interrupt Priority Level Select)

These bits select the priority level of the corresponding interrupt source.

Priority levels selected by the IPR[3:0] bits are used only to determine the priority of interrupt requests to be transferred to the CPU, and do not affect transfer requests to the DTC and DMACA.

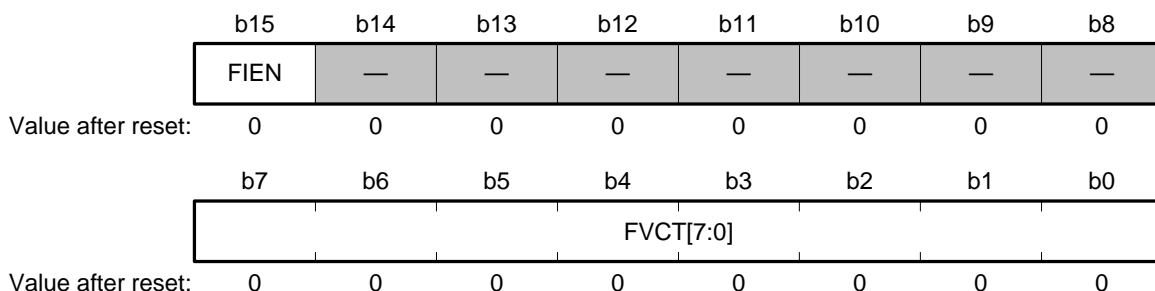
The CPU accepts only interrupt requests higher than the priority level specified by the IPL[3:0] bits in PSW, and handles accepted interrupts.

If two or more interrupt requests are generated at the same time, their priority levels are compared with the value of the IPR[3:0] bits. If interrupt requests of the same priority level are generated at the same time, an interrupt source with a smaller vector number takes precedence.

These bits should be written to while an interrupt request is disabled (the IERm.IENj bit = 0).

11.2.4 Fast Interrupt Register (FIR)

Address: 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 always be 0.	R/W
b15	FIEN	Fast Interrupt Enable	0: Fast interrupt is disabled 1: Fast interrupt is enabled	R/W

The FIR register is used to set the fast interrupt function.

The fast interrupt function based on the FIR register setting is applicable only to interrupts to the CPU. It will not affect any transfer request to the DTC or DMACA.

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 that uses the fast interrupt function.

FIEN Bit (Fast Interrupt Enable)

This bit enables 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 IPRm register. When using the fast interrupt for returning from the software standby mode, see section 11.6.3, Return from Software Standby Mode.

If the setting of the IERm.IENj (m = 02h to 1Fh, j = 7 to 0) bit has disabled interrupt requests from the interrupt source with the vector number in this register, fast interrupt requests are not output to the CPU.

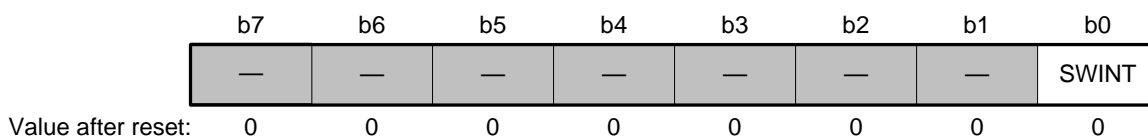
For settable vector numbers, see Table 11.4, Interrupt Vector Table.

Do not write any reserved vector numbers to the FVCT[7:0] bits.

For details of fast interrupt, see section 10, Exceptions, and section 11.4.5, Fast Interrupt.

11.2.5 Software Interrupt Activation Register (SWINTR)

Address: 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*
b7 to b1	—	Reserved	These bits are read as 0. The write value should always be 0.	R/W

Note: * Only 1 can be written. This bit is read as 0.

The SWINTR register is used to issue a software interrupt request.

SWINT Bit (Software Interrupt Activation)

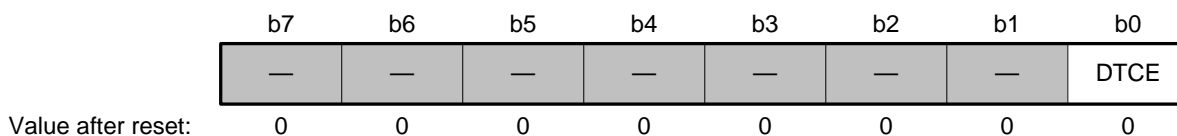
When 1 is written to the SWINT bit, the interrupt request register 27 (IR27) is set to 1.

If 1 is written to the SWINT bit when the DTC activation enable register 27 (DTCER27) is set to 0, an interrupt to the CPU is generated.

If 1 is written to the SWINT bit when the DTC activation enable register 27 (DTCER27) is set to 1, a DTC activation request is issued.

11.2.6 DTC Activation Enable Register n (DTCERn) (n = interrupt vector number)

Addresses: 0008 711Bh to 0008 71FCh



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 always be 0.	R/W

The DTCERn register is used to select an interrupt source to activate the DTC.

The interrupt source that has been selected for the DMACA activation should not be specified as the source for the 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

11.2.7 DMACA Activation Source Select Register n (DMRSRn) (n = DMACA channel number)

Addresses: DMRSR0 0008 7400h, DMRSR1 0008 7404h
DMRSR2 0008 7408h, DMRSR3 0008 740Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DMRS[7:0]	DMACA Activation Request Select	These bits specify the vector number for the DMACA activation request.	R/W

The DMRSRn register selects an interrupt source to activate the DMACA.

To specify the same interrupt source for multiple DMRSRn registers is disabled. The interrupt source that has been selected for the DMRSRn activation should not be specified as the source for the DTC activation. Otherwise, the correct operation is not guaranteed.

DMRS[7:0] Bits (DMACA Activation Request Select)

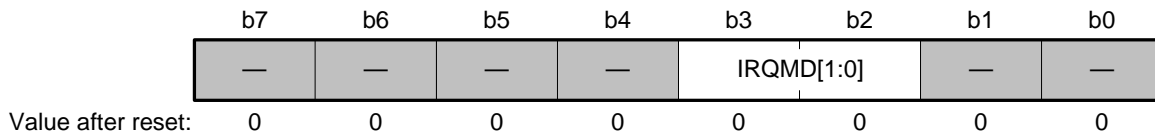
The vector number of the interrupt source for DMACA activation is specified in 8 bits. Do not set the vector numbers that are not assigned for the DMACA activation.

For the correspondence between interrupt sources and interrupt vector numbers, see Table 11.4, Interrupt Vector Table.

Write to the DMRSRn register while the DMA transfer enable bit of the DMACA transfer enable register (DMACn.DMCNT.DTE) is cleared to 0.

11.2.8 IRQ Control Register n (IRQCRn) (n = 0 to 15)

Addresses: 0008 7500h to 0008 750Fh



Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should always be 0.	R/W
b3, b2	IRQMD[1:0]	IRQ Detection Sense Select	$b^3 b^2$ 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Rising and falling edges	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should always be 0.	R/W

The IRQCRn register is used to set the external interrupt IRQn pin (n = 0 to 15).

Only change the settings of this register while the corresponding interrupt request enable bit is prohibiting the interrupt request (IENj bit in IERm is 0). After changing the setting, clear the IR flag in IRn before setting the interrupt enable bit. However, when the setting is changed to the low level, clearing the IR flag is not necessary.

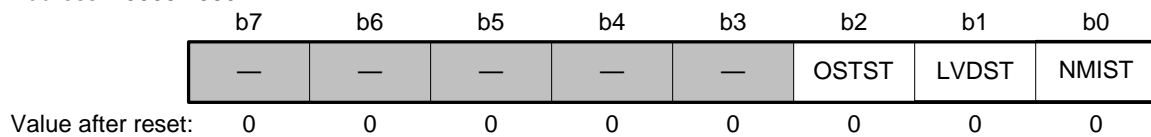
IRQMD[1:0] Bits (IRQ Detection Sense Select)

These bits select the detection sensing method of external pin interrupt sources IRQ0 to IRQ15.

For the external pin interrupt detection setting, see section 11.4.6, External Pin Interrupts.

11.2.9 Non-Maskable Interrupt Status Register (NMISR)

Address: 0008 7580h



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	LVDST	Voltage Monitoring Interrupt Status Flag	0: Voltage monitoring interrupt is not requested 1: Voltage monitoring interrupt is requested	R
b2	OSTST	Oscillation Stop Detection Interrupt Status Flag	0: Oscillation stop detection interrupt is not requested 1: Oscillation stop detection interrupt is requested	R
b7 to b3	—	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 ending the non-maskable interrupt routine, read the NMISR register and confirm the generation status of other non-maskable interrupts. Make sure to end the routine after confirming that all the bits in the NMISR register are set to 0.

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

LVDST Flag (Voltage Monitoring Interrupt Status Flag)

This flag indicates the request for voltage monitoring interrupt.

[Setting condition]

- When the voltage monitoring interrupt is generated

[Clearing condition]

- When the interrupt is cleared by the generation source

OSTST Flag (Oscillation Stop Detection Interrupt Status Flag)

This flag indicates the request for oscillation stop detection interrupt.

The OSTST flag is read-only, and cleared by NMICLR.OSTCLR bit.

[Setting condition]

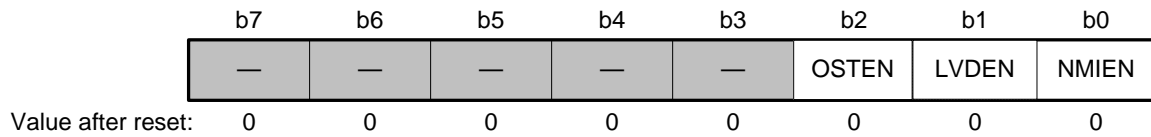
- When oscillation stop detection interrupt is generated

[Clearing condition]

- When 1 is written to the NMICLR.OSTCLR bit

11.2.10 Non-Maskable Interrupt Enable Register (NMIER)

Address: 0008 7581h



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)*
b1	LVDEN	Voltage Monitoring Interrupt Enable	0: Voltage monitoring interrupt is disabled 1: Voltage monitoring interrupt is enabled	R/(W)*
b2	OSTEN	Oscillation Stop Detection Interrupt Enable	0: Oscillation stop detection interrupt is disabled 1: Oscillation stop detection interrupt is enabled	R/(W)*
b7 to b3	—	Reserved	These bits are read as 0. The write value should always be 0.	R/W

Note: * A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

The NMIER register enables/disables a non-maskable interrupt source.

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.

Once the NMI pin interrupt is enabled, the interrupt cannot be canceled and writing 0 to this bit is disabled.

LVDEN Bit (Voltage Monitoring Interrupt Enable)

This bit enables the voltage monitoring interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Once the voltage monitoring interrupt is enabled, the interrupt cannot be canceled and writing 0 to this bit is disabled.

OSTEN Bit (Oscillation Stop Detection Interrupt Enable)

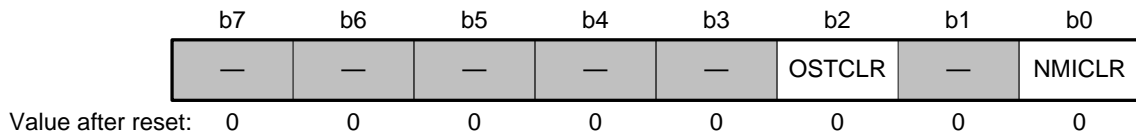
This bit enables the oscillation stop detection interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Once the oscillation stop detection interrupt is enabled, the interrupt cannot be canceled and writing 0 to this bit is disabled.

11.2.11 Non-Maskable Interrupt Clear Register (NMICLR)

Address: 0008 7582h



Bit	Symbol	Bit Name	Description	R/W
b0	NMICLR	NMI Clear	This bit is always 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	—	Reserved	This bit is read as 0. The write value should always be 0.	R/W
b2	OSTCLR	OST Clear	This bit is always read as 0. Writing 1 to this bit clears the NMISR.OSTST flag. Writing 0 to this bit has no effect.	R/(W)*2
b7 to b1	—	Reserved	These bits are read as 0. The write value should always be 0.	R/W

Note 1. Only 1 can be written to this bit, to clear the NMISR.NMIST flag.

Note 2. Only 1 can be written to this bit, to clear the NMISR.OSTST flag.

The NMICLR register is used to clear the non-maskable interrupt status register (NMISR).

NMICLR Bit (NMI Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag.

The NMICLR bit does not retain the "1" state. This bit is read as 0.

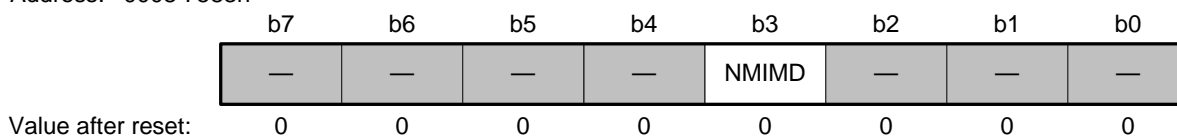
OSTCLR Bit (OST Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag.

The OSTCLR bit does not retain the "1" state. This bit is read as 0.

11.2.12 NMI Pin Interrupt Control Register (NMICR)

Address: 0008 7583h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should always be 0.	R/W
b3	NMIMD	NMI Detection Set	0: Falling edge 1: Rising edge	R/(W)*2
b7 to b4	—	Reserved	These bits are read as 0 and cannot be modified.	R

The NMICR register is used to set the NMI pin interrupt.

Change the setting of the NMICR register before the NMI pin interrupt is enabled (before 1 is written to the NMIER.NMIEN bit).

NMIMD Bit (NMI Detection Set)

This bit specifies the detection edge of the NMI pin interrupt.

Table 11.4 Interrupt Vector Table (2 / 6)

Priority	Source of Interrupt Request	Name	Vector No.	Vector Address Offset	Form of Detection	Selectable Interrupt Request Destination			Sstb Return	Sacs Return	IER	IPR
						CPU	DTC	DMA				
<div style="display: flex; align-items: center;"> <div style="border-left: 1px solid black; border-right: 1px solid black; width: 10px; height: 100%; margin-right: 5px;"></div> <div style="writing-mode: vertical-rl; transform: rotate(180deg); font-size: small;">High ↑</div> </div>	USB0	D0FIFO0	36	0090h	Edge	√	√	√	x	x	IER04.IEN4	IPR0C
		D1FIFO0	37	0094h	Edge	√	√	√	x	x	IER04.IEN5	IPR0D
		USBI0	38	0098h	Edge	√	x	x	x	x	IER04.IEN6	IPR0E
	—	Reserved	39	009Ch	—	x	x	x	x	x	IER04.IEN7	IPR0F
	USB1	D0FIFO1	40	00A0h	Edge	√	√	√	x	x	IER05.IEN0	IPR10
		D1FIFO1	41	00A4h	Edge	√	√	√	x	x	IER05.IEN1	IPR11
		USBI1	42	00A8h	Edge	√	x	x	x	x	IER05.IEN2	IPR12
	—	Reserved	43	00ACh	—	x	x	x	x	x	IER05.IEN3	IPR13
	RSPIO	SPEI0	44	00B0h	Level	√	x	x	x	x	IER05.IEN4	IPR14
		SPRI0	45	00B4h	Edge	√	√	√	x	x	IER05.IEN5	
		SPTI0	46	00B8h	Edge	√	√	√	x	x	IER05.IEN6	
		SPII0	47	00BCh	Level	√	x	x	x	x	IER05.IEN7	
	RSPI1	SPEI1	48	00C0h	Level	√	x	x	x	x	IER06.IEN0	IPR15
		SPRI1	49	00C4h	Edge	√	√	√	x	x	IER06.IEN1	
		SPTI1	50	00C8h	Edge	√	√	√	x	x	IER06.IEN2	
		SPII1	51	00CCh	Level	√	x	x	x	x	IER06.IEN3	
	—	Reserved	52	00D0h	—	x	x	x	x	x	IER06.IEN4	
	—	Reserved	53	00D4h	—	x	x	x	x	x	IER06.IEN5	—
	—	Reserved	54	00D8h	—	x	x	x	x	x	IER06.IEN6	—
	—	Reserved	55	00DCh	—	x	x	x	x	x	IER06.IEN7	—
	CAN0	ERS0	56	00E0h	Edge	√	x	x	x	x	IER07.IEN0	IPR18
		RXF0	57	00E4h	Edge	√	x	x	x	x	IER07.IEN1	
		TXF0	58	00E8h	Edge	√	x	x	x	x	IER07.IEN2	
		RXM0	59	00ECh	Edge	√	x	x	x	x	IER07.IEN3	
		TXM0	60	00F0h	Edge	√	x	x	x	x	IER07.IEN4	
	—	Reserved	61	00F4h	—	x	x	x	x	x	IER07.IEN5	IPR1D
	RTC	PRD	62	00F8h	Edge	√	x	x	x	x	IER07.IEN6	IPR1E
		CUP	63	00FCh	Edge	√	x	x	x	x	IER07.IEN7	IPR1F
	External pins	IRQ0	64	0100h	Edge/level	√	√	√	√	√	IER08.IEN0	IPR20
		IRQ1	65	0104h	Edge/level	√	√	√	√	√	IER08.IEN1	IPR21
		IRQ2	66	0108h	Edge/level	√	√	√	√	√	IER08.IEN2	IPR22
		IRQ3	67	010Ch	Edge/level	√	√	√	√	√	IER08.IEN3	IPR23
		IRQ4	68	0110h	Edge/level	√	√	x	√	√	IER08.IEN4	IPR24
		IRQ5	69	0114h	Edge/level	√	√	x	√	√	IER08.IEN5	IPR25
		IRQ6	70	0118h	Edge/level	√	√	x	√	√	IER08.IEN6	IPR26
		IRQ7	71	011Ch	Edge/level	√	√	x	√	√	IER08.IEN7	IPR27
		IRQ8	72	0120h	Edge/level	√	√	x	√	√	IER09.IEN0	IPR28
		IRQ9	73	0124h	Edge/level	√	√	x	√	√	IER09.IEN1	IPR29
		IRQ10	74	0128h	Edge/level	√	√	x	√	√	IER09.IEN2	IPR2A
		IRQ11	75	012Ch	Edge/level	√	√	x	√	√	IER09.IEN3	IPR2B
		IRQ12	76	0130h	Edge/level	√	√	x	√	√	IER09.IEN4	IPR2C
		IRQ13	77	0134h	Edge/level	√	√	x	√	√	IER09.IEN5	IPR2D
		IRQ14	78	0138h	Edge/level	√	√	x	√	√	IER09.IEN6	IPR2E
	IRQ15	79	013Ch	Edge/level	√	√	x	√	√	IER09.IEN7	IPR2F	
	—	Reserved	80 to 87	0140h to 015Ch	—	x	x	x	x	x	—	—
	—	Reserved	88	0160h	—	x	x	x	x	x	IER0B.IEN0	—
	Low	—	Reserved	89	0164h	—	x	x	x	x	IER0B.IEN1	—

Table 11.4 Interrupt Vector Table (3 / 6)

Priority	Source of Interrupt Request	Name	Vector No.	Vector Address Offset	Form of Detection	Selectable Interrupt Request Destination			Sstb Return	Sacs Return	IER	IPR
						CPU	DTC	DMA				
High ↑	USB	USB0	90	0168h	Level	√	x	x	√	√	IER0B.IEN2	IPR3A
		USB1	91	016Ch	Level	√	x	x	√	√	IER0B.IEN3	IPR3B
	RTC	ALM	92	0170h	Edge	√	x	x	√	√	IER0B.IEN4	IPR3C
	—	Reserved	93	0174h	—	x	x	x	x	x	IER0B.IEN5	—
	—	Reserved	94	0178h	—	x	x	x	x	x	IER0B.IEN6	—
	—	Reserved	95	017Ch	—	x	x	x	x	x	IER0B.IEN7	—
	WDT	WOVI	96	0180h	Edge	√	x	x	x	√	IER0C.IEN0	IPR40
	—	Reserved	97	0184h	—	x	x	x	x	x	IER0C.IEN1	—
	AD0	ADIO	98	0188h	Edge	√	√	√	x	x	IER0C.IEN2	IPR44
	AD1	ADI1	99	018Ch	Edge	√	√	√	x	x	IER0C.IEN3	IPR45
	—	Reserved	100	0190h	—	x	x	x	x	x	IER0C.IEN4	—
	—	Reserved	101	0194h	—	x	x	x	x	x	IER0C.IEN5	—
	S12AD	S12ADIO	102	0198h	Edge	√	√	√	x	x	IER0C.IEN6	IPR48
	—	Reserved	103 to 113	019Ch to 01C4h	—	x	x	x	x	x		—
	↓ Low	MTU0	TGIA0	114	01C8h	Edge	√	√	√	x	x	IER0E.IEN2
TGIB0			115	01CCh	Edge	√	√	x	x	x	IER0E.IEN3	
TGIC0			116	01D0h	Edge	√	√	x	x	x	IER0E.IEN4	
TGID0			117	01D4h	Edge	√	√	x	x	x	IER0E.IEN5	
TCIV0			118	01D8h	Edge	√	x	x	x	x	IER0E.IEN6	
TGIE0			119	01DCh	Edge	√	x	x	x	x	IER0E.IEN7	
TGIF0			120	01E0h	Edge	√	x	x	x	x	IER0F.IEN0	
MTU1		TGIA1	121	01E4h	Edge	√	√	√	x	x	IER0F.IEN1	IPR53
		TGIB1	122	01E8h	Edge	√	√	x	x	x	IER0F.IEN2	
		TCIV1	123	01ECh	Edge	√	x	x	x	x	IER0F.IEN3	
		TCIU1	124	01F0h	Edge	√	x	x	x	x	IER0F.IEN4	
MTU2		TGIA2	125	01F4h	Edge	√	√	√	x	x	IER0F.IEN5	IPR55
		TGIB2	126	01F8h	Edge	√	√	x	x	x	IER0F.IEN6	
		TCIV2	127	01FCh	Edge	√	x	x	x	x	IER0F.IEN7	
		TCIU2	128	0200h	Edge	√	x	x	x	x	IER10.IEN0	
MTU3	TGIA3	129	0204h	Edge	√	√	√	x	x	IER10.IEN1	IPR57	
	TGIB3	130	0208h	Edge	√	√	x	x	x	IER0E.IEN2		
	TGIC3	131	020Ch	Edge	√	√	x	x	x	IER10.IEN3		
	TGID3	132	0210h	Edge	√	√	x	x	x	IER10.IEN4		
	TCIV3	133	0214h	Edge	√	x	x	x	x	IER10.IEN5		
MTU4	TGIA4	134	0218h	Edge	√	√	√	x	x	IER10.IEN6	IPR59	
	TGIB4	135	021Ch	Edge	√	√	x	x	x	IER10.IEN7		
	TGIC4	136	0220h	Edge	√	√	x	x	x	IER11.IEN0		
	TGID4	137	0224h	Edge	√	√	x	x	x	IER11.IEN1		
	TCIV4	138	0228h	Edge	√	√	x	x	x	IER11.IEN2		
MTU5	TGIU5	139	022Ch	Edge	√	√	x	x	x	IER11.IEN3	IPR5B	
	TGIV5	140	0230h	Edge	√	√	x	x	x	IER11.IEN4		
	TGIW5	141	0234h	Edge	√	√	x	x	x	IER11.IEN5		
MTU6	TGIA6	142	0238h	Edge	√	√		x	x	IER11.IEN6	IPR5C	
	TGIB6	143	023Ch	Edge	√	√	x	x	x	IER11.IEN7		
	TGIC6	144	0240h	Edge	√	√	x	x	x	IER12.IEN0		
	TGID6	145	0244h	Edge	√	√	x	x	x	IER12.IEN1		

Table 11.4 Interrupt Vector Table (4 / 6)

Priority	Source of Interrupt Request	Name	Vector No.	Vector Address Offset	Form of Detection	Selectable Interrupt Request Destination			Sstb Return	Sacs Return	IER	IPR	
						CPU	DTC	DMA					
High ↑	MTU6	TCIV6	146	0248h	Edge	√	x	x	x	x	IER12.IEN2	IPR5D	
		TGIE6	147	024Ch	Edge	√	x	x	x	x	IER12.IEN3		
		TGIF6	148	0250h	Edge	√	x	x	x	x	IER12.IEN4		
	MTU7	TGIA7	149	0254h	Edge	√	√	√	x	x	IER12.IEN5	IPR5E	
		TGIB7	150	0258h	Edge	√	√	x	x	x	IER12.IEN6		
		TCIV7	151	025Ch	Edge	√	x	x	x	x	IER12.IEN7		IPR5F
		TCIU7	152	0260h	Edge	√	x	x	x	x	IER13.IEN0		
	MTU8	TGIA8	153	0264h	Edge	√	√	√	x	x	IER13.IEN1	IPR60	
		TGIB8	154	0268h	Edge	√	√	x	x	x	IER13.IEN2		
		TCIV8	155	026Ch	Edge	√	x	x	x	x	IER13.IEN3		IPR61
		TCIU8	156	0270h	Edge	√	x	x	x	x	IER13.IEN4		
	MTU9	TGIA9	157	0274h	Edge	√	√	√	x	x	IER13.IEN5	IPR62	
		TGIB9	158	0278h	Edge	√	√	x	x	x	IER13.IEN6		
		TGIC9	159	027Ch	Edge	√	√	x	x	x	IER13.IEN7		
		TGID9	160	0280h	Edge	√	√	x	x	x	IER14.IEN0		
		TCIV9	161	0284h	Edge	√	x	x	x	x	IER14.IEN1		IPR63
	MTU10	TGIA10	162	0288h	Edge	√	√	√	x	x	IER14.IEN2	IPR64	
		TGIB10	163	028Ch	Edge	√	√	x	x	x	IER14.IEN3		
		TGIC10	164	0290h	Edge	√	√	x	x	x	IER14.IEN4		
		TGID10	165	0294h	Edge	√	√	x	x	x	IER14.IEN5		
		TCIV10	166	0298h	Edge	√	√	x	x	x	IER14.IEN6		IPR65
	MTU11	TGIU11	167	029Ch	Edge	√	√	x	x	x	IER14.IEN7	IPR66	
		TGIV11	168	02A0h	Edge	√	√	x	x	x	IER15.IEN0		
		TGIW11	169	02A4h	Edge	√	√	x	x	x	IER15.IEN1		
	POE	OEI1	170	02A8h	Level	√	x	x	x	x	IER15.IEN2	IPR67	
		OEI2	171	02ACh	Level	√	x	x	x	x	IER15.IEN3		
		OEI3	172	02B0h	Level	√	x	x	x	x	IER15.IEN4		
		OEI4	173	02B4h	Level	√	x	x	x	x	IER15.IEN5		
	TMR0	CMIA0	174	02B8h	Edge	√	√	x	x	√	IER15.IEN6	IPR68	
		CMIB0	175	02BCh	Edge	√	√	x	x	√	IER15.IEN7		
		OVI0	176	02C0h	Edge	√	x	x	x	√	IER16.IEN0		
	TMR1	CMIA1	177	02C4h	Edge	√	√	x	x	√	IER16.IEN1	IPR69	
CMIB1		178	02C8h	Edge	√	√	x	x	√	IER16.IEN2			
OVI1		179	02CCh	Edge	√	x	x	x	√	IER16.IEN3			
TMR2	CMIA2	180	02D0h	Edge	√	√	x	x	√	IER16.IEN4	IPR6A		
	CMIB2	181	02D4h	Edge	√	√	x	x	√	IER16.IEN5			
	OVI2	182	02D8h	Edge	√	x	x	x	√	IER16.IEN6			
TMR3	CMIA3	183	02DCh	Edge	√	√	x	x	√	IER16.IEN7	IPR6B		
	CMIB3	184	02E0h	Edge	√	√	x	x	√	IER17.IEN0			
	OVI3	185	02E4h	Edge	√	x	x	x	√	IER17.IEN1			
—	Reserved	186	02E8h	—	x	x	x	x	x	IER17.IEN2			
—	Reserved	187	02ECh	—	x	x	x	x	x	IER17.IEN3			
—	Reserved	188	02F0h	—	x	x	x	x	x	IER17.IEN4			
—	Reserved	189	02F4h	—	x	x	x	x	x	IER17.IEN5			
—	Reserved	190	02F8h	—	x	x	x	x	x	IER17.IEN6			
—	Reserved	191	02FCh	—	x	x	x	x	x	IER17.IEN7			
Low	—	Reserved	192	0300h	—	x	x	x	x	x	IER18.IEN0		

Table 11.4 Interrupt Vector Table (5 / 6)

Priority	Source of Interrupt Request	Name	Vector No.	Vector Address Offset	Form of Detection	Selectable Interrupt Request Destination			Sstb Return	Sacs Return	IER	IPR
						CPU	DTC	DMA				
High ↑	—	Reserved	193	0304h	—	x	x	x	x	x	IER18.IEN1	
	—	Reserved	194	0308h	—	x	x	x	x	x	IER18.IEN2	
	—	Reserved	195	030Ch	—	x	x	x	x	x	IER18.IEN3	
	—	Reserved	196	0310h	—	x	x	x	x	x	IER18.IEN4	
	—	Reserved	197	0314h	—	x	x	x	x	x	IER18.IEN5	
	DMACA	DMACI0	198	0318h	Edge	√	√	x	x	x	IER18.IEN6	IPR70
		DMACI1	199	031Ch	Edge	√	√	x	x	x	IER18.IEN7	IPR71
		DMACI2	200	0320h	Edge	√	√	x	x	x	IER19.IEN0	IPR72
		DMACI3	201	0324h	Edge	√	√	x	x	x	IER19.IEN1	IPR73
	EXDMAC	EXDMACI0	202	0328h	Edge	√	√	x	x	x	IER19.IEN2	IPR74
		EXDMACI1	203	032Ch	Edge	√	√	x	x	x	IER19.IEN3	IPR75
	—	Reserved	204	0330h	—	x	x	x	x	x	IER19.IEN4	
	—	Reserved	205	0334h	—	x	x	x	x	x	IER19.IEN5	
	—	Reserved	206	0338h	—	x	x	x	x	x	IER19.IEN6	
	—	Reserved	207	033Ch	—	x	x	x	x	x	IER19.IEN7	
	—	Reserved	208	0340h	—	x	x	x	x	x	IER1A.IEN0	
	—	Reserved	209	0344h	—	x	x	x	x	x	IER1A.IEN1	
	—	Reserved	210	0348h	—	x	x	x	x	x	IER1A.IEN2	
	—	Reserved	211	034Ch	—	x	x	x	x	x	IER1A.IEN3	
	—	Reserved	212	0350h	—	x	x	x	x	x	IER1A.IEN4	
	—	Reserved	213	0354h	—	x	x	x	x	x	IER1A.IEN5	
	SCI0	ERI0	214	0358h	Level	√	x	x	x	x	IER1A.IEN6	IPR80
		RXI0	215	035Ch	Edge	√	√	√	x	x	IER1A.IEN7	
		TXI0	216	0360h	Edge	√	√	√	x	x	IER1B.IEN0	
		TEI0	217	0364h	Level	√	x	x	x	x	IER1B.IEN1	
	SCI1	ERI1	218	0368h	Level	√	x	x	x	x	IER1B.IEN2	IPR81
		RXI1	219	036Ch	Edge	√	√	√	x	x	IER1B.IEN3	
		TXI1	220	0370h	Edge	√	√	√	x	x	IER1B.IEN4	
		TEI1	221	0374h	Level	√	x	x	x	x	IER1B.IEN5	
	SCI2	ERI2	222	0378h	Level	√	x	x	x	x	IER1B.IEN6	IPR82
		RXI2	223	037Ch	Edge	√	√	√	x	x	IER1B.IEN7	
		TXI2	224	0380h	Edge	√	√	√	x	x	IER1C.IEN0	
		TEI2	225	0384h	Level	√	x	x	x	x	IER1C.IEN1	
	SCI3	ERI3	226	0388h	Level	√	x	x	x	x	IER1C.IEN2	IPR83
		RXI3	227	038Ch	Edge	√	√	√	x	x	IER1C.IEN3	
		TXI3	228	0390h	Edge	√	√	√	x	x	IER1C.IEN4	
		TEI3	229	0394h	Level	√	x	x	x	x	IER1C.IEN5	
	—	Reserved	230	0398h	—	x	x	x	x	x	IER1C.IEN6	—
	—	Reserved	231	039Ch	—	x	x	x	x	x	IER1C.IEN7	
	—	Reserved	232	03A0h	—	x	x	x	x	x	IER1D.IEN0	
	—	Reserved	233	03A4h	—	x	x	x	x	x	IER1D.IEN1	
	SCI5	ERI5	234	03A8h	Level	√	x	x	x	x	IER1D.IEN2	IPR85
	RXI5	235	03ACh	Edge	√	√	√	x	x	IER1D.IEN3		
	TXI5	236	03B0h	Edge	√	√	√	x	x	IER1D.IEN4		
	TEI5	237	03B4h	Level	√	x	x	x	x	IER1D.IEN5		
SCI6	ERI6	238	03B8h	Level	√	x	x	x	x	IER1D.IEN6	IPR86	
	RXI6	239	03BCh	Edge	√	√	√	x	x	IER1D.IEN7		
Low												

Table 11.4 Interrupt Vector Table (6 / 6)

Priority	Source of Interrupt Request	Name	Vector No.	Vector Address Offset	Form of Detection	Selectable Interrupt Request Destination			Sstb Return	Sacs Return	IER	IPR	
						CPU	DTC	DMA					
High ↑ Low	SCI6	TXI6	240	03C0h	Edge	√	√	√	x	x	IER1E.IEN0	IPR86	
		TEI6	241	03C4h	Level	√	x	x	x	x	IER1E.IEN1		
	—	Reserved	242	03C8h	—	x	x	x	x	x	IER1E.IEN2	—	
	—	Reserved	243	03CCh	—	x	x	x	x	x	IER1E.IEN3	—	
	—	Reserved	244	03D0h	—	x	x	x	x	x	IER1E.IEN4	—	
	—	Reserved	245	03D4h	—	x	x	x	x	x	IER1E.IEN5	—	
	RIIC0	ICEEI0	ICRXI0	246	03D8h	Level	√	x	x	x	x	IER1E.IEN6	IPR88
			ICTXI0	247	03DCh	Edge	√	√	√	x	x	IER1E.IEN7	IPR89
		ICTEI0	ICRXI0	248	03E0h	Edge	√	√	√	x	x	IER1F.IEN0	IPR8A
			ICTEI0	249	03E4h	Level	√	x	x	x	x	IER1F.IEN1	IPR8B
	RIIC1	ICEEI1	ICRXI1	250	03E8h	Level	√	x	x	x	x	IER1F.IEN2	IPR8C
			ICTXI1	251	03ECh	Edge	√	√	√	x	x	IER1F.IEN3	IPR8D
		ICTEI1	ICRXI1	252	03F0h	Edge	√	√	√	x	x	IER1F.IEN4	IPR8E
			ICTEI1	253	03F4h	Level	√	x	x	x	x	IER1F.IEN5	IPR8F
	—	Reserved	254	03F8h	—	x	x	x	x	x	IER1F.IEN6	IPR90	
—	Reserved	255	03FCh	—	x	x	x	x	x	IER1F.IEN7	IPR91		

[Legend] √: Selectable x: Not selectable

11.3.2 Fast Interrupt Vector

The address of the entry in the interrupt vector table that corresponds to the vector number of the fast interrupt is specified by the fast interrupt vector register (FINTV) of the CPU.

11.3.3 Non-maskable Interrupt Vector

The vector for the NMI pin interrupt is at FFFF FFF8h.

11.4 Interrupt Operation

The interrupt controller performs the following processing.

- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations (CPU interrupt, DTC activation, or DMACA activation)
- Determining priority

11.4.1 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 IRQn pins (n = 15 to 0) as interrupt sources by the setting of the IRQMD[1:0] bits in IRQCRn.

For interrupts from peripheral modules, either edge detection or level detection is determined per interrupt source.

For the correspondence between interrupt sources and methods of detection, see Table 11.4, Interrupt Vector Table.

11.4.1.1 Operation of Status Flags for Edge-Detected Interrupts

Figure 11.2 shows the operation of the IR flag in IRi in the case of edge detection of an interrupt from a peripheral module or on an external pin.

The IR flag in IRi 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 in IRi is automatically cleared on acceptance of the interrupt. If the DMACA or DTC is the request destination for the interrupt, the timing of the IR flag clear depends on the DMAC/DTC transfer setting and number of times of transfer. For details, see Table 11.5. Therefore, the software does not have to clear the IR flag in IRi.

With regard to interrupt signals, the timing of interrupts with interrupt vector numbers from 64 to 95 differs from the timing for the other interrupts. In the case of the IRQ-pin interrupts, which take up the interrupt vector numbers from 64 to 79, internal input of the IRQ signal is extended by an internal delay plus two cycles of PCLK from actual input to the IRQ pin. In the case of the interrupts with interrupt vector numbers from 80 to 95, internal input of the interrupt signal is extended by two cycles of PCLK.

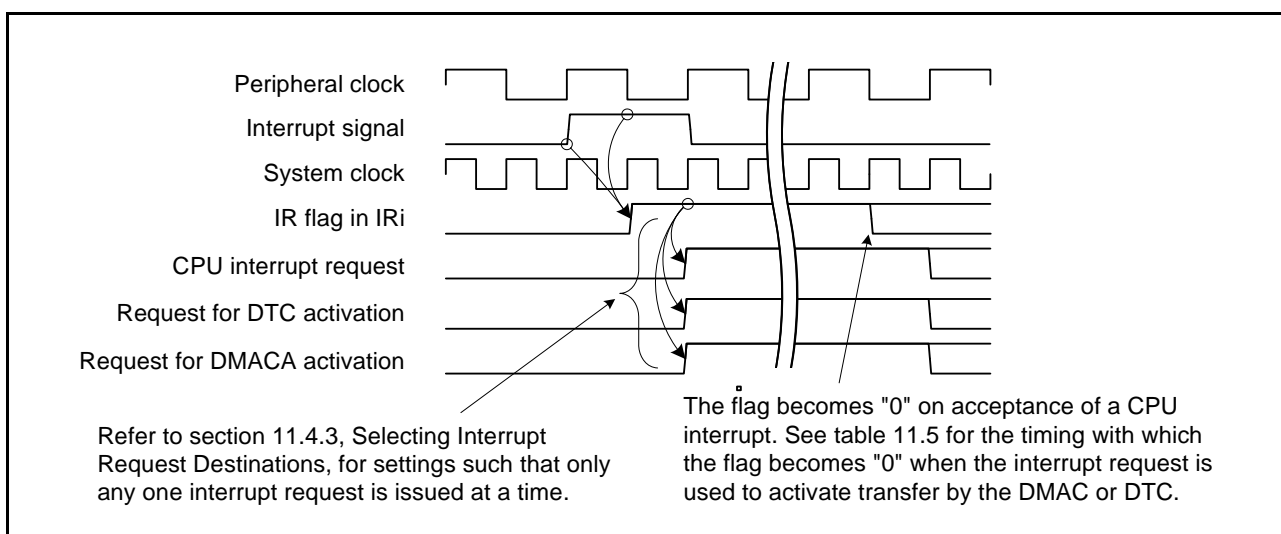


Figure 11.2 Operation of the IRi.IR Flag in the Case of Edge Detection

If the corresponding IR flag in IRI has already been set to 1 by a generated interrupt request, the generation of a further interrupt request will be ignored. If the IR flag in IRI has been cleared, it will be set again by the generation of a further interrupt request. The timing for re-setting of the IR flag in IRI is shown in Figure 11.3.

When communication functions (SCI, RIIC, RSPI, USB) and DTC or DMACA are combined, an interrupt request is ignored and transfer requests may be lost in some cases. For details, see section 11.7, Usage Notes.

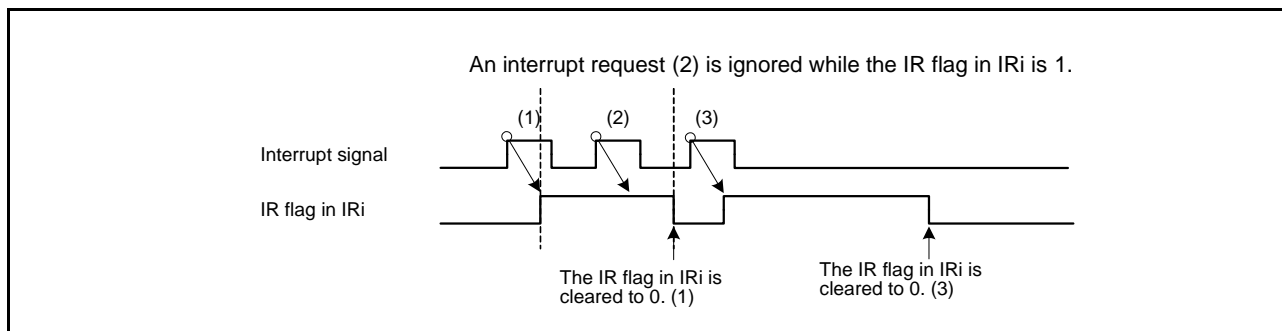


Figure 11.3 Timing for Re-setting of the IRI. IR Flag

If a source for interrupt generation is disabled over the period where the IR flag in the corresponding IRI is set (output of the interrupt request by the relevant peripheral module is disabled by clearing the interrupt enable bit), the IR flag is not affected but retains its state. Figure 11.4 shows operation when the source for interrupt generation has been disabled.

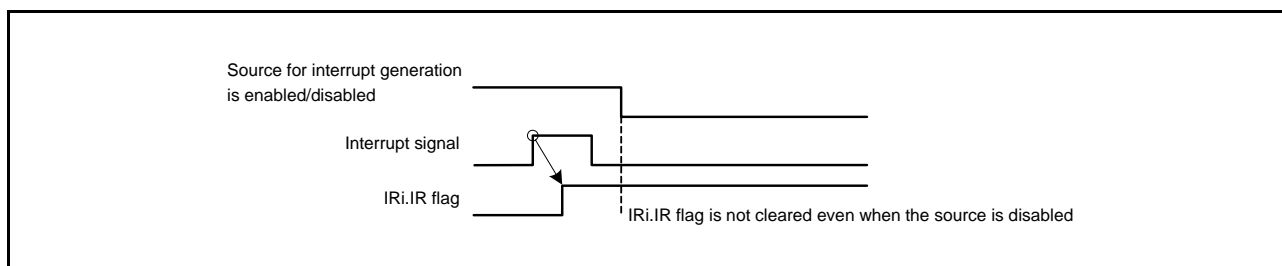


Figure 11.4 Relation between the IRn. IR Flag and Disabling of the Corresponding Interrupt Source

11.4.1.2 Operation of Status Flags for Level-Detected Interrupts

Figure 11.5 shows the operation of the interrupt status flag (IR flag) in IRi in the case of level detection of an interrupt from a peripheral module or an external pin.

The IR flag in IRi remains set as long as the interrupt signal is asserted. To clear the IR flag in IRi, clear the interrupt request in the source generating the interrupt.

Confirm that the interrupt request flag in the source generating the interrupt has actually been cleared to 0 before stopping the interrupt routine. Figure 11.6 shows the level-detected interrupt handling procedure.

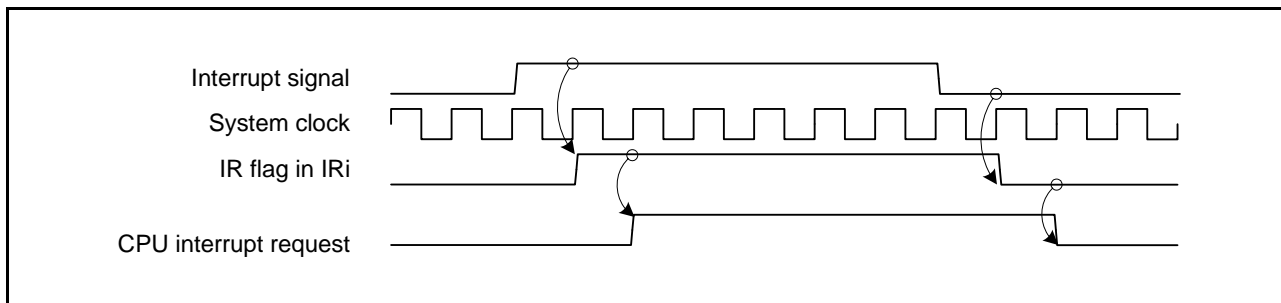


Figure 11.5 Operation of the IRi.IR Flag in the Case of a Level-Detected Interrupt

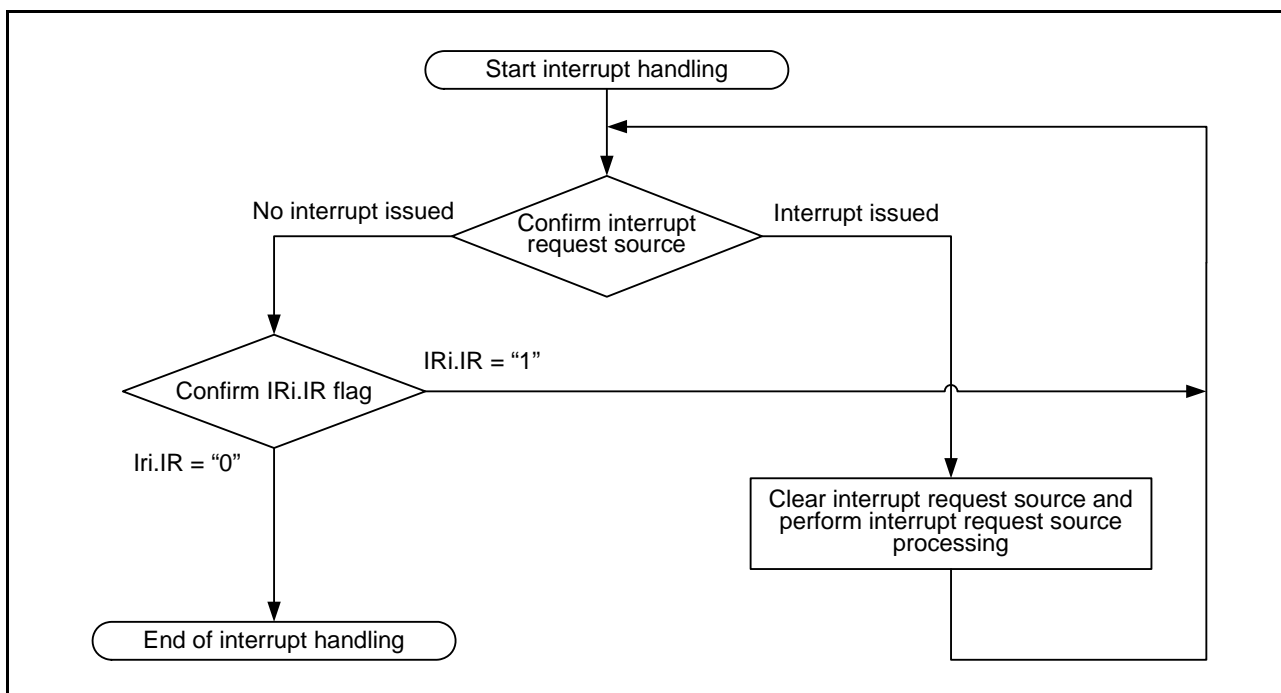


Figure 11.6 Level-Detected Interrupt Handling Procedure

11.4.2 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 IENj bit in IERm

When an interrupt request is generated by a source for interrupt generation for which interrupt output has been enabled, the IR flag in the corresponding IRi register is set.

Setting the IENj bit in IERm to enable an interrupt request causes a set IR flag in the IRi register corresponding to that interrupt request to be output to the interrupt request destination. Furthermore, if the IENj bit in IERm has been used to disable the interrupt request, the state of the interrupt request for which the IR flag in the corresponding IRi register has been set is held.

The IR flag in IRi is not affected by the IENj bit in IERm.

11.4.3 Selecting Interrupt Request Destinations

Possible settings for the request destination of each interrupt are fixed. That is, settings for request destination other than those indicated in Table 11.4, Interrupt Vector Table, are not possible. Do not make an interrupt request destination setting that is not indicated by a O in Table 11.4.

If the DMACA or DTC is selected as the destination for requests from an IRQ pin, be sure to set the IRQMD[1:0] bits in IRQCRn for that interrupt to select edge detection.

The following describes how to specify the destinations of interrupt requests.

(1) DMACA Activation

Make the following settings while the IERm.IENj bit is 0.

- Specify the vector number of the desired interrupt in the DMACA activation source select register (DMRSRn) for the required channel of the DMACA.*
- Set the activation source for the target DMAC channel (DMACAn.DMTMD.DCTG[1:0]) to 01b (interrupt module detection).
- Set the DMACA activation enable bit for the target DMAC channel (DMACAn.DMCNT.DTE) to 1.

After making the above settings, set the IERm.IENj bit to 1.

For the DMACA setting, see section 14.3.5, Activating the DMACA.

(2) DTC Activation

Make the following settings while the IERm.IENj bit is 0.

- Set the DTC activation enable bit (DTCERn.DTCE) in the DTC activation enable register for the required source to 1.*
- Set the DTC module activation bit (DTCST.DTCST) to 1.

After making the above settings, set the IERm.IENj bit to 1.

In addition, set the DTC module activation bit (DTCST.DTCST) to 1. The order of making settings for each interrupt and enabling the DMACA operation enable bit does not matter.

For the DTC setting, see section 16.5, DTC Setting Procedure.

(3) CPU Interrupt Request

If the interrupt request destination is neither the DMACA nor the DTC, the interrupt request is sent to the CPU. Set the IERm.IENj bit to 1 while neither the DMACA activation settings nor the DTC activation settings described above are in place.

Note: * Do not set a DTC activation enable bit (DTCERn.DTCE) and a DMACA activation request selection register (DMRSRn) to select the same source. Do not select the same source in more than one DMRSRn register.

Table 11.5 covers operation when the DTC or the DMACA is the request destination.

Table 11.5 Operation at the Time of DMACA/DTC Activation

Interrupt Request Destination	DISEL	Remaining Number of Transfer Operations	Operation per Request	IR*1*4	Interrupt Request Destination after Transfer
DMACA	1	≠0	DMA transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DMACA
		=0	DMA transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DTE bit in DMACn.DMCNT is cleared and the CPU becomes the destination.
	0	≠0	DMA transfer	Cleared on the start of DMACA transfer	DMACA
		=0	DMA transfer*2	Cleared on the start of DMACA transfer*2	The DTE bit in DMACn.DMCNT is cleared and the CPU becomes the destination.
DTC*3	1	≠0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		=0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DTCE bit in DTCR is cleared and the CPU becomes the destination.
	0	≠0	DTC transfer	Cleared on the start of DTC data transfer after DTC transfer information has been read	DTC
		=0	DTC transfer → CPU interrupt*2	Cleared on interrupt acceptance by the CPU*2	The DTCE bit in DTCR is cleared and the CPU becomes the destination.

DISEL for the DMACA is set by the DMACm.DMCSL.DISEL bit; DISEL for the DTC is set by the DTC.MRB.DISEL bit.

Note 1. An interrupt request (or request for DTC or DMACA activation) that is generated again while the IRn.IR flag is still set from a previous request will be ignored.

Note 2. Operation when DISEL = 0 differs according to whether the source is for DTC or DMACA activation.

Note 3. For the chain transfer, DTC transfer is continued until the last chain transfer completes. Each of operations of CPU interrupt, IR flag clear, and interrupt request destination after transfer for the last chain transfer, depends on the setting of DISEL bit for the last chain transfer and the remaining specified number of transfers.

Note 4. If a re-generated interrupt request (DTC/DMACA activation request) is ignored while the IRI.IR flag is set, some problems may occur to the communication functions (SCI, RIIC, RSPI, USB). For details, see section 11.7, Usage Notes.

The request destination for an interrupt should be changed while the IERm.IENj bit is 0.

When a source is to be changed to an interrupt request or the DMA activation source is to be changed while a transfer is not complete (i.e. while the DMACn.DMCNT.DTE bit has not been cleared) after the settings described under (1) DMACA Activation have been made, follow the procedure below.

1. For both the source to be withdrawn and the source that will have a new target for activation, clear the IENj bits in IERm to 0.
2. Check the state of transfer by the DMACA. If transfer is in progress, wait for its completion.
3. Make the settings described under (1) DMACA Activation.

When a source is to be changed to an interrupt request or the DTC transfer information is to be changed while a transfer is not complete (i.e. while the DTCERn.DTCE bit has not been cleared) after the settings described under (2) DTC Activation have been made, follow the procedure below.

1. For both the source to be withdrawn and the source that will have a new target for activation, clear the IENj bits in IERm to 0.
2. Check the state of transfer by the DTC. If transfer is in progress, wait for its completion.
3. Make the settings described under (2) DTC Activation.

11.4.4 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 Signal

A source selected for the fast interrupt has the highest priority. After that, an interrupt source with a larger value of the interrupt priority level select bits (IPR[3:0]) in IPRm takes priority. If interrupts with the same priority level are generated by multiple sources, the source with the smallest vector number takes precedence.

(2) Determining Priority when the DTC is the Request Destination of the Interrupt Signal

The IPR[3:0] bits in IPRm have no effect. An interrupt source with a smaller vector number takes precedence.

(3) Determining Priority when the DMACA is the Request Destination of the Interrupt Signal

The IPR[3:0] bits in IPRm have no effect. Regarding the order of priority of DMAC channels, see section 14, DMA Controller (DMACA).

11.4.5 Fast Interrupt

The fast interrupt is a facility for faster interrupt processing by the CPU, so is only effective for an interrupt request being conveyed to the CPU. That is, the fast interrupt setting has no effect on transfer requests for the DTC and DMACA.

The fast interrupt is set up by specifying the vector number of an interrupt source in the FVCT[7:0] bits of FIR and enabling the fast interrupt by setting the FIEN bit in FIR to 1. When the given source generates an interrupt, the interrupt is output to the CPU for handling as the fast interrupt.

The interrupt source selected for the fast interrupt has the highest priority regardless of the setting of the IPR[3:0] bits in IPRm.

For details on the fast interrupt, see section 10, Exceptions.

11.4.6 External Pin Interrupts

The procedure for using the signal on an external pin as an interrupt is given below.

1. Clear the IENj bit in IERm (IERm.IENj = 0).
2. Make or confirm the I/O port settings.
3. Set the method of detection for the interrupt in the IRQMD[1:0] bits of IRQCRn.
4. Clear the IR flag in the corresponding IRn register to 0 (if edge detection is in use).
5. If the interrupt is to be used for DMACA activation, set the DMRS[7:0] bits in DMRSRn. If the interrupt is to be used for DTC activation, set the DTCE bit in DTCERn. The interrupt will be a CPU interrupt if neither of these settings is made.
6. Set the IENj bit in IERm (IERm.IENj = 1).

11.5 Non-maskable Interrupt Operation

There are three types of non-maskable interrupts: the NMI pin interrupt, voltage monitoring interrupt, and oscillation stop detection interrupt. The DTC and DMACA are not selectable as destinations for non-maskable interrupts. Non-maskable interrupts take precedence over all maskable interrupts, including the fast interrupt.

A non-maskable interrupt request is accepted regardless of the settings of the I bit (interrupt enable bit) and the IPL[3:0] bits (processor interrupt priority level) in the PSW of the CPU. Whether a non-maskable interrupt is requested or not can be checked by the non-maskable interrupt status register (NMISR).

Confirm that all bits of the NMISR have returned to 0 from within the routine for the non-maskable interrupt.

To prevent malfunctions in systems that do not require non-maskable interrupts from the NMI pin, the NMI pin interrupt is disabled by default. If a system is to use NMI pin 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. Make the detection sense setting for the NMI in the NMIMD bit.
3. Write 1 to the NMIEN bit in NMIER to enable the NMI pin interrupts.
4. Enable the non-maskable interrupt by writing 1 to the corresponding bit in the non-maskable interrupt enable register (NMIER).

After the NMIEN bit in NMIER is set to 1, subsequent write access to the NMIEN bit in NMIER is ignored. The NMI interrupt cannot be disabled. The non-maskable interrupt is disabled only by a reset.

For the flow of non-maskable interrupt processing, see [section 10, Exceptions](#).

Writing 1 to the NMICLR.NMICLR bit clears the NMI pin interrupt flag (NMISR.NMIST).

Writing 1 to NMICLR.OSTCLR bit clears the oscillation stop detection interrupt flag (NMISR.OSTST).

For clearing the voltage monitoring interrupt status flag (NMISR.LVDST), see [section 7, Voltage Detection Circuit \(LVD\)](#).

Do not issue a WAIT instruction while any bit in the NMISR has the value 1.

11.6 Return from Power-Down States

The interrupt sources that can be used to return operation from sleep mode, all-module clock stop mode, or software standby mode are shown in Table 11.4, Interrupt Vector Table.

For details, refer to section 9, Low Power Consumption. The following describes how to use an interrupt to return operation from each power-down mode.

11.6.1 Return from Sleep Mode

If the interrupt control unit 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 IEN_j bit in IER_m to enable the given interrupt request.
 3. Set a priority level higher than that set by the IPL[3:0] bits in the PSW of CPU.
- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

11.6.2 Return from All-Module Clock Stop Mode

If the interrupt control unit is to return operation from all-module clock stop mode in response to an interrupt or non-maskable interrupt, make the following settings for the interrupt.

- Interrupts
 1. Select the interrupt source that enables the return from the all-module clock stop mode.
 2. Select the CPU as the interrupt request destination.
 3. Use the IEN_j bit in IER_m to enable the given interrupt request.
 4. Set a priority level higher than that set by the IPL[3:0] bits in the PSW of CPU.
- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

11.6.3 Return from Software Standby Mode

The interrupt control unit can return operation from a non-maskable interrupt or an interrupt that enables the return from the software standby mode.

The conditions for the return are listed below.

- Interrupts
 1. Select the interrupt source that enables the return from the software standby mode.
 2. Select the CPU as the interrupt request destination.
 3. Use the IEN_j bit in IER_m to enable the given interrupt request.
 4. Set a priority level higher than that set by the IPL[3:0] bits in the PSW of CPU.

(For the interrupt source specified as a fast interrupt, as well as setting the fast interrupt set register (FIR), the interrupt priority level (IPR_m) should be set above the level set by IPL in the PSW of the CPU.)

Interrupt requests through the IRQ pins that do not satisfy the above conditions are not detected while the clock is stopped in software standby mode.

- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

11.7 Usage Notes

Notes on Communication Using DTC or DMACA Transfer

In the RX62N and RX621 Groups, be careful of using the communication functions using DTC or DMACA transfer. When an interrupt request is generated while the IRI.IR flag is set to 1, re-generated interrupt request is ignored. If the DMACA or DTC is the request destination for the interrupt, the timing of the IR flag clear depends on the DMACA/DTC transfer setting and number of times of transfer. Therefore, a transfer request may be lost depending on the combinations of the settings. When communication functions (SCI, RIIC, RSPI, USB) and DTC or DMACA are combined, consider the following notes.

(1) Conditions of Transfer Request Loss during Communication using DTC or DMACA Transfer

Table 11.6 shows the combinations of the DTC/DMACA function which require special care.

Table 11.6 Combinations of the DTC/DMACA Function which Require Special Care

	Chain Transfer Used or Not Used	Communication Interrupts to CPU Issued or Not Issued after Each Transfer*1 (DISEL Setting)	DMAC/DTC Transfer Request Loss
DMACA	— (Chain transfer not provided)	DISEL = 0	During transmission: Not occur
		DISEL = 1	During reception: Might occur
DTC	Chain transfer not used	DISEL = 0 (Transfer counter value > 0)	Might occur
		DISEL = 0 (Transfer counter value = 0 at the last transfer)	Not occur
		DISEL = 1	Not occur*2
	Chain transfer used	DISEL = 0 (Transfer counter value > 0 and communication register is accessed at the end of the chain)	Might occur
		DISEL = 0 (Transfer counter value = 0 at the last transfer and communication register is accessed at the end of the chain)	Not occur
		DISEL = 1	Not occur*2
		DISEL = 1	Might occur

Note 1. Communication interrupts include: transmit data empty and receive data full interrupts from SCI, RIIC, and RSPI, and DMA transfer request from USB.

Note 2. If the IRI.IR flag is cleared too late for the transfer request of the next packet to be transmitted/received, the same problem may occur as with the case in DISEL = 1.

(2) Notes on Reception using DMACA Transfer when DISEL = 0

The IR flag is automatically cleared when the transfer source (received data) is read and the data is written to the transfer destination. A transfer request will be lost if the next transfer request is issued before the IR flag is automatically cleared.

Therefore, the DMACA should be used so that it assures the interval between reception from communication functions long enough to transfer data, or the DTC which has no possibility of causing a problem should be used.

When receiving data from the USB, the DTC should be used.

(3) Notes on Communication using DMACA Transfer when DISEL = 1

When the DMACA is used with the DISEL bit set to 1, a transfer request from the communication functions will be lost. Do not use the DMACA with the DISEL bit set to 1 for communicating with the communication functions.

When the feature provided by setting the DISEL bit to 1 is necessary, use the DTC. (See the notes below, (4) Notes on Communication using DTC Transfer when DISEL = 1.)

(4) Notes on Communication using DTC Transfer when DISEL = 1

The IR flag is automatically cleared on interrupt acceptance by the CPU after data transfer (DTC operation). A transfer request will be lost if the next transfer request is issued before the IR flag is automatically cleared. When DISEL is set to 1, a CPU interrupt is always generated: therefore, take the preventive measures through software shown below.

(5) Flowchart of Software Preventive Measures (for SCI, RIIC, RSPI)

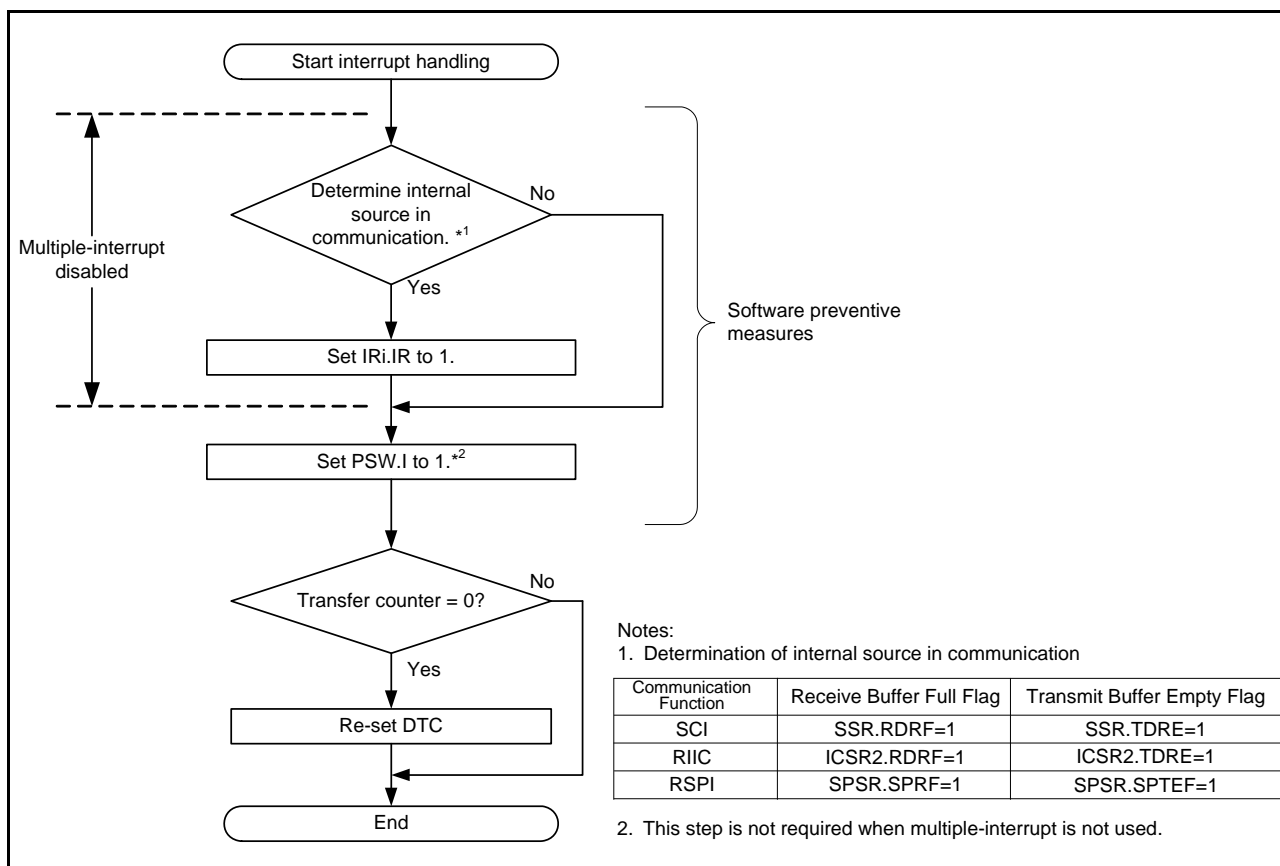


Figure 11.7 Flowchart of Software Preventive Measures (for SCI, RIIC, RSPI)

(6) Flowchart of Software Preventive Measures (for USB)

If the following conditions are all satisfied, software measures are not necessary.

- The DTC is used in block transfer mode.
- The DISEL bit is set to 0.
- Chain transfer is not used.
- The same value is specified for the DTC and USB as the number of bytes to be transmitted or received.
- The USB is prevented from issuing a DTC transfer request after the last transfer before the start of CPU interrupt handling.

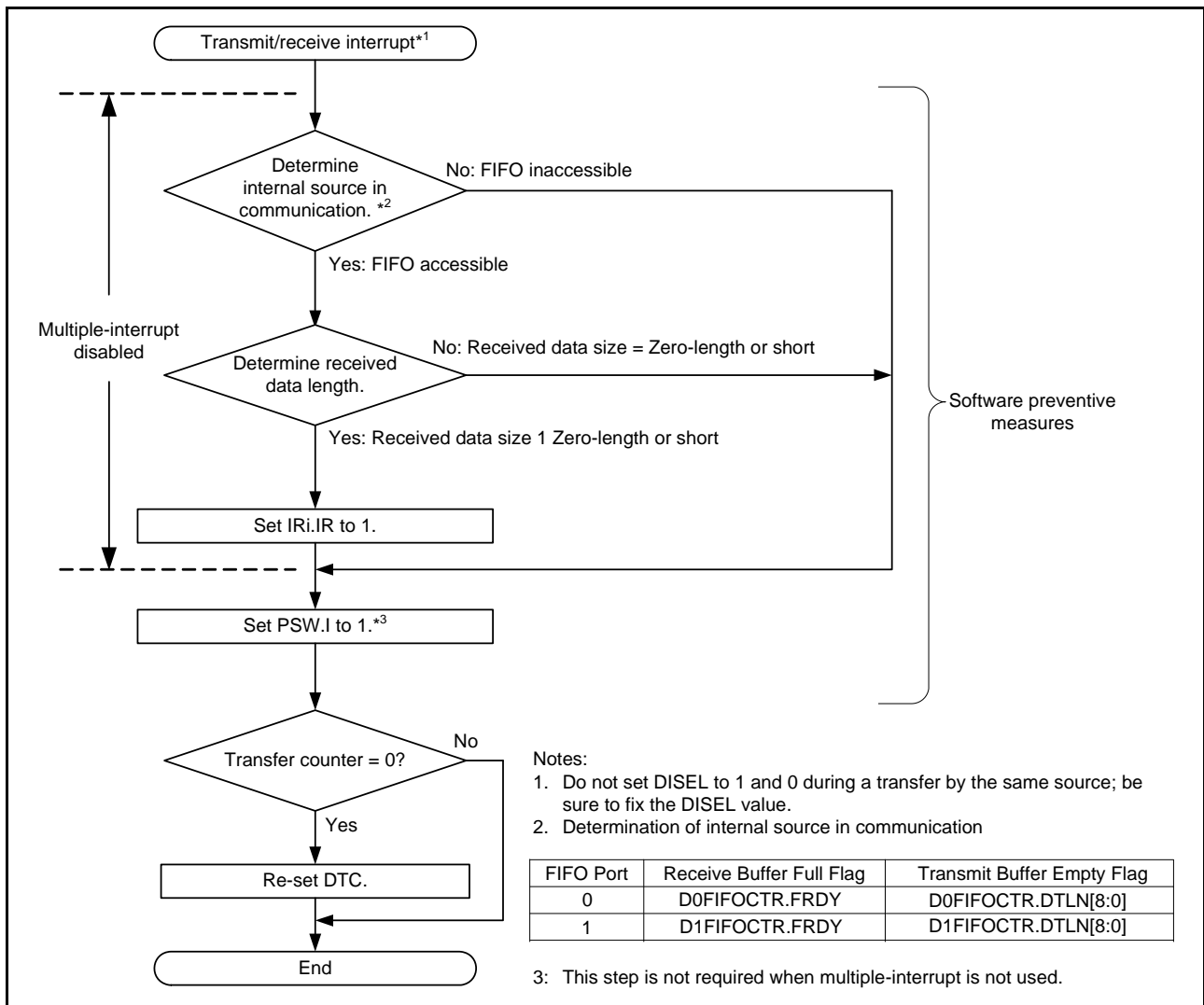


Figure 11.8 Flowchart of Software Preventive Measures (for USB)

12. Buses

12.1 Overview

Table 12.1 lists the bus specifications, figure 12.1 shows the bus configuration, and table 12.2 shows the addresses assigned for each bus.

Table 12.1 Bus Specifications

Bus Type		Description
CPU bus	Instruction bus	<ul style="list-style-type: none"> Connected to the CPU (for instructions) Connected to on-chip memory (on-chip RAM, on-chip ROM) Operates in synchronization with the system clock (ICLK)
	Operand bus	<ul style="list-style-type: none"> Connected to the CPU (for operands) Connected to on-chip memory (on-chip RAM, on-chip ROM) Operates in synchronization with the system clock (ICLK)
Memory bus	Memory bus 1	<ul style="list-style-type: none"> Connected to on-chip RAM
	Memory bus 2	<ul style="list-style-type: none"> Connected to on-chip ROM
Internal main bus	Internal main bus 1	<ul style="list-style-type: none"> Connected to the CPU Operates in synchronization with the system clock (ICLK)
	Internal main bus 2	<ul style="list-style-type: none"> Connected to the DMACA, DTC, and EDMAC Connected to on-chip memory (on-chip RAM, on-chip ROM) Operates in synchronization with the system clock (ICLK)
Internal peripheral bus	Internal peripheral bus 1	<ul style="list-style-type: none"> Connected to peripheral modules Operates in synchronization with the system clock (ICLK)
	Internal peripheral bus 2	<ul style="list-style-type: none"> Connected to peripheral modules, on-chip ROM (for programming and erasure), and data-flash memory Operates in synchronization with the peripheral-module clock (PCLK)
	Internal peripheral bus 3	<ul style="list-style-type: none"> Connected to peripheral modules (USB) Operates in synchronization with the peripheral-module clock (PCLK)
	Internal peripheral bus 4	<ul style="list-style-type: none"> Connected to peripheral modules (EDMAC and ETHERC) Operates in synchronization with the system clock (ICLK)
	Internal peripheral bus 5	<ul style="list-style-type: none"> Connected to peripheral modules Operates in synchronization with the system clock (ICLK)
	Internal peripheral bus 6	<ul style="list-style-type: none"> Connected to on-chip ROM (for programming and erasure) and data-flash memory Operates in synchronization with the peripheral-module clock (PCLK)
External bus	CS area	<ul style="list-style-type: none"> Connected to the external devices Operates in synchronization with the external-bus clock (BCLK)
	SDRAM area	<ul style="list-style-type: none"> Connected to the SDRAM Operates in synchronization with the SDRAM clock (SDCLK)

[Legend]

BCLK (external-bus clock):	The CSC (CS area controller) and the EXDMAC operate in synchronization with the BCLK.
SDCLK (SDRAM clock):	The SDRAMC (SDRAM area controller) operates in synchronization with the SDCLK.
BCLK pin output:	The frequency is the same as the BCLK as default. 1/2 BCLK can be supplied by setting the BCLK pin output select bit (BCLKDIV) in the system clock control register (SCKCR). For details, see section 8, Clock Generation Circuit.

Note: The BCLK and the SDCLK should be operated with the same frequency (50 MHz as a maximum) when the SDRAM is in use.

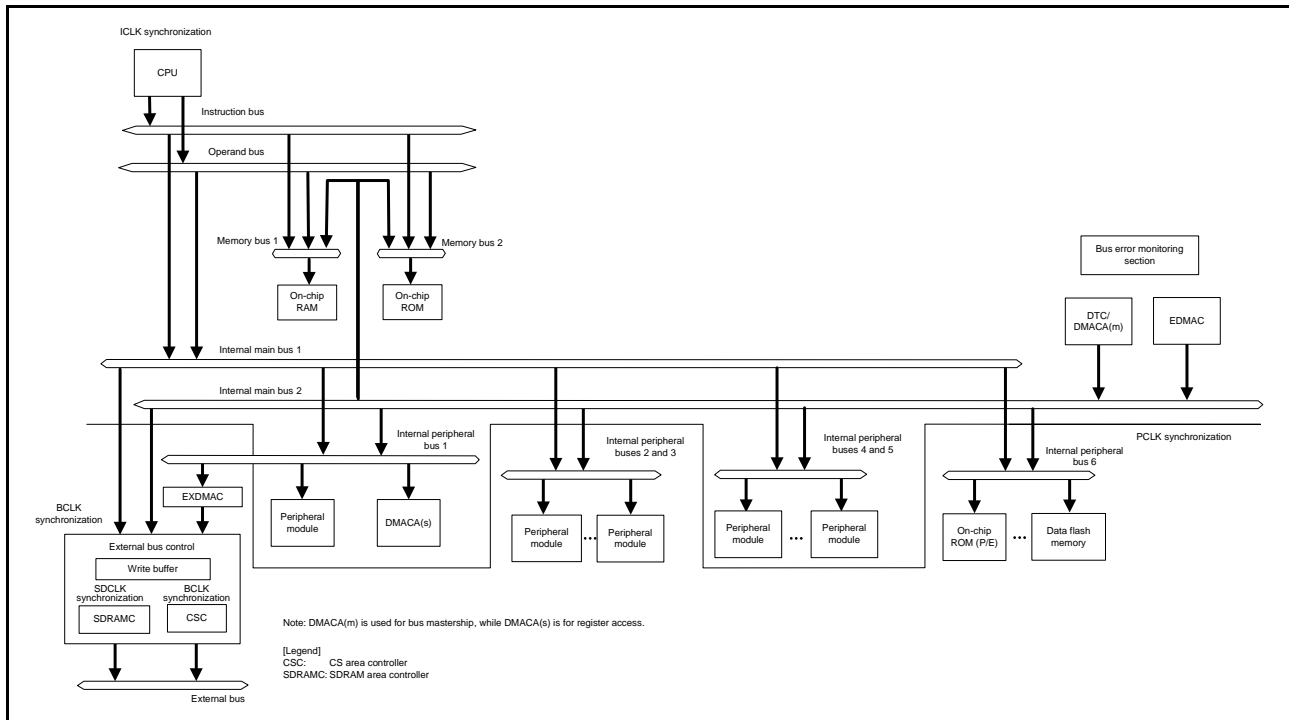


Figure 12.1 Bus Configuration

Table 12.2 Addresses Assigned for Each Bus

Address	Type of Bus	
	On-Chip ROM Mode	
	Enabled	Disabled
0000 0000h to 0007 FFFFh	Memory bus 1	
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	
000A 0000h to 000B FFFFh	Internal peripheral bus 3	
000C 0000h to 000D FFFFh	Internal peripheral bus 4	
000E 0000h to 000F FFFFh	Internal peripheral bus 5	
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	Reserved
0100 0000h to 0FFF 7FFFh	External bus	
1000 0000h to 7FFF FFFFh	Reserved	
8000 0000h to FEFF FFFFh	Memory bus 2	Reserved
FF00 0000h to FFFF FFFFh		External bus

12.2 Description of Buses

12.2.1 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 on-chip RAM and on-chip 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 on-chip ROM by the CPU; programming and erasure are handled via an internal peripheral bus.

Bus requests for instruction fetching and operand access, that are not for the on-chip RAM or on-chip ROM, 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 peripheral bus 1), the bus-access operations can proceed simultaneously. For example, parallel access to on-chip ROM and on-chip RAM or to on-chip ROM and external space is possible.

12.2.2 Memory Buses

The memory buses consist of memory bus 1 and memory bus 2. On-chip RAM is connected to memory bus 1 and on-chip 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 order of priority is internal memory bus 2 then CPU bus (operand then instruction fetching).

12.2.3 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, DMACA, and EDMAC (internal main bus 2).

Bus requests for instruction fetching and operand access, that are not for the on-chip RAM or on-chip ROM, are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.

Requests for bus mastership from the DTC/DMACA and EDMAC are arbitrated by internal main bus 2. The order of priority is EDMAC and then DTC/DMACA as shown in table 12.3.

Between the DTC and DMACA, only the one that accepted the activation request issues the bus mastership request. The priority order of activation requests between the DTC and DMACA is DMAC0, DMAC1, DMAC2, DMAC3, and then DTC.

If the CPU and another bus master are requesting access to different buses (on-chip memory, internal peripheral buses 1 to 6, and external bus), the respective bus-access operations can proceed simultaneously.

The order of priority is internal main bus 2 then internal main bus 1. 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. 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 12.3 Order of Priority for Bus Masters

Priority	Bus Master
High	EDMAC
↑	DTC/DMACA (DMACA>DTC)
Low	CPU

12.2.4 Internal Peripheral Buses

Connection of peripheral modules to the internal peripheral buses is as described in table 12.4.

Table 12.4 Connection of Peripheral Modules to the Internal Peripheral Buses

Type of Bus	Peripheral Modules
Internal peripheral bus 1	<ul style="list-style-type: none"> • DMACA • EXDMAC • Interrupt controller • Bus error monitoring section
Internal peripheral bus 2	Peripheral modules other than those connected to internal peripheral buses 1, 3, 4, and 5
Internal peripheral bus 3	USB
Internal peripheral bus 4	EDMAC and ETHERC
Internal peripheral bus 5	Reserved area
Internal peripheral bus 6	On-chip ROM (P/E)/Data flash memory

Requests for bus mastership from the CPU (internal main bus 1) and other bus masters (internal main bus 2) are arbitrated through internal peripheral buses 1 to 6.

The order of priority is internal main bus 2 then internal main bus 1.

12.2.5 External Bus

Table 12.5 lists the specifications of the external bus.

The external bus controller arbitrates requests for bus mastership from internal main bus 1, internal main bus 2, or the EXDMAC for access to the external address space and registers (CSC and SDRAMC) of the external bus controller. However, the EXDMAC is only capable of access to the external address space.

The order of priority is EXDMAC, internal main bus 2, and then internal main bus 1.

Table 12.5 Specifications of the External Bus

Item	Description
External address space	<ul style="list-style-type: none"> An external address space is divided into eight areas (CS0 to CS7) and the SDRAM area (SDCS) for management. Chip select signals can be output for each area. An 8/16/32-bit bus space is selectable for each area. An endian mode can be specified for each area.
CS area controller	<ul style="list-style-type: none"> Recovery cycles can be inserted. Read recovery: Up to 15 cycles Write recovery: Up to 15 cycles Cycle wait function: Wait for up to 31 cycles (page access: up to 7 cycles) Wait control can be used to set up the following. Timing of assertion and negation for chip-select signals (CS0# to CS7#) The timing of assertion of the read signal (RD#) and write signals (WR#, WR0# to WR3#) The timing with which data output starts and ends Write access mode: Single write strobe mode/byte strobe mode
SDRAM area controller	<ul style="list-style-type: none"> Multiplexing output of row address/column address (8, 9, 10, or 11 bits) Self-refresh and auto-Refresh selectable CAS latency can be specified from one to three cycles
Write buffer function	<ul style="list-style-type: none"> When write data from the bus master has been written to the write buffer, write access by the bus master is completed.
Frequency	<ul style="list-style-type: none"> The CS area controller (CSC) operates in synchronization with the external-bus clock (BCLK)*. The SDRAM area controller (SDRAMC) operates in synchronization with the SDRAM clock (SDCLK).

Note: * The BCLK and the SDCLK should be operated with the same frequency when the SDRAM is in use.

Table 12.6 shows the input/output pins of the external bus.

Table 12.6 Pin Configuration of the External Bus

Pin Name	I/O	Description
A23 to A0*	Output	Address output pins
D31 to D0	I/O	Data input/output pins D31 to D0 pins are enabled when the 32-bit bus space is specified. D15 to D0 pins are enabled when the 16-bit bus space is specified. D7 to D0 pins are enabled when the 8-bit bus space is specified.
BC0#*1*2	Output	A strobe signal; during access to an external bus interface space in single write strobe mode indicates that data buses D7 to D0 are valid.
BC1#*2	Output	A strobe signal; during access to an external bus interface space in single write strobe mode indicates that data buses D15 to D8 are valid.
BC2#*2	Output	A strobe signal; during access to an external bus interface space in single write strobe mode indicates that data buses D23 to D16 are valid.
BC3#*2	Output	A strobe signal; during access to an external bus interface space in single write strobe mode indicates that data buses D31 to D24 are valid.
CS0#	Output	A chip select signal for area 0 (CS0)
CS1#	Output	A chip select signal for area 1 (CS1)
CS2#	Output	A chip select signal for area 2 (CS2)
CS3#	Output	A chip select signal for area 3 (CS3)
CS4#	Output	A chip select signal for area 4 (CS4)
CS5#	Output	A chip select signal for area 5 (CS5)
CS6#	Output	A chip select signal for area 6 (CS6)
CS7#	Output	A chip select signal for area 7 (CS7)
RD#	Output	A strobe signal indicating that reading from an external address space (CS0 to CS7) is in progress
WR#	Output	A strobe signal; during writing to an external bus interface space in single write strobe mode
WR0#	Output	A strobe signal; during writing to an external bus interface space in byte strobe mode indicates that data buses D7 to D0 are valid.
WR1#	Output	A strobe signal; during writing to an external bus interface space in byte strobe mode indicates that data buses D15 to D8 are valid.
WR2#	Output	A strobe signal; during writing to an external bus interface space in byte strobe mode indicates that data buses D23 to D16 are valid.
WR3#	Output	A strobe signal; during writing to an external bus interface space in byte strobe mode indicates that data buses D31 to D24 are valid.
WAIT#	Input	A wait request signal when accessing the external address space (CS0 to CS7) (Low: Wait request)
SDCLK	Output	SDRAM clock
CKE	Output	SDRAM clock enable signal
SDCS#	Output	SDRAM chip select signal
RAS#	Output	SDRAM low address strobe signal
CAS#	Output	SDRAM column address strobe signal
WE#	Output	SDRAM write enable signal
DQM0	Output	SDRAM I/O data mask enable signal for D7 to D0
DQM1	Output	SDRAM I/O data mask enable signal for D15 to D8
DQM2	Output	SDRAM I/O data mask enable signal for D23 to D16
DQM3	Output	SDRAM I/O data mask enable signal for D31 to D24

Note 1. The A0 and BC0# pin functions share the same pin, and either becomes effective according to the area, with the function being A0 in byte-writing mode and BC0# in single write strobe mode. However, single write strobe mode can be set only for the 16-bit and 32-bit bus spaces. In the 8-bit bus space, setting single write strobe mode is prohibited. For information on other multiplexed pin functions, see section 17, I/O Ports.

Note 2. Read/write access to the BC0# to BC3# signals is enabled.

12.2.6 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 on-chip ROM and an operand from on-chip RAM, the DMACA is able to handle transfer between a peripheral bus and the external bus at the same time.

An example of parallel operations is given in figure 12.2. In this example, the CPU is able to employ the instruction and operand buses for simultaneous access to on-chip ROM and on-chip RAM, respectively. Furthermore, the DMACA simultaneously employs internal main bus 2 for access to a peripheral bus or the external bus during access to on-chip RAM and ROM by the CPU.

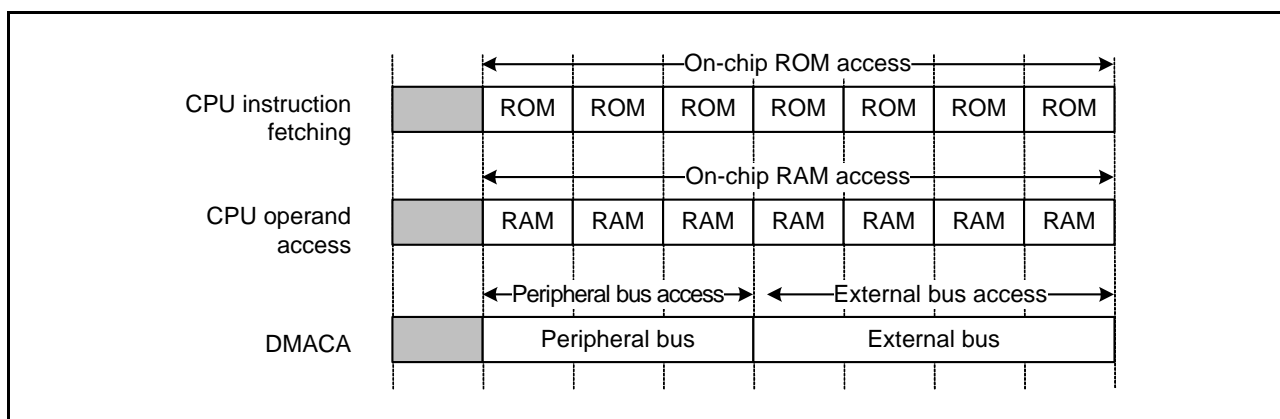


Figure 12.2 Example of Parallel Operations

12.2.7 Limitations

(1) Prohibition of Access that Spans Areas of Address Space

Single access that spans two areas of the address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.

(2) Restrictions in relation to RMPA and string-manipulation instructions

- (a) Although the external space has a per-area ending-switching facility (only for data), the allocation of data to be handled by RMPA or string-manipulation instructions to an area where the endian differs from that of the chip is prohibited, and operation is not guaranteed if this restriction is not observed. If data to be handled by RMPA or string-manipulation instructions are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.
- (b) The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

12.3 Register Descriptions

Table 12.7 lists the registers of the external bus controller.

Table 12.7 Registers of the External Bus Controller (1 / 2)

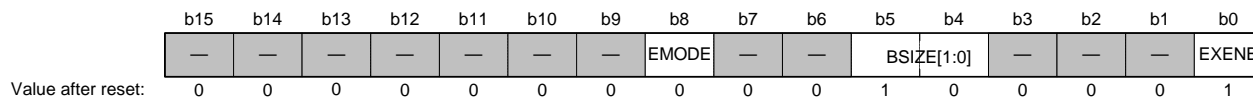
Register Name	Symbol	Value after Reset	Address	Access Size
CS0 mode register	CS0MOD	0000h	0008 3002h	16
CS0 wait control register 1	CS0WCR1	0707 0707h	0008 3004h	32
CS0 wait control register 2	CS0WCR2	0000 0007h	0008 3008h	32
CS1 mode register	CS1MOD	0000h	0008 3012h	16
CS1 wait control register 1	CS1WCR1	0707 0707h	0008 3014h	32
CS1 wait control register 2	CS1WCR2	0000 0007h	0008 3018h	32
CS2 mode register	CS2MOD	0000h	0008 3022h	16
CS2 wait control register 1	CS2WCR1	0707 0707h	0008 3024h	32
CS2 wait control register 2	CS2WCR2	0000 0007h	0008 3028h	32
CS3 mode register	CS3MOD	0000h	0008 3032h	16
CS3 wait control register 1	CS3WCR1	0707 0707h	0008 3034h	32
CS3 wait control register 2	CS3WCR2	0000 0007h	0008 3038h	32
CS4 mode register	CS4MOD	0000h	0008 3042h	16
CS4 wait control register 1	CS4WCR1	0707 0707h	0008 3044h	32
CS4 wait control register 2	CS4WCR2	0000 0007h	0008 3048h	32
CS5 mode register	CS5MOD	0000h	0008 3052h	16
CS5 wait control register 1	CS5WCR1	0707 0707h	0008 3054h	32
CS5 wait control register 2	CS5WCR2	0000 0007h	0008 3058h	32
CS6 mode register	CS6MOD	0000h	0008 3062h	16
CS6 wait control register 1	CS6WCR1	0707 0707h	0008 3064h	32
CS6 wait control register 2	CS6WCR2	0000 0007h	0008 3068h	32
CS7 mode register	CS7MOD	0000h	0008 3072h	16
CS7 wait control register 1	CS7WCR1	0707 0707h	0008 3074h	32
CS7 wait control register 2	CS7WCR2	0000 0007h	0008 3078h	32
CS0 control register	CS0CR	0021h	0008 3802h	16
CS0 recovery cycle register	CS0REC	0000h	0008 380Ah	16
CS1 control register	CS1CR	0000h	0008 3812h	16
CS1 recovery cycle register	CS1REC	0000h	0008 381Ah	16
CS2 control register	CS2CR	0000h	0008 3822h	16
CS2 recovery cycle register	CS2REC	0000h	0008 382Ah	16
CS3 control register	CS3CR	0000h	0008 3832h	16
CS3 recovery cycle register	CS3REC	0000h	0008 383Ah	16
CS4 control register	CS4CR	0000h	0008 3842h	16
CS4 recovery cycle register	CS4REC	0000h	0008 384Ah	16
CS5 control register	CS5CR	0000h	0008 3852h	16
CS5 recovery cycle register	CS5REC	0000h	0008 385Ah	16
CS6 control register	CS6CR	0000h	0008 3862h	16
CS6 recovery cycle register	CS6REC	0000h	0008 386Ah	16
CS7 control register	CS7CR	0000h	0008 3872h	16
CS7 recovery cycle register	CS7REC	0000h	0008 387Ah	16
SDC control register	SDCCR	00h	0008 3C00h	8
SDC mode register	SDCMOD	00h	0008 3C01h	8

Table 12.7 Registers of the External Bus Controller (2 / 2)

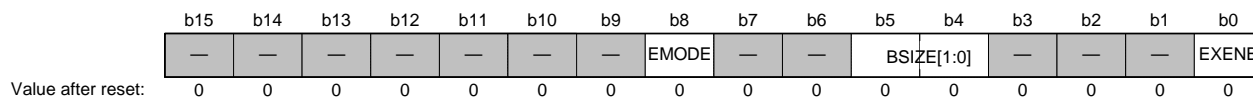
Register Name	Symbol	Value after Reset	Address	Access Size
SDRAM access mode register	SDAMOD	00h	0008 3C02h	8
SDRAM self-refresh control register	SDSELF	00h	0008 3C10h	8
SDRAM refresh control register	SDRFCR	0001h	0008 3C14h	16
SDRAM auto-refresh control register	SDRFEN	00h	0008 3C16h	8
SDRAM initialization sequence control register	SDICR	00h	0008 3C20h	8
SDRAM initialization register	SDIR	0010h	0008 3C24h	16
SDRAM address register	SDADR	00h	0008 3C40h	8
SDRAM timing register	SDTR	0000 0002h	0008 3C44h	32
SDRAM mode register	SDMOD	0000h	0008 3C48h	16
SDRAM status register	SDSR	00h	0008 3C50h	8
Bus error status clear register	BERCLR	00h	0008 1300h	8
Bus error monitoring enable register	BEREN	00h	0008 1304h	8
Bus error status register 1	BERSR1	00h	0008 1308h	8
Bus error status register 2	BERSR2	0000h	0008 130Ah	16

12.3.1 CSn Control Register (CSnCR) (n = 0 to 7)

Address: 0008 3802h (CS0CR)



Addresses: 0008 3812h to 0008 3872h (CS1CR to CS7CR)



Bit	Symbol	Bit Name	Description	R/W
b0	EXENB	Operation Enable	0: Operation is disabled 1: Operation is enabled	R/W*1
b3 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b5, b4	BSIZE[1:0]	External Bus Width Select	b5 b4 0 0: A 16-bit bus space is selected*3 0 1: A 32-bit bus space is selected*4 1 0: An 8-bit bus space is selected 1 1: Setting prohibited	R/W*2
b7, b6	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b8	EMODE	Endian Mode	0: Endian of area n (n = 0 to 7) is the same as the endian of operating mode. 1: Endian of area n (n = 0 to 7) is not the endian of operating mode.	R/W
b15 to b9	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note 1. The initial value of the EXENB bit in CS0CR is 1 and that in CSnCR (n = 1 to 7) is 0.

Note 2. The initial value of the BSIZE[1:0] bits in CS0CR is 10b.

Note 3. The 85-pin TFLGA does not support the 16-bit bus space.

Note 4. The 145-pin TFLGA, 144-pin LQFP, 100-pin LQFP, and 85-pin TFLGA do not support the 32-bit bus space.

CSnCR is used to set enabling/disabling of the operation, data bus width, and endian for each area in the external address space.

EXENB Bit (Operation Enable)

This bit enables or disables operation of the respective CS areas.

After this LSI is reset, operation is enabled (EXENB = 1) only for area 0; operation in other areas is disabled (EXENB = 0).

An attempt at access to an area for which operation has been disabled does not lead to access via the external bus.

However, if the illegal address access detection enable bit in the bus error monitoring enable register has been set to enable detection (BEREN.IGAEN = 1), such an attempt will lead to an illegal-access error.

BSIZE[1:0] Bits (External Bus Width Select)

These bits specify the data bus width of each area.

The data bus width of area 0 after a reset is 8 bits.

EMODE Bit (Endian Mode)

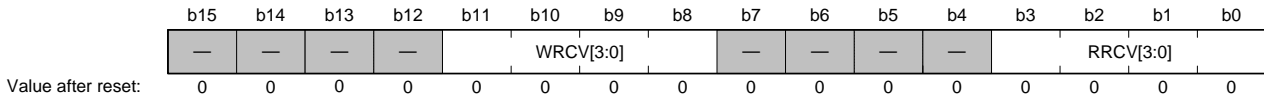
This bit specifies the endian of each area.

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area.

The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

12.3.2 CSn Recovery Cycle Register (CSnREC) (n = 0 to 7)

Addresses: 0008 380Ah to 0008 387Ah



Bit	Symbol	Bit Name	Description	R/W																																																			
b3 to b0	RRCV[3:0]	Read Recovery	<table border="0"> <tr> <td>b3</td> <td>b0</td> <td></td> </tr> <tr> <td>0 0 0</td> <td>0</td> <td>No recovery cycle is inserted.</td> </tr> <tr> <td>0 0 0</td> <td>1</td> <td>1 recovery cycle is inserted.</td> </tr> <tr> <td>0 0 1</td> <td>0</td> <td>2 recovery cycles are inserted.</td> </tr> <tr> <td>0 0 1</td> <td>1</td> <td>3 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 0</td> <td>0</td> <td>4 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 0</td> <td>1</td> <td>5 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 1</td> <td>0</td> <td>6 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 1</td> <td>1</td> <td>7 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 0</td> <td>0</td> <td>8 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 0</td> <td>1</td> <td>9 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 1</td> <td>0</td> <td>10 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 1</td> <td>1</td> <td>11 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 0</td> <td>0</td> <td>12 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 0</td> <td>1</td> <td>13 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 1</td> <td>0</td> <td>14 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 1</td> <td>1</td> <td>15 recovery cycles are inserted.</td> </tr> </table>	b3	b0		0 0 0	0	No recovery cycle is inserted.	0 0 0	1	1 recovery cycle is inserted.	0 0 1	0	2 recovery cycles are inserted.	0 0 1	1	3 recovery cycles are inserted.	0 1 0	0	4 recovery cycles are inserted.	0 1 0	1	5 recovery cycles are inserted.	0 1 1	0	6 recovery cycles are inserted.	0 1 1	1	7 recovery cycles are inserted.	1 0 0	0	8 recovery cycles are inserted.	1 0 0	1	9 recovery cycles are inserted.	1 0 1	0	10 recovery cycles are inserted.	1 0 1	1	11 recovery cycles are inserted.	1 1 0	0	12 recovery cycles are inserted.	1 1 0	1	13 recovery cycles are inserted.	1 1 1	0	14 recovery cycles are inserted.	1 1 1	1	15 recovery cycles are inserted.	R/W
b3	b0																																																						
0 0 0	0	No recovery cycle is inserted.																																																					
0 0 0	1	1 recovery cycle is inserted.																																																					
0 0 1	0	2 recovery cycles are inserted.																																																					
0 0 1	1	3 recovery cycles are inserted.																																																					
0 1 0	0	4 recovery cycles are inserted.																																																					
0 1 0	1	5 recovery cycles are inserted.																																																					
0 1 1	0	6 recovery cycles are inserted.																																																					
0 1 1	1	7 recovery cycles are inserted.																																																					
1 0 0	0	8 recovery cycles are inserted.																																																					
1 0 0	1	9 recovery cycles are inserted.																																																					
1 0 1	0	10 recovery cycles are inserted.																																																					
1 0 1	1	11 recovery cycles are inserted.																																																					
1 1 0	0	12 recovery cycles are inserted.																																																					
1 1 0	1	13 recovery cycles are inserted.																																																					
1 1 1	0	14 recovery cycles are inserted.																																																					
1 1 1	1	15 recovery cycles are inserted.																																																					
b7 to b4	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W																																																			
b11 to b8	WRCV[3:0]	Write Recovery	<table border="0"> <tr> <td>b11</td> <td>b8</td> <td></td> </tr> <tr> <td>0 0 0</td> <td>0</td> <td>No recovery cycle is inserted.</td> </tr> <tr> <td>0 0 0</td> <td>1</td> <td>1 recovery cycle is inserted.</td> </tr> <tr> <td>0 0 1</td> <td>0</td> <td>2 recovery cycles are inserted.</td> </tr> <tr> <td>0 0 1</td> <td>1</td> <td>3 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 0</td> <td>0</td> <td>4 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 0</td> <td>1</td> <td>5 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 1</td> <td>0</td> <td>6 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 1</td> <td>1</td> <td>7 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 0</td> <td>0</td> <td>8 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 0</td> <td>1</td> <td>9 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 1</td> <td>0</td> <td>10 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 1</td> <td>1</td> <td>11 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 0</td> <td>0</td> <td>12 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 0</td> <td>1</td> <td>13 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 1</td> <td>0</td> <td>14 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 1</td> <td>1</td> <td>15 recovery cycles are inserted.</td> </tr> </table>	b11	b8		0 0 0	0	No recovery cycle is inserted.	0 0 0	1	1 recovery cycle is inserted.	0 0 1	0	2 recovery cycles are inserted.	0 0 1	1	3 recovery cycles are inserted.	0 1 0	0	4 recovery cycles are inserted.	0 1 0	1	5 recovery cycles are inserted.	0 1 1	0	6 recovery cycles are inserted.	0 1 1	1	7 recovery cycles are inserted.	1 0 0	0	8 recovery cycles are inserted.	1 0 0	1	9 recovery cycles are inserted.	1 0 1	0	10 recovery cycles are inserted.	1 0 1	1	11 recovery cycles are inserted.	1 1 0	0	12 recovery cycles are inserted.	1 1 0	1	13 recovery cycles are inserted.	1 1 1	0	14 recovery cycles are inserted.	1 1 1	1	15 recovery cycles are inserted.	R/W
b11	b8																																																						
0 0 0	0	No recovery cycle is inserted.																																																					
0 0 0	1	1 recovery cycle is inserted.																																																					
0 0 1	0	2 recovery cycles are inserted.																																																					
0 0 1	1	3 recovery cycles are inserted.																																																					
0 1 0	0	4 recovery cycles are inserted.																																																					
0 1 0	1	5 recovery cycles are inserted.																																																					
0 1 1	0	6 recovery cycles are inserted.																																																					
0 1 1	1	7 recovery cycles are inserted.																																																					
1 0 0	0	8 recovery cycles are inserted.																																																					
1 0 0	1	9 recovery cycles are inserted.																																																					
1 0 1	0	10 recovery cycles are inserted.																																																					
1 0 1	1	11 recovery cycles are inserted.																																																					
1 1 0	0	12 recovery cycles are inserted.																																																					
1 1 0	1	13 recovery cycles are inserted.																																																					
1 1 1	0	14 recovery cycles are inserted.																																																					
1 1 1	1	15 recovery cycles are inserted.																																																					
b15 to b12	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W																																																			

CSnREC is used to set the number of recovery cycles to be inserted after a write access and read access of each area in the external address space.

RRCV[3:0] Bits (Read Recovery)

These bits specify the number of recovery cycles to be inserted after a read access to the external bus.

When a value except 0000b is written to these bits, one to 15 recovery cycles are inserted in the following cases.

- When a write access is made to the external bus after a read access to the external bus (Recovery cycles are also inserted when consecutive accesses are made in the same area.)
- When a read access is made to another area after a read access to the external bus (No recovery cycle is inserted when consecutive accesses are made in the same area.)

WRCV[3:0] Bits (Write Recovery)

These bits specify the number of recovery cycles to be inserted after a write access to the external bus.

When a value except 0000b is written to these bits, one to 15 recovery cycles are inserted in the following cases.

- When a read access is made to the external bus after a write access to the external bus
(Recovery cycles are also inserted when consecutive accesses are made in the same area.)

No recovery cycle is inserted during a write access after a write access.

Table 12.8 Insertion of Recovery Cycles

Access Type	External Address	
	Space	Insertion of Recovery Cycles
Read access after write access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted.
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted.
Write access after write access	Same area	No recovery cycle is inserted.
	Different area	No recovery cycle is inserted.
Write access after read access	Same area	Recovery cycles specified by the RRCV[3:0] bits are inserted.
	Different area	Recovery cycles specified by the RRCV[3:0] bits are inserted.
Read access after read access	Same area	No recovery cycle is inserted.
	Different area	Recovery cycles specified by the RRCV[3:0] bits are inserted.

12.3.3 CSn Mode Register (CSnMOD) (n = 0 to 7)

Addresses: 0008 3002h to 0008 3072h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PRMOD	—	—	—	—	—	PWENB	PRENB	—	—	—	—	EWENB	—	—	WRMOD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WRMOD	Write Access Mode Select	0: Byte strobe mode 1: Single write strobe mode	R/W
b2, b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b3	EWENB	External Wait Enable	0: External wait is disabled 1: External wait is enabled	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b8	PRENB	Page Read Access Enable	0: Page read access is disabled 1: Page read access is enabled	R/W
b9	PWENB	Page Write Access Enable	0: Page write access is disabled 1: Page write access is enabled	R/W
b14 to b10	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b15	PRMOD	Page Read Access Mode Select	0: Normal access compatible mode 1: External data read continuous assertion mode	R/W

CSnMOD is used to set access modes of each area in the external address space.

WRMOD Bit (Write Access Mode Select)

This bit selects a write access operating mode.

Writing 0 to this bit selects byte strobe mode where data write operation is controlled by the WRn# (n = 0 to 3) signal corresponding to the respective byte positions.

Writing 1 to this bit selects single write strobe mode where data write operation is controlled by the BCn# (n = 0 to 3) signal and the WR# signal corresponding to respective byte positions.

Note: When the A0 and BC0# pin functions share the same pin, single write strobe mode can be set only for the 16-bit and 32-bit bus spaces. In the 8-bit bus space, setting single write strobe mode is prohibited.

Table 12.9 Valid and Invalid Control Signals in Write Access

Write Access Mode	Data Write Signal					Byte Control Signal			
	WR3#	WR2#	WR1#	WR0#	WR#	BC3#	BC2#	BC1#	BC0#
Byte strobe mode	Enabled	Enabled	Enabled	Enabled	Disabled	Disabled	Disabled	Disabled	Disabled
Single write strobe mode	Disabled	Disabled	Disabled	Disabled	Enabled	Enabled	Enabled	Enabled	Enabled

EWENB Bit (External Wait Enable)

This bit enables or disables external wait.

Writing 1 to this bit selects external wait and allows control of the number of waits in each cycle with the WAIT# signal. In this state, wait cycles are inserted while the WAIT# signal is at the low level.

Writing 0 to this bit disables the WAIT# signal.

PRENB Bit (Page Read Access Enable)

This bit enables or disables page read accesses.

PWENB Bit (Page Write Access Enable)

This bit enables or disables page write accesses.

PRMOD Bit (Page Read Access Mode Select)

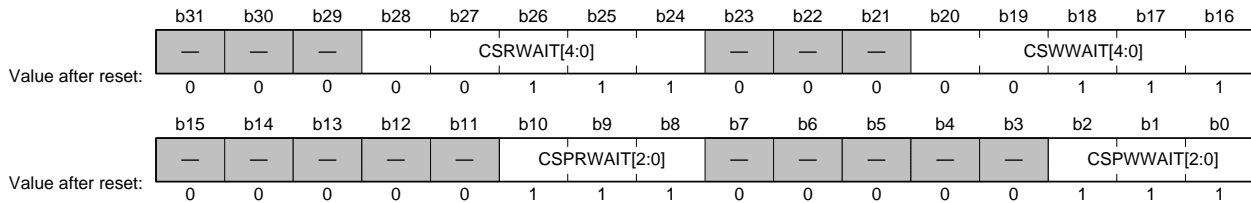
This bit selects a page read access operating mode.

Writing 0 to this bit selects normal access compatible mode where the RD# signal is negated and RD assert wait is inserted each time a piece of data is read. However, when there is no RD assert wait, the RD# signal is negated only in the final transfer of the external bus access.

Writing 1 to this bit selects external data read continuous assertion mode where RD assert wait is inserted and the RD# signal is continuously asserted during this time period.

12.3.4 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 7)

Addresses: 0008 3004h to 0008 3074h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CSPWWAIT[2:0]	Page Write Cycle Wait Select*1	b2 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b10 to b8	CSPRWAIT[2:0]	Page Read Cycle Wait Select*2	b10 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b15 to b11	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b20 to b16	CSWWAIT[4:0]	Normal Write Cycle Wait Select	b20 b16 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles is inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles is inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles is inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles is inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles is inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles is inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles is inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles is inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles is inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles is inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles is inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles is inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles is inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles is inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles is inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles is inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles is inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles is inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles is inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles is inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles is inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles is inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles is inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles is inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles is inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles is inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles is inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles is inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles is inserted.	R/W

Bit	Symbol	Bit Name	Description	R/W																																																																																																			
b23 to b21	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W																																																																																																			
b28 to b24	CSRWAIT[4:0]	Normal Read Cycle Wait Select	<table border="0"> <tr> <td>b28</td> <td>b24</td> <td></td> </tr> <tr> <td>0 0 0 0</td> <td>0 0</td> <td>0: No wait is inserted.</td> </tr> <tr> <td>0 0 0 0</td> <td>0 1</td> <td>1: Wait with a length of 1 clock cycle is inserted.</td> </tr> <tr> <td>0 0 0 1</td> <td>0 0</td> <td>0: Wait with a length of 2 clock cycles is inserted.</td> </tr> <tr> <td>0 0 0 1</td> <td>0 1</td> <td>1: Wait with a length of 3 clock cycles is inserted.</td> </tr> <tr> <td>0 0 1 0</td> <td>0 0</td> <td>0: Wait with a length of 4 clock cycles is inserted.</td> </tr> <tr> <td>0 0 1 0</td> <td>0 1</td> <td>1: Wait with a length of 5 clock cycles is inserted.</td> </tr> <tr> <td>0 0 1 1</td> <td>0 0</td> <td>0: Wait with a length of 6 clock cycles is inserted.</td> </tr> <tr> <td>0 0 1 1</td> <td>0 1</td> <td>1: Wait with a length of 7 clock cycles is inserted.</td> </tr> <tr> <td>0 1 0 0</td> <td>0 0</td> <td>0: Wait with a length of 8 clock cycles is inserted.</td> </tr> <tr> <td>0 1 0 0</td> <td>0 1</td> <td>1: Wait with a length of 9 clock cycles is inserted.</td> </tr> <tr> <td>0 1 0 1</td> <td>0 0</td> <td>0: Wait with a length of 10 clock cycles is inserted.</td> </tr> <tr> <td>0 1 0 1</td> <td>0 1</td> <td>1: Wait with a length of 11 clock cycles is inserted.</td> </tr> <tr> <td>0 1 1 0</td> <td>0 0</td> <td>0: Wait with a length of 12 clock cycles is inserted.</td> </tr> <tr> <td>0 1 1 0</td> <td>0 1</td> <td>1: Wait with a length of 13 clock cycles is inserted.</td> </tr> <tr> <td>0 1 1 1</td> <td>0 0</td> <td>0: Wait with a length of 14 clock cycles is inserted.</td> </tr> <tr> <td>0 1 1 1</td> <td>0 1</td> <td>1: Wait with a length of 15 clock cycles is inserted.</td> </tr> <tr> <td>1 0 0 0</td> <td>0 0</td> <td>0: Wait with a length of 16 clock cycles is inserted.</td> </tr> <tr> <td>1 0 0 0</td> <td>0 1</td> <td>1: Wait with a length of 17 clock cycles is inserted.</td> </tr> <tr> <td>1 0 0 1</td> <td>0 0</td> <td>0: Wait with a length of 18 clock cycles is inserted.</td> </tr> <tr> <td>1 0 0 1</td> <td>0 1</td> <td>1: Wait with a length of 19 clock cycles is inserted.</td> </tr> <tr> <td>1 0 1 0</td> <td>0 0</td> <td>0: Wait with a length of 20 clock cycles is inserted.</td> </tr> <tr> <td>1 0 1 0</td> <td>0 1</td> <td>1: Wait with a length of 21 clock cycles is inserted.</td> </tr> <tr> <td>1 0 1 1</td> <td>0 0</td> <td>0: Wait with a length of 22 clock cycles is inserted.</td> </tr> <tr> <td>1 0 1 1</td> <td>0 1</td> <td>1: Wait with a length of 23 clock cycles is inserted.</td> </tr> <tr> <td>1 1 0 0</td> <td>0 0</td> <td>0: Wait with a length of 24 clock cycles is inserted.</td> </tr> <tr> <td>1 1 0 0</td> <td>0 1</td> <td>1: Wait with a length of 25 clock cycles is inserted.</td> </tr> <tr> <td>1 1 0 1</td> <td>0 0</td> <td>0: Wait with a length of 26 clock cycles is inserted.</td> </tr> <tr> <td>1 1 0 1</td> <td>0 1</td> <td>1: Wait with a length of 27 clock cycles is inserted.</td> </tr> <tr> <td>1 1 1 0</td> <td>0 0</td> <td>0: Wait with a length of 28 clock cycles is inserted.</td> </tr> <tr> <td>1 1 1 0</td> <td>0 1</td> <td>1: Wait with a length of 29 clock cycles is inserted.</td> </tr> <tr> <td>1 1 1 1</td> <td>0 0</td> <td>0: Wait with a length of 30 clock cycles is inserted.</td> </tr> <tr> <td>1 1 1 1</td> <td>0 1</td> <td>1: Wait with a length of 31 clock cycles is inserted.</td> </tr> </table>	b28	b24		0 0 0 0	0 0	0: No wait is inserted.	0 0 0 0	0 1	1: Wait with a length of 1 clock cycle is inserted.	0 0 0 1	0 0	0: Wait with a length of 2 clock cycles is inserted.	0 0 0 1	0 1	1: Wait with a length of 3 clock cycles is inserted.	0 0 1 0	0 0	0: Wait with a length of 4 clock cycles is inserted.	0 0 1 0	0 1	1: Wait with a length of 5 clock cycles is inserted.	0 0 1 1	0 0	0: Wait with a length of 6 clock cycles is inserted.	0 0 1 1	0 1	1: Wait with a length of 7 clock cycles is inserted.	0 1 0 0	0 0	0: Wait with a length of 8 clock cycles is inserted.	0 1 0 0	0 1	1: Wait with a length of 9 clock cycles is inserted.	0 1 0 1	0 0	0: Wait with a length of 10 clock cycles is inserted.	0 1 0 1	0 1	1: Wait with a length of 11 clock cycles is inserted.	0 1 1 0	0 0	0: Wait with a length of 12 clock cycles is inserted.	0 1 1 0	0 1	1: Wait with a length of 13 clock cycles is inserted.	0 1 1 1	0 0	0: Wait with a length of 14 clock cycles is inserted.	0 1 1 1	0 1	1: Wait with a length of 15 clock cycles is inserted.	1 0 0 0	0 0	0: Wait with a length of 16 clock cycles is inserted.	1 0 0 0	0 1	1: Wait with a length of 17 clock cycles is inserted.	1 0 0 1	0 0	0: Wait with a length of 18 clock cycles is inserted.	1 0 0 1	0 1	1: Wait with a length of 19 clock cycles is inserted.	1 0 1 0	0 0	0: Wait with a length of 20 clock cycles is inserted.	1 0 1 0	0 1	1: Wait with a length of 21 clock cycles is inserted.	1 0 1 1	0 0	0: Wait with a length of 22 clock cycles is inserted.	1 0 1 1	0 1	1: Wait with a length of 23 clock cycles is inserted.	1 1 0 0	0 0	0: Wait with a length of 24 clock cycles is inserted.	1 1 0 0	0 1	1: Wait with a length of 25 clock cycles is inserted.	1 1 0 1	0 0	0: Wait with a length of 26 clock cycles is inserted.	1 1 0 1	0 1	1: Wait with a length of 27 clock cycles is inserted.	1 1 1 0	0 0	0: Wait with a length of 28 clock cycles is inserted.	1 1 1 0	0 1	1: Wait with a length of 29 clock cycles is inserted.	1 1 1 1	0 0	0: Wait with a length of 30 clock cycles is inserted.	1 1 1 1	0 1	1: Wait with a length of 31 clock cycles is inserted.	R/W
b28	b24																																																																																																						
0 0 0 0	0 0	0: No wait is inserted.																																																																																																					
0 0 0 0	0 1	1: Wait with a length of 1 clock cycle is inserted.																																																																																																					
0 0 0 1	0 0	0: Wait with a length of 2 clock cycles is inserted.																																																																																																					
0 0 0 1	0 1	1: Wait with a length of 3 clock cycles is inserted.																																																																																																					
0 0 1 0	0 0	0: Wait with a length of 4 clock cycles is inserted.																																																																																																					
0 0 1 0	0 1	1: Wait with a length of 5 clock cycles is inserted.																																																																																																					
0 0 1 1	0 0	0: Wait with a length of 6 clock cycles is inserted.																																																																																																					
0 0 1 1	0 1	1: Wait with a length of 7 clock cycles is inserted.																																																																																																					
0 1 0 0	0 0	0: Wait with a length of 8 clock cycles is inserted.																																																																																																					
0 1 0 0	0 1	1: Wait with a length of 9 clock cycles is inserted.																																																																																																					
0 1 0 1	0 0	0: Wait with a length of 10 clock cycles is inserted.																																																																																																					
0 1 0 1	0 1	1: Wait with a length of 11 clock cycles is inserted.																																																																																																					
0 1 1 0	0 0	0: Wait with a length of 12 clock cycles is inserted.																																																																																																					
0 1 1 0	0 1	1: Wait with a length of 13 clock cycles is inserted.																																																																																																					
0 1 1 1	0 0	0: Wait with a length of 14 clock cycles is inserted.																																																																																																					
0 1 1 1	0 1	1: Wait with a length of 15 clock cycles is inserted.																																																																																																					
1 0 0 0	0 0	0: Wait with a length of 16 clock cycles is inserted.																																																																																																					
1 0 0 0	0 1	1: Wait with a length of 17 clock cycles is inserted.																																																																																																					
1 0 0 1	0 0	0: Wait with a length of 18 clock cycles is inserted.																																																																																																					
1 0 0 1	0 1	1: Wait with a length of 19 clock cycles is inserted.																																																																																																					
1 0 1 0	0 0	0: Wait with a length of 20 clock cycles is inserted.																																																																																																					
1 0 1 0	0 1	1: Wait with a length of 21 clock cycles is inserted.																																																																																																					
1 0 1 1	0 0	0: Wait with a length of 22 clock cycles is inserted.																																																																																																					
1 0 1 1	0 1	1: Wait with a length of 23 clock cycles is inserted.																																																																																																					
1 1 0 0	0 0	0: Wait with a length of 24 clock cycles is inserted.																																																																																																					
1 1 0 0	0 1	1: Wait with a length of 25 clock cycles is inserted.																																																																																																					
1 1 0 1	0 0	0: Wait with a length of 26 clock cycles is inserted.																																																																																																					
1 1 0 1	0 1	1: Wait with a length of 27 clock cycles is inserted.																																																																																																					
1 1 1 0	0 0	0: Wait with a length of 28 clock cycles is inserted.																																																																																																					
1 1 1 0	0 1	1: Wait with a length of 29 clock cycles is inserted.																																																																																																					
1 1 1 1	0 0	0: Wait with a length of 30 clock cycles is inserted.																																																																																																					
1 1 1 1	0 1	1: Wait with a length of 31 clock cycles is inserted.																																																																																																					
b31 to b29	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W																																																																																																			

Note 1. The CSPWAIT[2:0] value is valid only when the PWENB bit in CSnMOD is set to 1.

Note 2. The CSPRWAIT[2:0] value is valid only when the PRENB bit in CSnMOD is set to 1.

CSnWCR1 is used to select the number of wait cycles of each area in the external address space.

CSPWAIT[2:0] Bits (Page Write Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page write cycle.

This setting is enabled when the PWENB bit in CSnMOD is set to 1.

Note: Be sure to satisfy $1 \leq \text{CSnWCR2.WDON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSPWAIT}[2:0] \text{ value}$ and $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSPWAIT}[2:0] \text{ value}$.

CSPRWAIT[2:0] Bits (Page Read Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page read cycle.

This setting is enabled when the PRENB bit in CSnMOD is set to 1.

Note: Be sure to satisfy $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.RDON}[2:0] \text{ value} \leq \text{CSnWCR1.CSPRWAIT}[2:0] \text{ value}$.

CSWAIT[4:0] Bits (Normal Write Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the first access during a normal write cycle or page write cycle.

Note: Be sure to satisfy $1 \leq \text{CSnWCR2.WDON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSWWAIT}[2:0] \text{ value}$ and $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSWWAIT}[2:0] \text{ value}$.

CSRWAIT[4:0] Bits (Normal Read Cycle Wait Select)

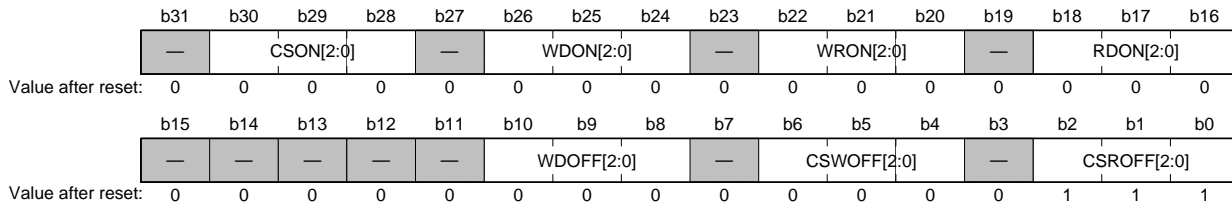
These bits specify the number of wait cycles to be inserted into the first access during a normal read cycle or page read cycle.

Note: Be sure to satisfy $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.RDON}[2:0] \text{ value} \leq \text{CSnWCR1.CSRWAIT}[4:0] \text{ value}$.

Note: Set each of these bits within a range of the restrictions described in section 12.5.6 (1) Limitations at the Time of Normal and Page Access. During the EXDMAC transfer in single address mode, set each of these bits within a range of the restrictions described in section 12.5.6 (4) Limitations on EXDMAC Single Address Transfer Mode.

12.3.5 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 7)

Addresses: 0008 3008h to 0008 3078h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CSROFF[2:0]	Read-Access CS Extension Cycle Select	b2 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b3	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W
b6 to b4	CSWOFF[2:0]	Write-Access CS Extension Cycle Select	b6 b4 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b7	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W
b10 to b8	WDOFF[2:0]	Write Data Output Extension Cycle Select	b10 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b15 to b11	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b18 to b16	RDON[2:0]	RD Assert Wait Select	b18 b16 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b19	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W
b22 to b20	WRON[2:0]	WR Assert Wait Select	b22 b20 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b23	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b26 to b24	WDON[2:0]	Write Data Output Wait Select	b26 b24 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b27	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W
b30 to b28	CSON[2:0]	CS Assert Wait Select	b30 b28 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
b31	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W

CSnWCR2 is used to select the number of wait cycles of each area in the external address space.

CSROFF[2:0] Bits (Read-Access CS Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (RD# signal negated) until the CSn# signal (n = 0 to 7) is negated in read access mode.

CSWOFF[2:0] Bits (Write-Access CS Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (WR0# to WR3#, and WR# signals negated) until the CSn# signal (n = 0 to 7) is negated in write access mode.

Note: Be sure to satisfy WDOFF[2:0] value \leq CSWOFF[2:0] value.

WDOFF[2:0] Bits (Write Data Output Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (WRn# signal (n = 0 to 3) negated) until the write data output is completed in write access mode.

Although the chip does not output write data during the EXDMAC transfer in single address mode, the specified number of write data output extension cycles is valid between split bus cycles during page access.

Note: Be sure to satisfy WDOFF[2:0] value \leq CSWOFF[2:0] value.

RDON[2:0] Bits (RD Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the RD# signal is asserted.

Note 1. For normal read access, satisfy CSnWCR2.CSON[2:0] value \leq CSnWCR2.RDON[2:0] value \leq CSnWCR1.CSRWAIT[4:0] value.

For page read access, satisfy CSnWCR2.CSON[2:0] value \leq CSnWCR2.RDON[2:0] value \leq CSnWCR1.CSPRWAIT[4:0] value.

Note 2. In addition to the above conditions, satisfy CSnWCR2.RDON[2:0] value \geq 1 for read access in the EXDMAC transfer in single address mode.

WRON[2:0] Bits (WR Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the WR0# to WR3#, and WR# signals are asserted.

Note 1. For normal write access, satisfy $1 \leq \text{CSnWCR2.WDON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSWWAIT}[4:0] \text{ value}$ and $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSWWAIT}[4:0] \text{ value}$.

For page write access, satisfy $1 \leq \text{CSnWCR2.WDON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSPWAIT}[4:0] \text{ value}$ and $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSPWAIT}[4:0] \text{ value}$.

Note 2. In addition to the above conditions, satisfy $\text{CSnWCR2.WRON}[2:0] \text{ value} \geq 1$ for write access in the EXDMAC transfer in single address mode.

WDON[2:0] Bits (Write Data Output Wait Select)

These bits specify the number of wait cycles to be inserted before the write data is output. However, the write data is only output when CS# signals are asserted.

Note: For normal write access, satisfy $1 \leq \text{CSnWCR2.WDON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSWWAIT}[4:0] \text{ value}$.

For page write access, satisfy $1 \leq \text{CSnWCR2.WDON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSPWAIT}[4:0] \text{ value}$.

CSON[2:0] Bits (CS Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the CSn# signal (n = 0 to 7) is asserted.

Note: For normal read access, satisfy $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.RDON}[2:0] \text{ value} \leq \text{CSnWCR1.CSRWAIT}[4:0] \text{ value}$.

For page read access, satisfy $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.RDON}[2:0] \text{ value} \leq \text{CSnWCR1.CSPRWAIT}[4:0] \text{ value}$.

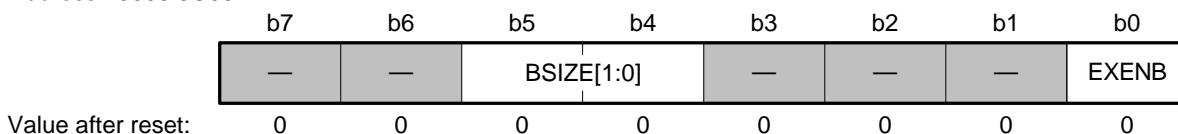
For normal write access, satisfy $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSWWAIT}[4:0] \text{ value}$.

For page write access, satisfy $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSPWAIT}[4:0] \text{ value}$.

Note: Set each of these bits within a range of the restrictions described in section 12.5.6 (1) Limitations at the Time of Normal and Page Access. During the EXDMAC transfer in single address mode, set each of these bits within a range of the restrictions described in section 12.5.6 (4) Limitations on EXDMAC Single Address Transfer Mode.

12.3.6 SDC Control Register (SDCCR)

Address: 0008 3C00h



Bit	Symbol	Bit Name	Description	R/W															
b0	EXENB	Operation Enable	0: Operation is disabled 1: Operation is enabled	R/W															
b3 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W															
b5, b4	BSIZE[1:0]	SDRAM Bus Width Select	<table border="0" style="font-size: small;"> <tr> <td style="padding-right: 10px;">b5</td> <td>b4</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0: A 16-bit bus space is selected</td> </tr> <tr> <td>0</td> <td>1</td> <td>1: A 32-bit bus space is selected</td> </tr> <tr> <td>1</td> <td>0</td> <td>0: An 8-bit bus space is selected</td> </tr> <tr> <td>1</td> <td>1</td> <td>1: Setting prohibited</td> </tr> </table>	b5	b4		0	0	0: A 16-bit bus space is selected	0	1	1: A 32-bit bus space is selected	1	0	0: An 8-bit bus space is selected	1	1	1: Setting prohibited	R/W
b5	b4																		
0	0	0: A 16-bit bus space is selected																	
0	1	1: A 32-bit bus space is selected																	
1	0	0: An 8-bit bus space is selected																	
1	1	1: Setting prohibited																	
b7, b6	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W															

SDCCR enables or disables the operation of the SDRAM address space, and sets the data bus width.

EXENB Bit (Operation Enable)

This bit enables or disables the operation of the SDRAM address space.

After reset, this bit is cleared to 0 (operation disabled).

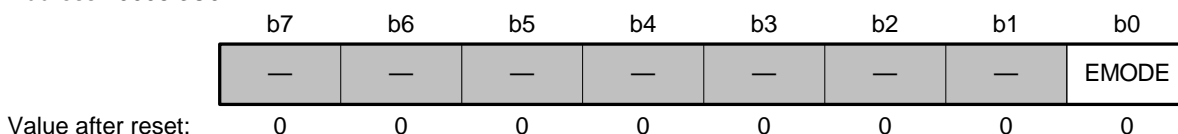
An attempt at access to an area for which operation has been disabled does not lead to SDRAM access. If the illegal address access detection enable bit (IGAEN) in the bus error monitoring enable register (BEREN) has been set to 1 (detection enabled), such an attempt will lead to a bus error.

BSIZE[1:0] Bits (Operation Enable)

These bits set the data bus width of the SDRAM address space.

12.3.7 SDC Mode Register (SDCMOD)

Address: 0008 3C01h



Bit	Symbol	Bit Name	Description	R/W
b0	EMODE	Endian Mode	0: Endian of SDRAM address space is the same as the endian of operating mode. 1: Endian of SDRAM address space is not the endian of operating mode.	R/W
B7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

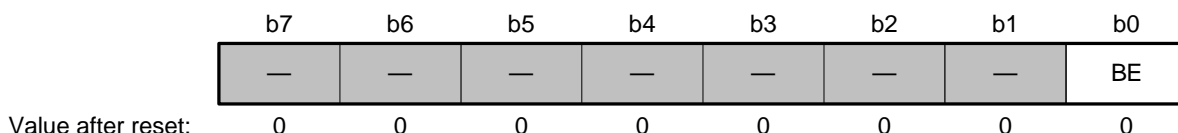
SDCMOD specifies the endian of the SDRAM address space.

EMODE Bit (Endian Mode)

This bit specifies the endian of the SDRAM address space.

12.3.8 SDRAM Access Mode Register (SDAMOD)

Address: 0008 3C02h



Bit	Symbol	Bit Name	Description	R/W
b0	BE	Continuous Access Enable	0: Continuous access is disabled 1: Continuous access is enabled	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

SDAMOD enables or disables continuous access to the SDRAM access space.

BE Bit (Continuous Access Enable)

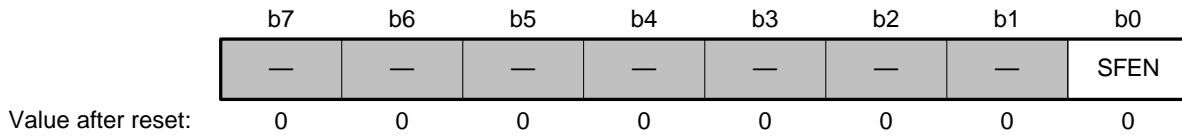
This bit enables or disables continuous access to the SDRAM access space.

Note 1. Set SDAMOD while the conditions listed in table 12.15, Conditions for Register Modification are satisfied. The operation is not guaranteed if this register is set while these conditions are not satisfied.

Note 2. When the SDRAM area is accessed from bus masters other than EXDMAC, continuous access is always disabled regardless of the setting.

12.3.9 SDRAM Self-Refresh Control Register (SDSELF)

Address: 0008 3C10h



Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b0	SFEN	SDRAM Self-Refresh Enable	0: Self-refresh is disabled 1: Self-refresh is enabled	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

SDSELF controls self-refresh operation.

SFEN Bit (SDRAM Self-Refresh Enable)

This bit controls self-refresh operation.

Setting this bit to 1 performs auto-refresh cycle operation, after which self-refresh operation begins.

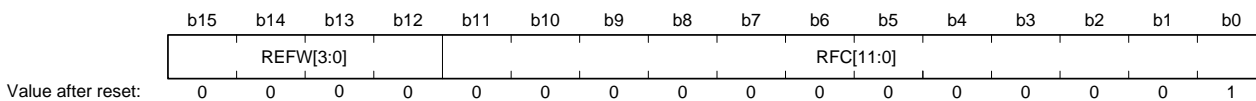
Clearing this bit to 0 ends self-refresh operation, and auto-refresh operation resumes afterwards.

If this bit was set to 1, the value written to this bit is reflected when self-refresh operation starts. If this bit was cleared to 0, the value written to this bit has already been reflected when auto-refresh operation starts following the end of self-refresh operation.

Note: Set the SFEN bit to enable/disable self-refresh operation while the conditions listed in table 12.15, Conditions for Register Modification are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

12.3.10 SDRAM Refresh Control Register (SDRFCR)

Address: 0008 3C14h



Bit	Symbol	Bit Name	Description	R/W
b11 to b0	RFC[11:0]	Auto-Refresh Request Interval Setting	b11 b0 0 0 0 0 0 0 0 0 0 0 0 0: Setting prohibited 0 0 0 0 0 0 0 0 0 0 0 1: 2 cycles 0 0 0 0 0 0 0 0 0 0 1 0: 3 cycles : 1 1 1 1 1 1 1 1 1 1 1 1: 4096 cycles	R/W
b15 to b12	REFW[3:0]	Auto-Refresh Cycle/ Self-Refresh Clearing Cycle Count Setting	b15 b12 0 0 0 0: 1 cycle 0 0 0 1: 2 cycles 0 0 1 0: 3 cycles 0 0 1 1: 4 cycles 0 1 0 0: 5 cycles 0 1 0 1: 6 cycles 0 1 1 0: 7 cycles 0 1 1 1: 8 cycles 1 0 0 0: 9 cycles 1 0 0 1: 10 cycles 1 0 1 0: 11 cycles 1 0 1 1: 12 cycles 1 1 0 0: 13 cycles 1 1 0 1: 14 cycles 1 1 1 0: 15 cycles 1 1 1 1: 16 cycles	R/W

SDRFCR is used to set the number of refresh cycles.

RFC[11:0] Bits (Auto-Refresh Request Interval Setting)

These bits specify the auto-refresh request interval.

These bits can be written to at any time, regardless of the state of the auto-refresh operation enable (RFEN) bit in SDRFEN.

If auto-refresh is enabled, the value written to these bits is reflected after the end of auto-refresh cycles. The refresh counter operates in SDCLK.

REFW[3:0] Bits (Auto-Refresh Cycle/ Self-Refresh Clearing Cycle Count Setting)

These bits specify the number of auto-refresh cycles and the number of self-refresh clearing cycles.

These bits can be written to at any time, regardless of the state of the auto-refresh operation enable (RFEN) bit in SDRFEN.

If an auto-refresh cycle is in progress, the value written to these bits while auto-refresh is enabled takes effect after the cycle completes.

Note: Auto-refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes, so the auto-refresh interval may become enlarged in some cases. Set the RFC[11:0] bits to an auto-refresh request interval value that satisfies the auto-refresh interval specification of the SDRAM being used. Furthermore, make sure to set the auto-refresh request interval to a duration longer than the auto-refresh cycle. Note that the auto-refresh interval cannot be automatically adjusted when the frequency is changed during operation; in this case, perform self-refresh operation and set the auto-refresh interval appropriate for the frequency again.

- Auto-Refresh Request Interval and RFC Set Value

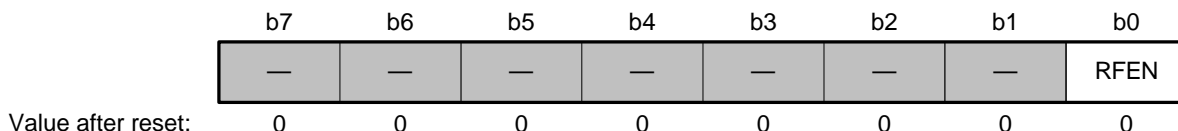
SDRAMC (SDRAM area controller) includes a 12-bit refresh counter that generates auto-refresh requests at fixed intervals. The following equation is used to calculate the set value for the RFC[11:0] bits from the auto-refresh request interval.

$$RFC = (\text{Auto-refresh request interval} / \text{SDCLK cycle}) - 1$$

Note: Auto-refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes. However, the counter value is updated regardless of whether or not the request was accepted. Note that if two or more auto-refresh requests are generated while SDRAM is being accessed, the second and subsequent requests are ignored.

12.3.11 SDRAM Auto-Refresh Control Register (SDRFEN)

Address: 0008 3C16h



Bit	Symbol	Bit Name	Description	R/W
b0	RFEN	Auto-Refresh Operation Enable	0: Auto-refresh operation is disabled 1: Auto-refresh operation is enabled	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

SDRFEN controls auto-refresh operation.

RFEN Bit (Auto-Refresh Operation Enable)

Clearing this bit to 0 while auto-refreshing is enabled causes RFEN to be cleared to 0 and auto-refresh operation to halt after the end of the auto-refresh cycle. However, if RFEN is again set to 1 before the end of the auto-refresh cycle, auto-refreshing continues and the RFEN bit is not cleared to 0.

Setting the RFEN bit to 1 while auto-refresh is disabled starts auto-refresh operation, and refresh requests are then generated at fixed intervals determined by a counter. The interval at which refresh requests are generated is determined by the value of the auto-refresh request interval setting (RFC[11:0]) bits in the SDRAM refresh control register (SDRFCR).

Refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes.

If an SDRAM access and a refresh request are generated at the same time, the refresh request takes precedence.

12.3.12 SDRAM Initialization Sequence Control Register (SDICR)

Address: 0008 3C20h



Bit	Symbol	Bit Name	Description	R/W
b0	INIRQ	Initialization Sequence Start	0: Invalid 1: Initialization sequence starts	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

SDICR controls activation of the SDRAM initialization sequence.

INIRQ Bit (Initialization Sequence Start)

Setting this bit to 1 causes the SDRAM initialization sequence to start and automatically sets the initialization status bit (INIST) in the SDRAM status register (SDSR) to 1. The INIST bit is cleared automatically after the initialization sequence ends.

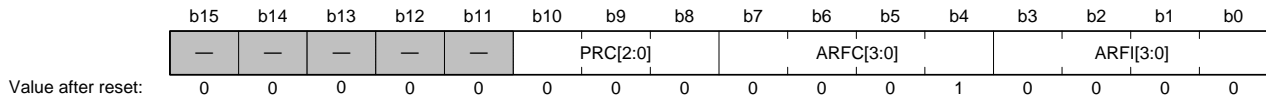
The value written to the INIRQ bit is not retained.

If access to an external address space or an external bus controller register occurs after the initialization sequence is started, the access is suspended until the initialization sequence ends.

Note: Set the INIRQ bit to start the SDRAM initialization sequence while the conditions listed in table 12.15, Conditions for Register Modification are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

12.3.13 SDRAM Initialization Register (SDIR)

Address: 0008 3C24h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ARFI[3:0]	Initialization Auto-Refresh Interval	b3 b0 0 0 0 0: 3 cycles 0 0 0 1: 4 cycles 0 0 1 0: 5 cycles 0 0 1 1: 6 cycles 0 1 0 0: 7 cycles 0 1 0 1: 8 cycles 0 1 1 0: 9 cycles 0 1 1 1: 10 cycles 1 0 0 0: 11 cycles 1 0 0 1: 12 cycles 1 0 1 0: 13 cycles 1 0 1 1: 14 cycles 1 1 0 0: 15 cycles 1 1 0 1: 16 cycles 1 1 1 0: 17 cycles 1 1 1 1: 18 cycles	R/W
b7 to b4	ARFC[3:0]	Initialization Auto-Refresh Count	b7 b4 0 0 0 0: Setting prohibited 0 0 0 1: 1 time 0 0 1 0: 2 times 0 0 1 1: 3 times 0 1 0 0: 4 times 0 1 0 1: 5 times 0 1 1 0: 6 times 0 1 1 1: 7 times 1 0 0 0: 8 times 1 0 0 1: 9 times 1 0 1 0: 10 times 1 0 1 1: 11 times 1 1 0 0: 12 times 1 1 0 1: 13 times 1 1 1 0: 14 times 1 1 1 1: 15 times	R/W
b10 to b8	PRC[2:0]	Initialization Precharge Cycle Count	b10 b8 0 0 0: 3 cycles 0 0 1: 4 cycles 0 1 0: 5 cycles 0 1 1: 6 cycles 1 0 0: 7 cycles 1 0 1: 8 cycles 1 1 0: 9 cycles 1 1 1: 10 cycles	R/W
b15 to b11	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

SDIR specifies the SDRAM initialization sequence timing.

ARFI[3:0] Bits (Initialization Auto-Refresh Interval)

These bits specify the interval at which auto-refresh commands are issued in the SDRAM initialization sequence.

ARFC[3:0] Bits (Initialization Auto-Refresh Count)

These bits specify the number of times auto-refresh is to be performed in the SDRAM initialization sequence.

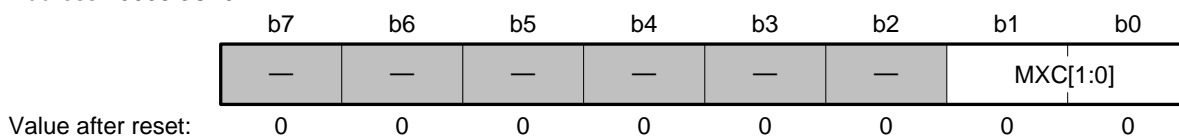
PRC[2:0] Bits (Initialization Precharge Cycle Count)

These bits specify the number of precharge cycles in the SDRAM initialization sequence.

Note: Make settings that satisfy the specifications of the connected SDRAM before starting the initialization sequence.

12.3.14 SDRAM Address Register (SDADR)

Address: 0008 3C40h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	MXC[1:0]	Address Multiplex Select	B1 b0 0 0: 8-bit shift 0 1: 9-bit shift 1 0: 10-bit shift 1 1: 11-bit shift	R/W
b7 to b2	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

SDADR select the size of the shift in address multiplexing.

MXC[1:0] Bits (Address Multiplex Select)

These bits select the size of the shift toward the lower half of the row address in row address/column address multiplexing. These bits also select the row address bits to be used for comparison in the SDRAMC continuous access operation.

For details, refer to table 12.20, Address Multiplexing.

Note: Set the SDADR register while the conditions listed in table 12.15, Conditions for Register Modification are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

12.3.15 SDRAM Timing Register (SDTR)

Address: 0008 3C44h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	RAS[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
—	—	RCD[1:0]		RP[2:0]		WR		—	—	—	—	—	CL[2:0]		
Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CL[2:0]	SDRAMC Column Latency	b2 b0 0 0 0: Setting prohibited 0 0 1: 1 cycle 0 1 0: 2 cycles 0 1 1: 3 cycles 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b8	WR	Write Recovery Interval	0: 1 cycle 1: 2 cycles	R/W
b11 to b9	RP[2:0]	Row Precharge Interval	b11 b9 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: 8 cycles	R/W
b13, b12	RCD[1:0]	Row Column Latency	b13 b12 0 0: 1 cycle 0 1: 2 cycles 1 0: 3 cycles 1 1: 4 cycles	R/W
b15, b14	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b18 to b16	RAS[2:0]	Row Active Interval	b18 b16 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: Setting prohibited	R/W
b31 to b19	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

SDTR specifies the timing for read and write accesses to SDRAM. For details, see section 12.6.12.3, Timing Register Settings and Access Timing.

CL[2:0] Bits (SDRAM Column Latency)

These bits specify the column latency of the SDRAM controller. This setting only affects the latency setting on the SDRAM controller side. To specify the column latency for externally connected SDRAM, use the separate SDRAM mode register (SDMOD), which is described below.

Note: Setting the CL[2:0] bits to 1 (CL = 1) in continuous access mode is prohibited, and operation is not guaranteed if this combination of settings is made.

WR Bit (Write Recovery Interval)

This bit specifies the interval that must elapse between the SDRAM write command (WRIT) and deactivation (PALL).

RP[2:0] Bits (Row Precharge Interval)

These bits specify the minimum number of cycles that must elapse between the SDRAM deactivation command (PALL) and the next valid command.

RCD[1:0] Bits (Row Column Latency)

These bits specify the SDRAM row column latency.

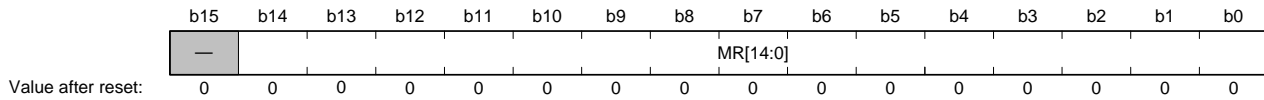
RAS[2:0] Bits (Row Active Interval)

These bits specify the minimum interval that must elapse between the SDRAM row activation command (ACTV) and deactivation (PALL). The value specified by these bits should be less than or equal to the sum of the row-column latency (RCD[1:0]) and column latency (CL[2:0]) settings.

Note: Set the SDTR register while the conditions listed in table 12.15, Conditions for Register Modification are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

12.3.16 SDRAM Mode Register (SDMOD)

Address: 0008 3C48h



Bit	Symbol	Bit Name	Description	R/W
b14 to b0	MR[14:0]	Mode Register Setting	Writing to these bits: mode register set command is issued	R/W
b15	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W

SDMOD specifies the value to be written to the SDRAM mode register. Writing to SDMOD causes a mode register set command to be issued automatically to SDRAM.

MR[14:0] Bits (Mode Register Setting)

Writing to these bits causes a mode register set command to be issued to SDRAM. The setting of the MR[14:0] bits is output to the lower bits of the address. For details, refer to section 12.6.11, Setting Mode Register.

When registers of the external address space and external bus controller are accessed after the mode register has been written to, the processing is suspended until a mode register set command is issued.

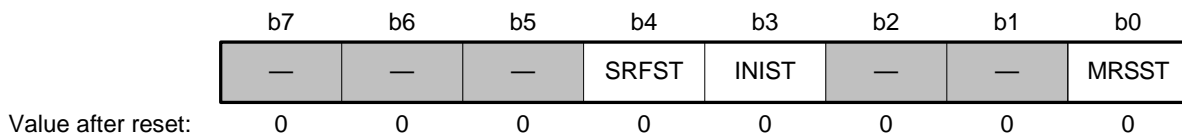
Note 1. The following points should be kept in mind regarding SDMOD mode register settings.

- Make sure to set a burst length of 1 for SDRAM. Operation cannot be guaranteed with settings other than burst length 1.
- The SDRAM column latency must match the setting of the SDRAMC column latency setting bits (CL[2:0]) in the SDRAM timing register (SDTR). Operation cannot be guaranteed if the latency settings do not agree.
- Check to make sure the status bits (SRFST, INIST, and MRSST) in the SDRAM status register (SDSR) are all 0.

Note 2. Set the SDMOD register while the conditions listed in table 12.15, Conditions for Register Modification are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

12.3.17 SDRAM Status Register (SDSR)

Address: 0008 3C50h



Bit	Symbol	Bit Name	Description	R/W
b0	MRSST	Mode Register Setting Status	0: Mode register setting not in progress 1: Mode register setting in progress	R
b2, b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b3	INIST	Initialization Status	0: Initialization sequence not in progress 1: Initialization sequence in progress	R
b4	SRFST	Self-Refresh Transition/ Recovery Status	0: Transition/recovery not in progress 1: Transition/recovery in progress	R
b7 to b5	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

SDSR indicates the status of SDRAMC operation during self-refresh, initialization sequences, and mode register setting.

MRSST Bit (Mode Register Setting Status)

When set to 1, this bit indicates that SDRAM mode register setting is in progress. If SDRS is accessed during the mode register setting operation, the CPU processing can be suspended until the setting operation ends.

INIST Bit (Initialization Status)

When set to 1, this bit indicates that the SDRAM initialization sequence is in progress. If SDRS is accessed during initialization sequence, the CPU processing can be suspended until the initialization sequence ends.

SRFST Bit (Self-Refresh Transition/Recovery Status)

When set to 1, this bit indicates that a transition to or recovery from self-refresh operation is in progress for SDRAM.

"Transition to or recovery from self-refresh operation in progress" refers to the interval from the point at which the bits listed in table 12.10 are written until the corresponding commands are issued.

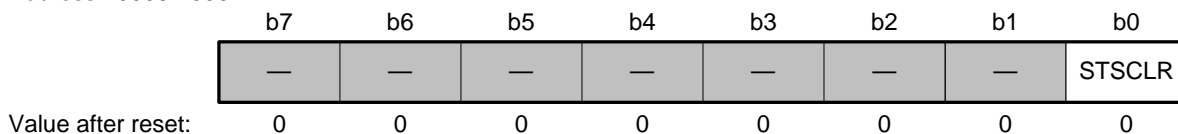
Note: Execution of a self-refresh, an initialization sequence, or mode register setting may only be performed when all the status bits are 0. Do not rewrite the registers (bits) listed in table 12.10 when any of the status bits (SRFST, INIST, MRSST) is set to 1.

Table 12.10 List of Registers and Bits Requiring Checking Status Bits

Function	Register	Bits
Self-refresh	SDSELF	SFEN
Initialization sequence	SDICR	INIRQ
Mode register setting	SDMOD	MR[14:0]

12.3.18 Bus Error Status Clear Register (BERCLR)

Address: 0008 1300h



Bit	Symbol	Bit Name	Description	R/W
b0	STSCLR	Status Clear	0: Invalid 1: Bus error status register cleared	R/W*
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

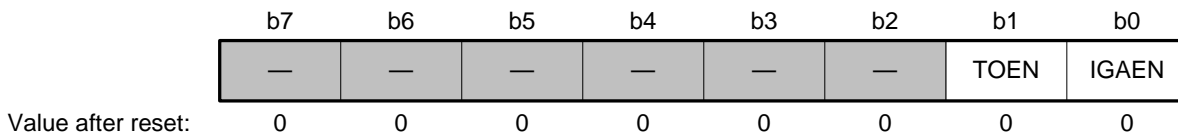
Note: * Only writing 1 is effective; i.e. writing 0 has no effect.

STSCLR Bit (Status Clear)

Writing 1 to this bit clears the bus-error status registers 1 and 2 (BERSR1 and BERSR2).

12.3.19 Bus Error Monitoring Enable Register (BEREN)

Address: 0008 1304h



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 always read as 0. The write value should be 0.	R/W

Note 1. When detection is disabled (the TOEN bit is cleared to 0), bus access can cause the bus to freeze.

Note 2. Do not clear the TOEN bit to 0 (bus timeout detection disabled) while timeout errors are being detected.

IGAEN Bit (Illegal Address Access Detection Enable)

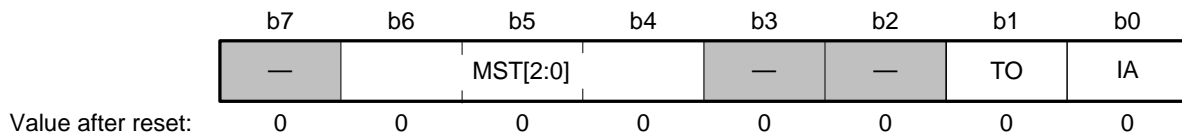
This bit enables or disables detection of access to illegal addresses.

TOEN Bit (Timeout Detection Enable)

This bit enables or disables bus timeout detection.

12.3.20 Bus Error Status Register 1 (BERSR1)

Address: 0008 1308h



Bit	Symbol	Bit Name	Description	R/W
b0	IA	Illegal Address Access	0: Illegal address access not made 1: Illegal address access made	R
b1	TO	Timeout	0: Timeout not generated 1: Timeout generated	R
b3, b2	—	Reserved	These bits are always read as 0. Writing to these bits has no effect.	R
b6 to b4	MST[2:0]	Bus Master Code	b6 b4 0 0 0: CPU 0 0 1: Setting prohibited 0 1 0: Setting prohibited 0 1 1: DTC/DMACA 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: EDMAC 1 1 1: EXDMAC	R
b7	—	Reserved	This bit is always read as 0. Writing to this bit has no effect.	R

BERSR1 indicates the bus error generation status. It indicates whether a timeout occurred (TO bit), whether an illegal address access was made (IA bit), or which bus master accessed the bus (MST[2:0] bits)

IA Bit (Illegal Address Access Flag)

This bit indicates whether an illegal address access was made.

TO Bit (Timeout Flag)

This bit indicates whether a timeout occurred.

MST[2:0] Bits (Bus Master Code)

These bits indicate the bus master that accessed a bus when a bus error occurred.

12.3.21 Bus Error Status Register 2 (BERSR2)

Address: 0008 130Ah



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are always 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

BERSR2 indicates the upper 13 bits of an address having caused a bus error.

ADDR[12:0] Bits (Bus Error Occurrence Address)

These bits indicate the upper 13 bits of an address having caused a bus error.

12.4 Endian and Data Alignment

The external bus has a data-alignment function to control which byte of the data bus (D31 to D24, D23 to D16, D15 to D8, or D7 to D0) is used according to the bus specifications of the area to be accessed (8-bit, 16-bit, or 32-bit bus space), data size, and endian format when accessing the external address space (the CS and SDRAM areas).

12.4.1 Data Alignment Control for CS Area

(1) 32-Bit Bus Space

When a 32-bit width is selected for a bus space by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A2 are enabled to output address signals in units of 32 bits, and the address buses A1 and A0 are disabled.

When byte strobe mode is selected (the WRMOD bit = 0 in CSnMOD), the WR0# to WR3# pins are enabled. The BC0# to BC3# pins are not used.

When single write strobe mode is selected (the WRMOD bit = 1 in CSnMOD), only the WR# pin is enabled and always outputs the low level during write access, regardless of the data size. Here, the valid byte position is indicated by the BC0# to BC3# pins.

In 32-bit bus space, the valid positions of data external to the chip and of control signals differ according to whether the endian is big or little.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	RD#				
						WR3#/BC3#	WR2#/BC2#	WR1#/BC1#	WR0#/BC0#	
						Data Bus				
						D31	D24 D23	D16 D15	D8 D7	D0
8 bits	4n	One	First	8 bits	4n	[7 0]				
	4n+1	One	First	8 bits	4n	[7 0]				
	4n+2	One	First	8 bits	4n	[7 0]				
	4n+3	One	First	8 bits	4n	[7 0]				
16 bits	4n	One	First	16 bits	4n	[15 8 7 0]				
	4n+1	Two	First	8 bits	4n	[7 0]				
			Second	8 bits	4n	[15 8]				
	4n+2	One	First	16 bits	4n	[15 8 7 0]				
4n+3	Two	First	8 bits	4n	[7 0]					
		Second	8 bits	4n+4	[15 8]					
32 bits	4n	One	First	32 bits	4n	[31 24 23 16 15 8 7 0]				
	4n+1	Three	First	8 bits	4n	[7 0]				
			Second	16 bits	4n	[23 16 15 8]				
			Third	8 bits	4n+4	[31 24]				
	4n+2	Two	First	16 bits	4n	[15 8 7 0]				
			Second	16 bits	4n+4	[31 24 23 16]				
	4n+3	Three	First	8 bits	4n	[7 0]				
Second			16 bits	4n+4	[23 16 15 8]					
Third			8 bits	4n+4	[31 24]					

Figure 12.3 Data Alignment (Little Endian) in 32-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	WR3#/BC3#	WR2#/BC2#	WR1#/BC1#	WR0#/BC0#				
						RD#							
						Data Bus							
						D31	D24	D23	D16	D15	D8	D7	D0
8 bits	4n	One	First	8 bits	4n	[7 0]							
	4n+1	One	First	8 bits	4n	[7 0]							
	4n+2	One	First	8 bits	4n	[7 0]							
	4n+3	One	First	8 bits	4n	[7 0]							
16 bits	4n	One	First	16 bits	4n	[15 8 7 0]							
	4n+1	Two	First	8 bits	4n	[15 8]							
			Second	8 bits	4n	[7 0]							
	4n+2	One	First	16 bits	4n	[15 8 7 0]							
4n+3	Two	First	8 bits	4n	[15 8]								
		Second	8 bits	4n+4	[7 0]								
32 bits	4n	One	First	32 bits	4n	[31 24 23 16 15 8 7 0]							
	4n+1	Three	First	8 bits	4n	[31 24]							
			Second	16 bits	4n	[23 16 15 8]							
			Third	8 bits	4n+4	[7 0]							
	4n+2	Two	First	16 bits	4n	[31 24 23 16]							
			Second	16 bits	4n+4	[15 8 7 0]							
4n+3	Three	First	8 bits	4n	[31 24]								
		Second	16 bits	4n+4	[23 16 15 8]								
		Third	8 bits	4n+4	[7 0]								

Figure 12.4 Data Alignment (Big Endian) in 32-Bit Bus Space

(2) 16-Bit Bus Space

When a 16-bit width is selected for a bus space by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A1 are enabled to output address signals in units of 16 bits, and the address bus A0 is disabled.

When byte strobe mode is selected (the WRMOD bit = 0 in CSnMOD), the WR0# and WR1# pins are enabled, and the WR2# and WR3# pins are disabled. The BC0# to BC3# pins are not used.

When single write strobe mode is selected (the WRMOD bit = 1 in CSnMOD), only the WR# pin is enabled and always outputs the low level during write access, regardless of the data size. Here, the valid byte position is indicated by the BC0# and BC1# pins. The WR2# and WR3# pins are not used.

In 16-bit bus space, page access can occur in access to data in 32-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary and causes no change in BC0# and BC1# signals. The situations in which page access occurs are indicated by the letter "(p)" in figure 12.5 and figure 12.6.

In 16-bit bus space, the valid positions of data external to the chip and of control signals differ according to whether the endian is big or little.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	Data Bus			
						D15	D8	D7	D0
8 bits	4n	One	First	8 bits	4n	7 0			
	4n+1	One	First	8 bits	4n	7 0			
	4n+2	One	First	8 bits	4n+2	7 0			
	4n+3	One	First	8 bits	4n+2	7 0			
16 bits	4n	One	First	16 bits	4n	15 8 7 0			
			First	8 bits	4n	7 0			
	4n+1	Two	Second	8 bits	4n+2	15 8			
			First	16 bits	4n+2	15 8 7 0			
4n+3	Two	First	8 bits	4n+2	7 0				
		Second	8 bits	4n+4	15 8				
32 bits	4n	Two	First	16 bits	4n	15 8 7 0			
			Second	16 bits	4n+2 (p)	31 24 23 16			
	4n+1	Three	First	8 bits	4n	7 0			
			Second	16 bits	4n+2	23 16 15 8			
			Third	8 bits	4n+4	31 24			
	4n+2	Two	First	16 bits	4n+2	15 8 7 0			
			Second	16 bits	4n+4	31 24 23 16			
	4n+3	Three	First	8 bits	4n+2	7 0			
Second			16 bits	4n+4	23 16 15 8				
Third			8 bits	4n+6	31 24				

[Legend]

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 12.5 Data Alignment (Little Endian) in 16-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	WR1#/BC1#	WR0#/BC0#	
						RD#		
						Data Bus		
						D15	D8 D7	D0
8 bits	4n	One	First	8 bits	4n	7	0	
	4n+1	One	First	8 bits	4n		7 0	
	4n+2	One	First	8 bits	4n+2	7	0	
	4n+3	One	First	8 bits	4n+2		7 0	
16 bits	4n	One	First	16 bits	4n	15	8 7 0	
	4n+1	Two	First	8 bits	4n		15 8	
			Second	8 bits	4n+2	7	0	
	4n+2	One	First	16 bits	4n+2	15	8 7 0	
	4n+3	Two	First	8 bits	4n+2		15 8	
			Second	8 bits	4n+4	7	0	
32 bits	4n	Two	First	16 bits	4n	31	24 23 16	
			Second	16 bits	4n+2 (p)	15	8 7 0	
	4n+1	Three	First	8 bits	4n		31 24	
			Second	16 bits	4n+2	23	16 15 8	
			Third	8 bits	4n+4	7	0	
	4n+2	Two	First	16 bits	4n+2	31	24 23 16	
			Second	16 bits	4n+4	15	8 7 0	
	4n+3	Three	First	8 bits	4n+2		31 24	
			Second	16 bits	4n+4	23	16 15 8	
Third			8 bits	4n+6	7	0		

[Legend]

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 12.6 Data Alignment (Big Endian) in 16-Bit Bus Space

(3) 8-Bit Bus Space

When an 8-bit bus space is selected by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A0 are enabled to output address signals in byte units.

Since setting single write strobe mode is prohibited in 8-bit bus space, always select the byte strobe mode (the CSnMOD.WRMOD bit is 0). In write access mode, the WR0# pin is only valid. The WR1# to WR3# pins and the BC0# to BC3# pins are not used.

Page access can occur in access to data in 16- or 32-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary. The situations in which page access occurs are indicated by the letter "(p)" in figure 12.7 and figure 12.8.

In 8-bit bus space, the valid positions of data external to the chip are D7 to D0 and WR0# is used as the control signal, regardless of the endian mode.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	WR1#/BC1# WR0#/BC0#	
						RD#	
						Data Bus	
						D15	D8 D7 D0
8 bits	4n	One	First	8 bits	4n	[7	0]
	4n+1	One	First	8 bits	4n+1	[7	0]
	4n+2	One	First	8 bits	4n+2	[7	0]
	4n+3	One	First	8 bits	4n+3	[7	0]
16 bits	4n	Two	First	8 bits	4n	[7	0]
			Second	8 bits	4n+1 (p)	[15	8]
	4n+1	Two	First	8 bits	4n+1	[7	0]
			Second	8 bits	4n+2 (p)	[15	8]
	4n+2	Two	First	8 bits	4n+2	[7	0]
			Second	8 bits	4n+3 (p)	[15	8]
	4n+3	Two	First	8 bits	4n+3	[7	0]
			Second	8 bits	4n+4	[15	8]
32 bits	4n	Four	First	8 bits	4n	[7	0]
			Second	8 bits	4n+1 (p)	[15	8]
			Third	8 bits	4n+2 (p)	[23	16]
			Fourth	8 bits	4n+3 (p)	[31	24]
	4n+1	Four	First	8 bits	4n+1	[7	0]
			Second	8 bits	4n+2 (p)	[15	8]
			Third	8 bits	4n+3 (p)	[23	16]
			Fourth	8 bits	4n+4	[31	24]
	4n+2	Four	First	8 bits	4n+2	[7	0]
			Second	8 bits	4n+3 (p)	[15	8]
			Third	8 bits	4n+4	[23	16]
			Fourth	8 bits	4n+5 (p)	[31	24]
	4n+3	Four	First	8 bits	4n+3	[7	0]
			Second	8 bits	4n+4	[15	8]
			Third	8 bits	4n+5 (p)	[23	16]
			Fourth	8 bits	4n+6 (p)	[31	24]

[Legend]

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 12.7 Data Alignment (Little Endian) in 8-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	WR1#/BC1#		WR0#/BC0#	
						RD#			
						Data Bus			
						D15	D8	D7	D0
8 bits	4n	One	First	8 bits	4n	7	0		
	4n+1	One	First	8 bits	4n+1	7	0		
	4n+2	One	First	8 bits	4n+2	7	0		
	4n+3	One	First	8 bits	4n+3	7	0		
16 bits	4n	Two	First	8 bits	4n	15	8		
			Second	8 bits	4n+1 (p)	7	0		
	4n+1	Two	First	8 bits	4n+1	15	8		
			Second	8 bits	4n+2 (p)	7	0		
	4n+2	Two	First	8 bits	4n+2	15	8		
			Second	8 bits	4n+3 (p)	7	0		
	4n+3	Two	First	8 bits	4n+3	15	8		
			Second	8 bits	4n+4	7	0		
32 bits	4n	Four	First	8 bits	4n	31	24		
			Second	8 bits	4n+1 (p)	23	16		
			Third	8 bits	4n+2 (p)	15	8		
			Fourth	8 bits	4n+3 (p)	7	0		
	4n+1	Four	First	8 bits	4n+1	31	24		
			Second	8 bits	4n+2 (p)	23	16		
			Third	8 bits	4n+3 (p)	15	8		
			Fourth	8 bits	4n+4	7	0		
	4n+2	Four	First	8 bits	4n+2	31	24		
			Second	8 bits	4n+3 (p)	23	16		
			Third	8 bits	4n+4	15	8		
			Fourth	8 bits	4n+5 (p)	7	0		
	4n+3	Four	First	8 bits	4n+3	31	24		
			Second	8 bits	4n+4	23	16		
			Third	8 bits	4n+5 (p)	15	8		
			Fourth	8 bits	4n+6 (p)	7	0		

[Legend]

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 12.8 Data Alignment (Big Endian) in 8-Bit Bus Space

12.4.2 Data Alignment Control for SDRAM Area

(1) 32-Bit Bus Space

When a 32-bit width is selected for a bus space by the BSIZE[1:0] bits in SDCCR, the address buses A27 to A2 are enabled to output address signals in units of 32 bits, and the address buses A1 and A0 are disabled.

The external data is accessed using the D31 to D24, D23 to D16, D15 to D8, and D7 to D0 pins. Either 8-, 16-, or 32-bit data can be accessed at a time. The valid byte position is indicated by DQM0 to DQM3 signals.

In 32-bit bus space, the valid positions of data external to the chip and of SDRAM control signals (DQM0 to DQM3) differ according to whether the endian is big or little. figure 12.9 and figure 12.10 show data alignment control when the endian is little and big, respectively.

In 32-bit bus space, consecutive access can occur in access to data in 8-, 16-, or 32-bit units. Specifically, consecutive access can occur when a single round of bus access is generated in response to a single transfer request. The situations in which consecutive access occurs are indicated by the letter "(r1)" in figure 12.9 and figure 12.10. Figure 12.15 shows a consecutive access example.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	Data Bus				
						D31	D24 D23	D16 D15	D8 D7	D0
8 bits	4n	One	First	8 bits	4n (r1)	[7 0]				
	4n+1	One	First	8 bits	4n (r1)	[7 0]				
	4n+2	One	First	8 bits	4n (r1)	[7 0]				
	4n+3	One	First	8 bits	4n (r1)	[7 0]				
16 bits	4n	One	First	16 bits	4n (r1)	[15 8 7 0]				
	4n+1	Two	First	8 bits	4n	[7 0]				
			Second	8 bits	4n	[15 8]				
	4n+2	One	First	16 bits	4n (r1)	[15 8 7 0]				
32 bits	4n	One	First	32 bits	4n (r1)	[31 24 23 16 15 8 7 0]				
	4n+1	Three	First	8 bits	4n	[7 0]				
			Second	16 bits	4n	[23 16 15 8]				
			Third	8 bits	4n+4	[31 24]				
4n+2	Two	First	16 bits	4n	[15 8 7 0]					
		Second	16 bits	4n+4	[31 24 23 16]					
		First	8 bits	4n	[7 0]					
4n+3	Two	Second	16 bits	4n+4	[23 16 15 8]					
		Third	8 bits	4n+4	[31 24]					

[Legend]

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the EXDMAC block or cluster transfer in single address mode)

Figure 12.9 Data Alignment (Little Endian) in 32-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	Data bus			
						WE#			
						D31	D24 D23	D16 D15	D8 D7
8 bits	4n	One	First	8 bits	4n (r1)	[7 0]			
	4n+1	One	First	8 bits	4n (r1)	[7 0]			
	4n+2	One	First	8 bits	4n (r1)	[7 0]			
	4n+3	One	First	8 bits	4n (r1)	[7 0]			
16 bits	4n	One	First	16 bits	4n (r1)	[15 8 7 0]			
	4n+1	Two	First	8 bits	4n	[15 8]			
			Second	8 bits	4n	[7 0]			
	4n+2	One	First	16 bits	4n (r1)	[15 8 7 0]			
	4n+3	Two	First	8 bits	4n	[15 8]			
			Second	8 bits	4n+4	[7 0]			
32 bits	4n	One	First	32 bits	4n (r1)	[31 24 23 16 15 8 7 0]			
	4n+1	Three	First	8 bits	4n	[31 24]			
			Second	16 bits	4n	[23 16 15 8]			
			Third	8 bits	4n+4	[7 0]			
	4n+2	Two	First	16 bits	4n	[31 24 23 16]			
			Second	16 bits	4n+4	[15 8 7 0]			
	4n+3	Two	First	8 bits	4n	[31 24]			
			Second	16 bits	4n+4	[23 16 15 8]			
			Third	8 bits	4n+4	[7 0]			

[Legend]

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the EXDMAC block or cluster transfer in single address mode)

Figure 12.10 Data Alignment (Big Endian) in 32-Bit Bus Space

(2) 16-Bit Bus Space

When a 16-bit width is selected for a bus space by the BSIZE[1:0] bits in SDCCR, the address buses A27 to A0 are enabled to output address signals in units of 16 bits, and the address buses A0 is disabled (always output the low level). The valid byte position is indicated by DQM0 and DQM1 signals. DQM2 and DQM3 signals are not used.

In 16-bit bus space, the external data is accessed using the D15 to D8 and D7 to D0 pins and DQM0 and DQM1 control signals. Either 8- or 16-bit data can be accessed at a time.

In 16-bit bus space, the valid positions of data external to the chip and of control signals differ according to whether the endian is big or little. Figure 12.11 and figure 12.12 show data alignment control when the endian is little and big, respectively.

In 16-bit bus space, consecutive access can occur in access to data in 8- or 16-bit units. Specifically, consecutive access can occur when a single round of bus access is generated in response to a single transfer request. The situations in which consecutive access occurs are indicated by the letter "(r1)" in figure 12.11 and figure 12.12. Figure 12.15 shows a consecutive access example.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	DQM1		DQM0	
						WE#		Data Bus	
						D15	D7	D0	
8 bits	4n	One	First	8 bits	4n (r1)	7		0	
	4n+1	One	First	8 bits	4n (r1)	7		0	
	4n+2	One	First	8 bits	4n+2 (r1)	7		0	
	4n+3	One	First	8 bits	4n+2 (r1)	7		0	
16 bits	4n	One	First	16 bits	4n (r1)	15	8	7	0
	4n+1	Two	First	8 bits	4n	7		0	
			Second	8 bits	4n+2	15		8	
	4n+2	One	First	16 bits	4n+2 (r1)	15	8	7	0
4n+3	Two	First	8 bits	4n+2	7		0		
		Second	8 bits	4n+4	15		8		
32 bits	4n	Two	First	16 bits	4n	15	8	7	0
			Second	16 bits	4n+2	31	24	23	16
	4n+1	Three	First	8 bits	4n	7		0	
			Second	16 bits	4n+2	23	16	15	8
			Third	8 bits	4n+4	31		24	
	4n+2	Two	First	16 bits	4n+2	15	8	7	0
			Second	16 bits	4n+4	31	24	23	16
	4n+3	Three	First	8 bits	4n+2	7		0	
Second			16 bits	4n+4	23	16	15	8	
Third			8 bits	4n+6	31		24		

[Legend]

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the EXDMAC block or cluster transfer in single address mode)

Figure 12.11 Data Alignment (Little Endian) in 16-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	DQM1		DQM0	
						WE#			
						D15	D7	D0	
8 bits	4n	One	First	8 bits	4n (r1)	7	0		
	4n+1	One	First	8 bits	4n (r1)		7	0	
	4n+2	One	First	8 bits	4n+2 (r1)	7	0		
	4n+3	One	First	8 bits	4n+2 (r1)		7	0	
16 bits	4n	One	First	16 bits	4n (r1)	15	8	7	0
	4n+1	Two	First	8 bits	4n		15	8	
			Second	8 bits	4n+2	7	0		
	4n+2	One	First	16 bits	4n+2 (r1)	15	8	7	0
	4n+3	Two	First	8 bits	4n+2		15	8	
			Second	8 bits	4n+4	7	0		
32 bits	4n	Two	First	16 bits	4n	31	24	23	16
			Second	16 bits	4n+2	15	8	7	0
	4n+1	Three	First	8 bits	4n		31	24	
			Second	16 bits	4n+2	23	16	15	8
			Third	8 bits	4n+4	7	0		
	4n+2	Two	First	16 bits	4n+2	31	24	23	16
			Second	16 bits	4n+4	15	8	7	0
	4n+3	Three	First	8 bits	4n+2		31	24	
			Second	16 bits	4n+4	23	16	15	8
Third			8 bits	4n+6	7	0			

[Legend]

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the EXDMAC block or cluster transfer in single address mode)

Figure 12.12 Data Alignment (Big Endian) in 16-Bit Bus Space

(3) 8-Bit Bus Space

When an 8-bit width is selected for a bus space by the BSIZE[1:0] bits in SDCCR, the address buses A27 to A0 are enabled to output address signals in units of 8 bits.

In 8-bit bus space, the external data is accessed using the D7 to D0 pins and DQM0 control signal. Eight-bit data can be accessed at a time; 16-bit data is accessed with two 8-bit accesses and 32-bit data is accessed with four 8-bit accesses.

Figure 12.13 and figure 12.14 show data alignment control when the endian is little and big, respectively.

In 8-bit bus space, consecutive access can occur in access to data in 8-bit units. Specifically, consecutive access can occur when a single round of bus access is generated in response to a single transfer request. The situations in which consecutive access occurs are indicated by the letter "(r1)" in figure 12.13 and figure 12.14. Figure 12.15 shows a consecutive access example.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	DQM1		DQM0	
						WE#			
						D15	D7	D0	
8 bits	4n	One	First	8 bits	4n (r1)	[7 0]			
	4n+1	One	First	8 bits	4n+1 (r1)	[7 0]			
	4n+2	One	First	8 bits	4n+2 (r1)	[7 0]			
	4n+3	One	First	8 bits	4n+3 (r1)	[7 0]			
16 bits	4n	Two	First	8 bits	4n	[7 0]			
			Second	8 bits	4n+1	[15 8]			
	4n+1	Two	First	8 bits	4n+1	[7 0]			
			Second	8 bits	4n+2	[15 8]			
	4n+2	Two	First	8 bits	4n+2	[7 0]			
			Second	8 bits	4n+3	[15 8]			
	4n+3	Two	First	8 bits	4n+3	[7 0]			
			Second	8 bits	4n+4	[15 8]			
32 bits	4n	Four	First	8 bits	4n	[7 0]			
			Second	8 bits	4n+1	[15 8]			
			Third	8 bits	4n+2	[23 16]			
			Fourth	8 bits	4n+3	[31 24]			
	4n+1	Four	First	8 bits	4n+1	[7 0]			
			Second	8 bits	4n+2	[15 8]			
			Third	8 bits	4n+3	[23 16]			
			Fourth	8 bits	4n+4	[31 24]			
	4n+2	Four	First	8 bits	4n+2	[7 0]			
			Second	8 bits	4n+3	[15 8]			
			Third	8 bits	4n+4	[23 16]			
			Fourth	8 bits	4n+5	[31 24]			
	4n+3	Four	First	8 bits	4n+3	[7 0]			
			Second	8 bits	4n+4	[15 8]			
			Third	8 bits	4n+5	[23 16]			
			Fourth	8 bits	4n+6	[31 24]			

[Legend]

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the EXDMAC block or cluster transfer in single address mode)

Figure 12.13 Data Alignment (Little Endian) in 8-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	(r1)	DQM1		DQM0	
							WE#		Data Bus	
							D15	D7	D0	
8 bits	4n	One	First	8 bits	4n	(r1)	7	0		
	4n+1	One	First	8 bits	4n+1	(r1)	7	0		
	4n+2	One	First	8 bits	4n+2	(r1)	7	0		
	4n+3	One	First	8 bits	4n+3	(r1)	7	0		
16 bits	4n	Two	First	8 bits	4n		15	8		
			Second	8 bits	4n+1		7	0		
	4n+1	Two	First	8 bits	4n+1		15	8		
			Second	8 bits	4n+2		7	0		
	4n+2	Two	First	8 bits	4n+2		15	8		
			Second	8 bits	4n+3		7	0		
	4n+3	Two	First	8 bits	4n+3		15	8		
			Second	8 bits	4n+4		7	0		
32 bits	4n	Four	First	8 bits	4n		31	24		
			Second	8 bits	4n+1		23	16		
			Third	8 bits	4n+2		15	8		
			Fourth	8 bits	4n+3		7	0		
	4n+1	Four	First	8 bits	4n+1		31	24		
			Second	8 bits	4n+2		23	16		
			Third	8 bits	4n+3		15	8		
			Fourth	8 bits	4n+4		7	0		
	4n+2	Four	First	8 bits	4n+2		31	24		
			Second	8 bits	4n+3		23	16		
			Third	8 bits	4n+4		15	8		
			Fourth	8 bits	4n+5		7	0		
	4n+3	Four	First	8 bits	4n+3		31	24		
			Second	8 bits	4n+4		23	16		
			Third	8 bits	4n+5		15	8		
			Fourth	8 bits	4n+6		7	0		

[Legend]

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the EXDMAC block or cluster transfer in single address mode)

Figure 12.14 Data Alignment (Big Endian) in 8-Bit Bus Space

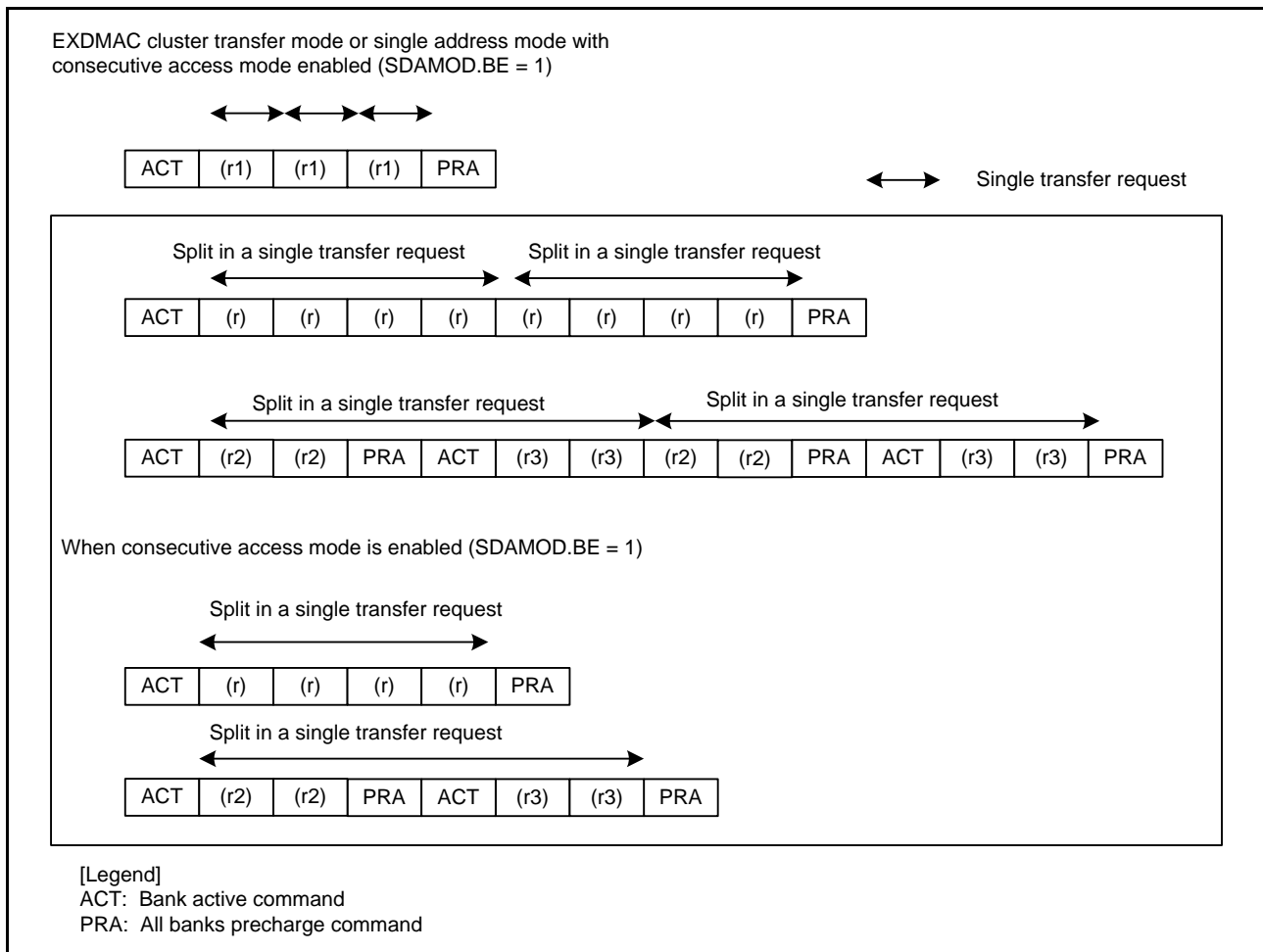


Figure 12.15 Consecutive Access Example

12.5 Operation of CS Area Controller

12.5.1 Timing of CS Area Access

The various periods in the timing charts are described below.

The CS area controller (CSC) operates in synchronization with the external bus clock (BCLK). The operation cycles, such as wait cycles specified with the CSC register, are counted on BCLK. In the following description, frequencies of BCLK and BCLK pin output are the same, unless otherwise noted.

1. Tw1 to Twn (Clock Cycles of Waiting for a Normal Read Cycle or Normal Write Cycle)

The period Tw1 to Twn is made up of the number of clock cycles between the start of access via the external bus clock and one cycle before the strobe signal is valid. The number of cycles is selectable within the range from zero to 31. Within this period, the timing of CSn#, RD#, and WRn# assertion (placing the signals at the low level) is determined by the respective wait settings. Specifically, the periods of waiting are controlled by the CS assert wait select bits (CSON), the RD assert wait select bits (RDON), the WR assert wait select bits (WRON), and the write data output wait select bits (WDON) of CSn wait control register 2 (CSnWCR2). The number of clock cycles for each of these periods of waiting is selectable as a value from zero to seven counted from the start of external bus access. Selectable numbers of cycles are also within the overall number of clock cycles of waiting for reading or writing.

The external bus access is started at the rising edge of the BCLK pin output pulse. However, if two or more rounds of external bus access are generated in response to a single transfer request from the bus master, the second and subsequent rounds of the external bus access may be started at the falling edge of the BCLK pin output pulse according to the wait settings (see figure 12.24 to figure 12.28).

2. Tend (Clock Cycle where the Strobe Signal is Valid)

Tend is the next clock cycle after completion of the period of waiting for a normal cycle of reading or writing or for a cycle of page reading or page writing. If the number of clock cycles in the period of waiting for a normal cycle of reading or writing or for a cycle of page reading or page writing is zero, the clock cycle where bus access starts is the clock cycle where the strobe signal is valid. The RD# and WRn# signals are negated in the next clock cycle after the cycle where the strobe signal is valid. In the case of read access, the clock cycle where the strobe signal is valid becomes the clock cycle where the data to be read are sampled.

If an external wait is enabled, the wait signal is sampled at the time of the cycle where the strobe signal is valid. The bus cycle is extended if the wait signal is at the low level. The bus cycle is completed in the next clock cycle if the wait signal is at the high level. Tend indicates the cycle where sampling of the wait signal starts.

After the first cycle where the strobe signal is valid during page access, second and subsequent page access operations (point 5. below) start in the next cycle except in cases of write access where a setting (other than zero) for write-data output extension clock cycles (point 4. below) has been made. If the setting for the RD or WR assertion wait is a value other than zero, the RD# and WRn# signals are negated in the next clock cycle. If the setting is zero, assertion continues. Furthermore, the CSn# signal continues to be asserted rather than being negated.

3. Tn1 to Tnm (Clock Cycles of CS Extension)

In the case of normal access, Tn1 to Tnm represent the clock cycles of the period following the cycle where the strobe signal is valid (Tend) up to negation of the CSn# signal. For read or write access, the timing of negation can be controlled by the read-access CS extension cycle select bits (CSROFF) and the write-access CS extension cycle select bits (CSWOFF) in the CSn wait control register 2 (CSnWCR2), respectively.

The number of cycles is counted from the cycle following the cycle where the strobe signal is valid.

In the case of page access, Tn1 to Tnm represent the clock cycles of the period for the cycle following the last cycle where the strobe signal is valid up to negation of the CSn# signal.

For write access, setting the write data output extension cycle select bits (WDOFF) controls extension of the period where the address and output data are valid.

4. Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles)

For write access, if the setting for write-data output extension wait is a value other than zero, clock cycles of write-data output extension are inserted from the cycle that follows the cycle where the strobe signal is valid (Tend).

In the case of normal access, this is inserted within the period of clock cycles of CS extension (point 3. above).

In the case of page access, this is inserted within the period of the cycle where the strobe signal is valid and subsequent page access or within the period of clock cycles of CS extension (point 3. above). Valid address and data output are extended over this period, and the WRn# signal is negated.

5. Tpw1 to Tpwn (Page-Read Cycle Wait or Page-Write Cycle Wait)

For the second and subsequent bus cycles during page access, the values for a page-read cycle wait or page-write cycle wait are used instead of the settings for a normal read or write cycle wait. Setting the WR assert wait select bits becomes enabled in the same way as for the first round of access. How the setting for RD assertion controls operation depends on the setting for page-read access mode (the PRMOD bit in CSnMOD) as described below.

CSnMOD.PRMOD = 0: A wait until RD assertion is inserted in the same way as for the first round of access, and the RD# signal is negated.

CSnMOD.PRMOD = 1: Although a wait until RD assertion is inserted in the same way as for normal-access compatibility mode, the RD# signal continues to be asserted over this period.

6. Tr1 to Trn (Recovery Cycles)

Recovery cycles can be inserted from the point where a bus cycle is completed (CSn# signal negation). The number of recovery cycles can be controlled by the setting of the read recovery (RRCV) or write recovery (WRCV) bits in the CSn recovery cycle register (CSnREC). Both numbers of recovery cycles are counted from the end of a bus cycle (CSn# negation) and can be selected from 0 to 15 cycles. For details on recovery cycles, see section 12.5.4, Insertion of Recovery Cycles.

(1) Normal Access

When the PRENB and PWENB bits in CSnMOD are set to 0 to disable page-read and page-write access, respectively, all bus accesses will take the form of normal access operations.

Even when the PRENB and PWENB bits in CSnMOD are set to 1 to enable page-read and page-write access, respectively, bus access other than page access will take the form of normal access operations.

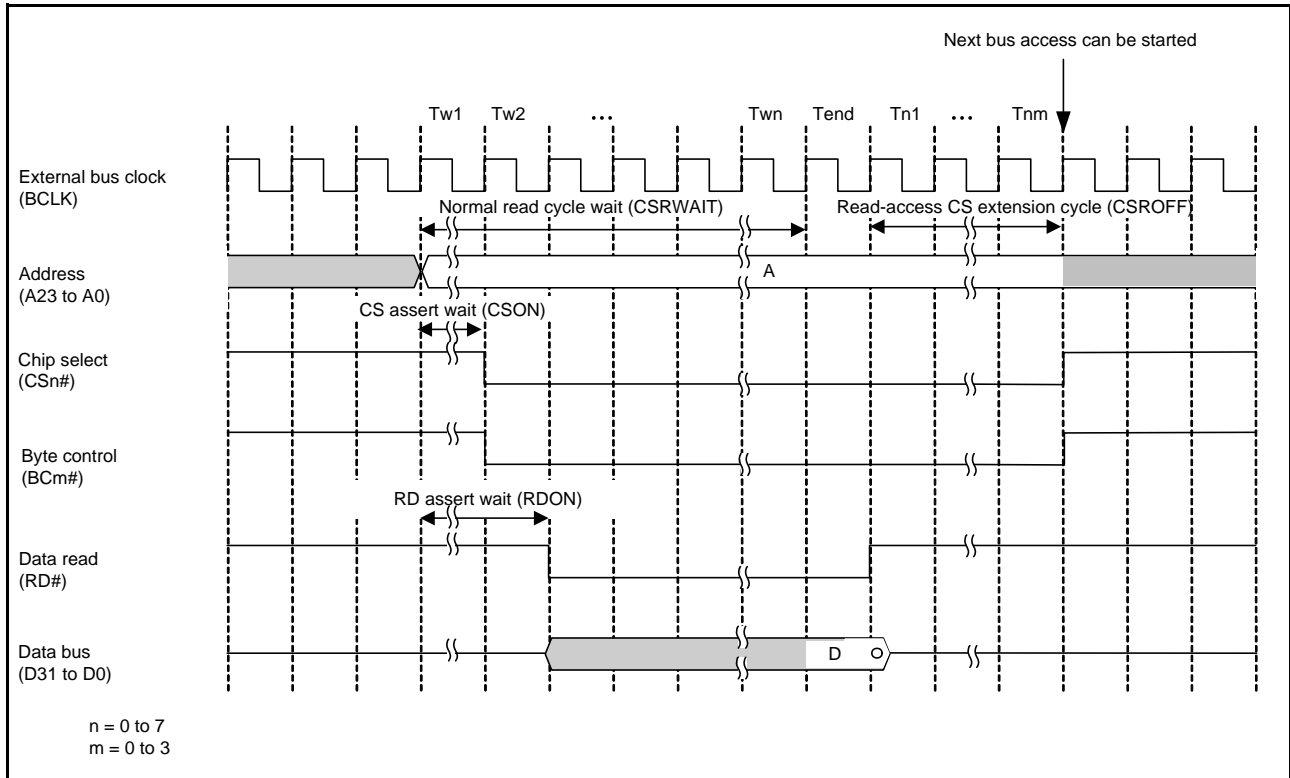


Figure 12.16 Bus Timing (Normal-Read Operation)

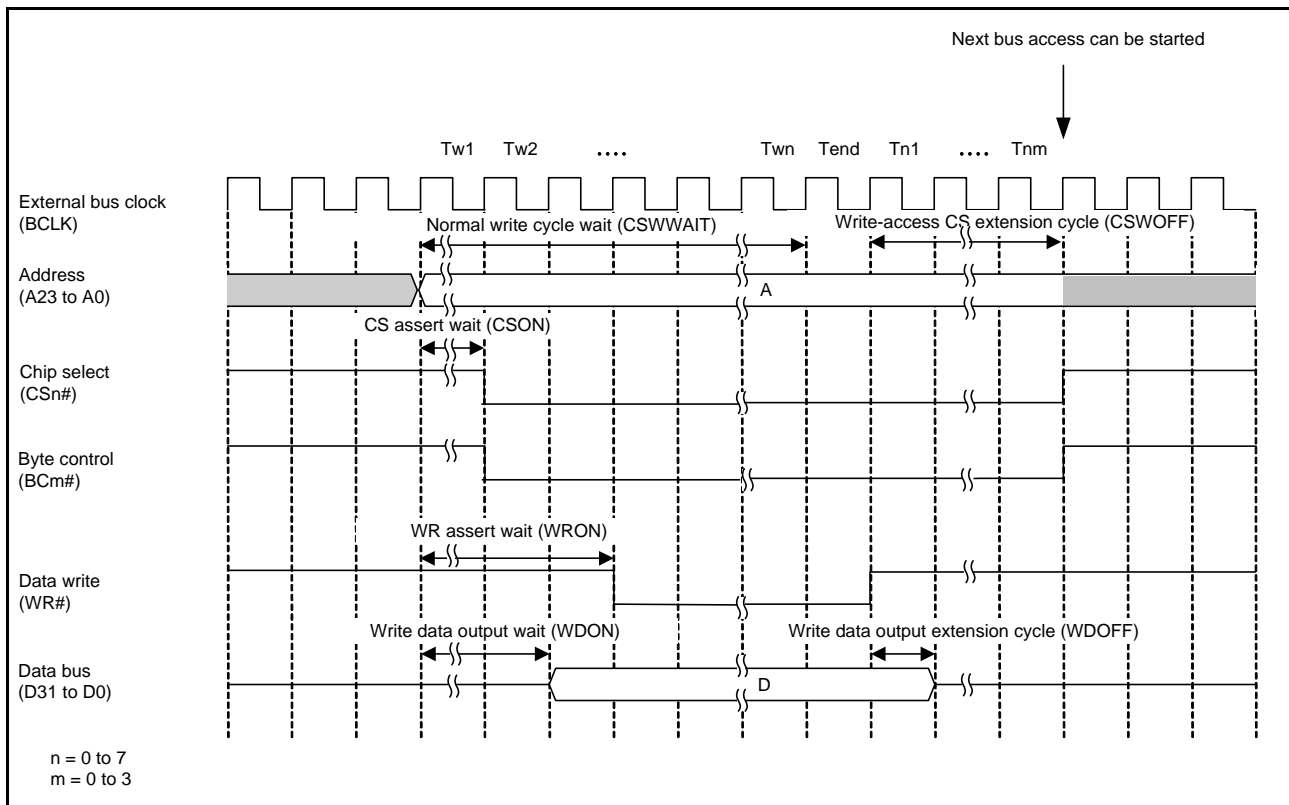


Figure 12.17 Bus Timing (Normal-Write Operation)

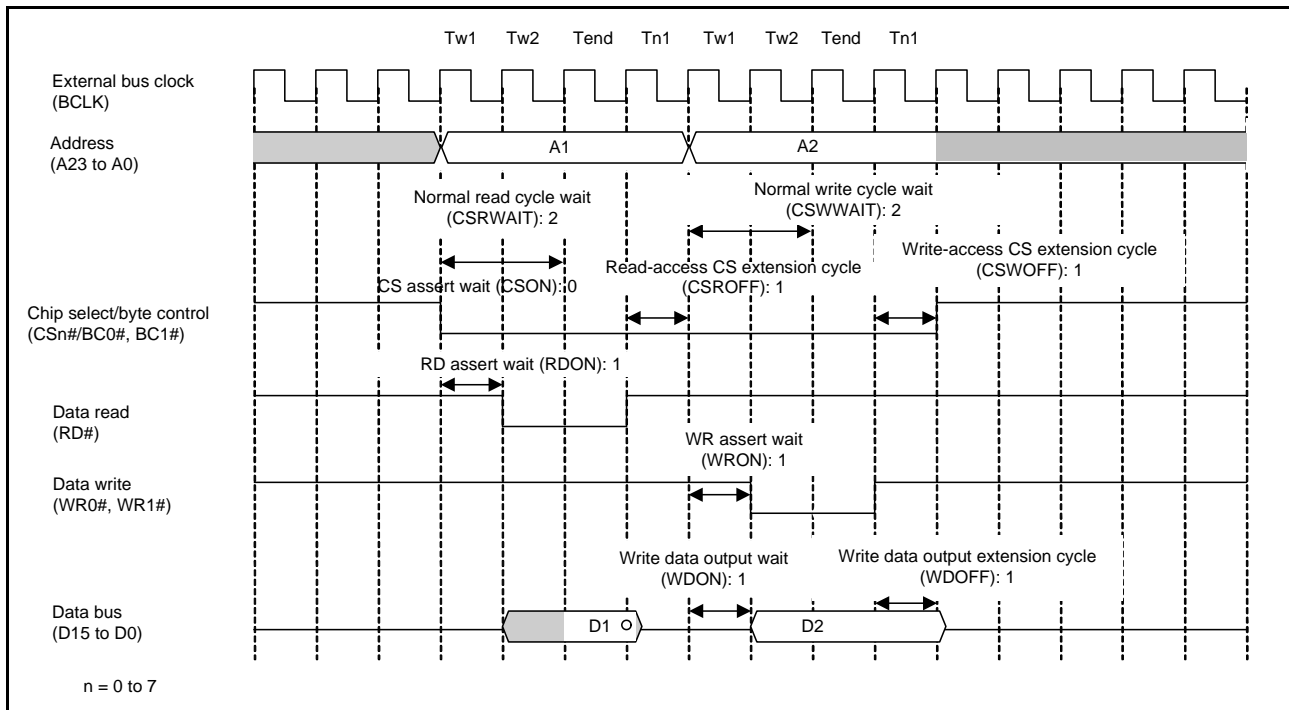


Figure 12.18 Example of Normal Access Operation (Read/Write)

When two or more rounds of external bus access are required in response to a single request for transfer from a bus master, normal access operations (points 1. to 4. above) are repeated. Figure 12.19 and figure 12.20 show examples of operations when two rounds of bus access are generated in response to a single request for transfer. The values of the wait control registers are example settings. In practice, set the register bits according to the specifications of connected devices.

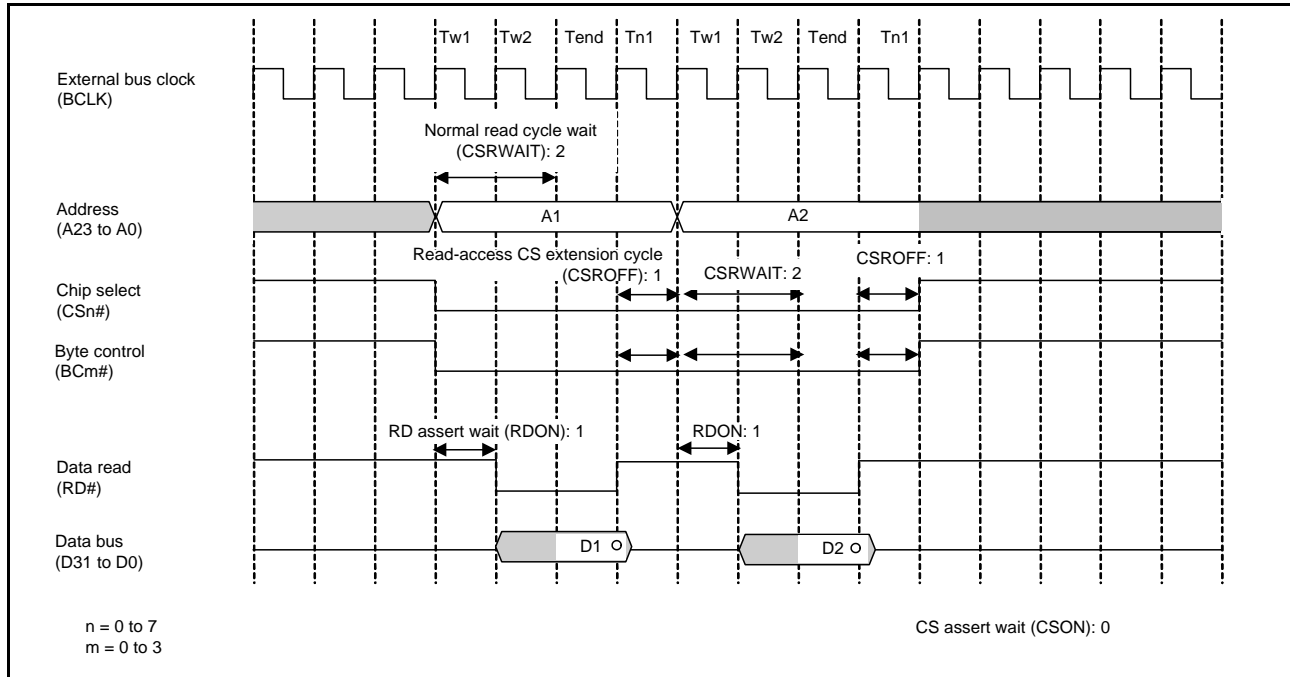


Figure 12.19 Example of Normal-Read Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer)

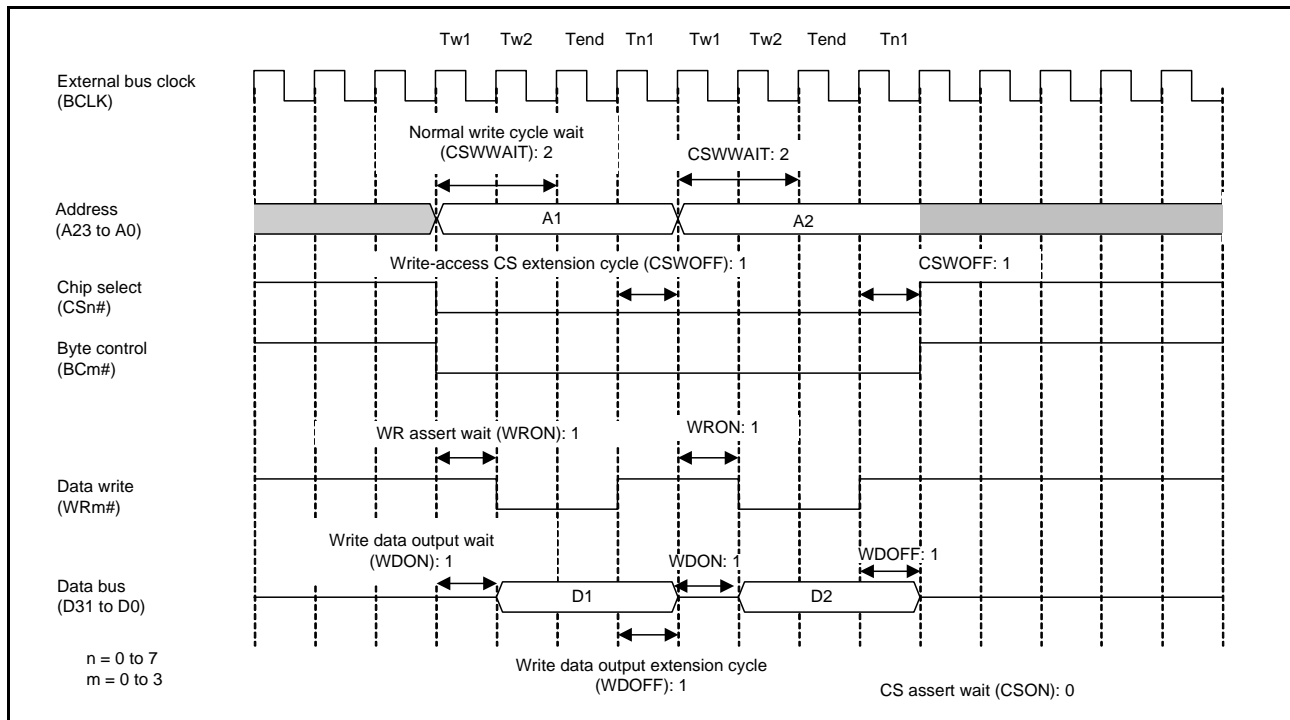


Figure 12.20 Example of Normal-Write Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode)

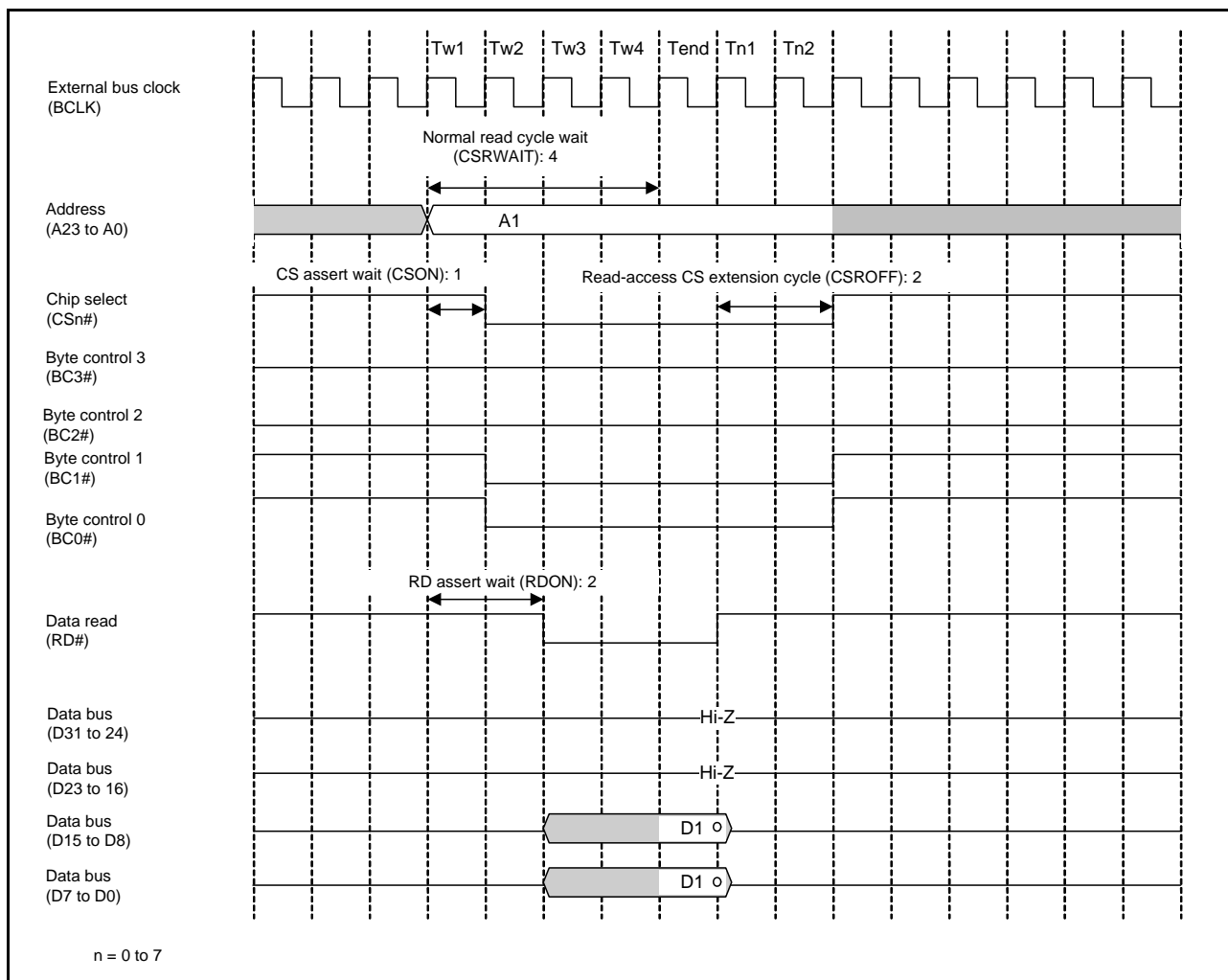


Figure 12.21 Example of Normal-Read Operation (when 32-Bit Bus Space is Accessed in 16 Bits)

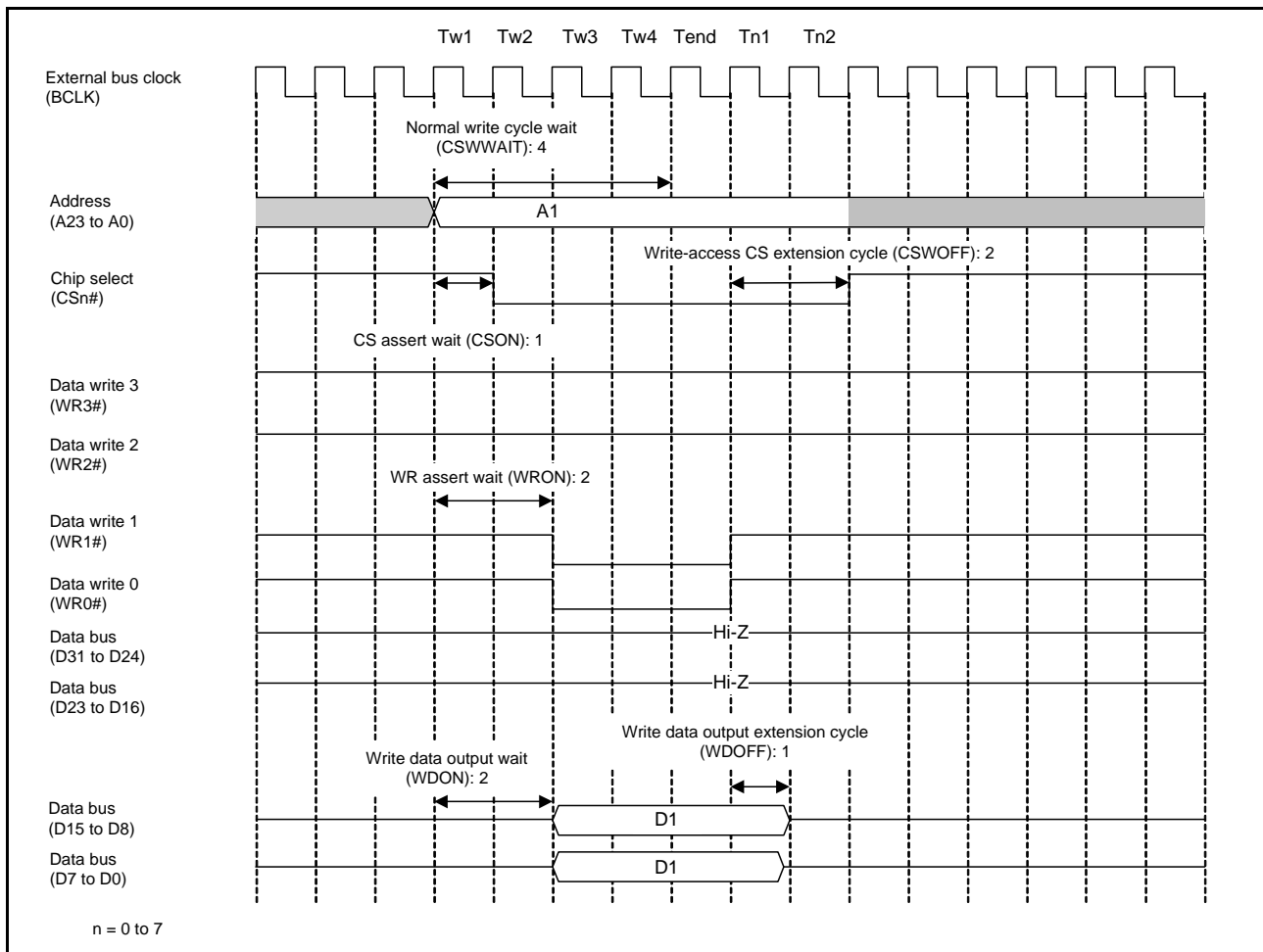


Figure 12.22 Example of Normal-Write Operation (when 32-Bit Bus Space is Accessed in 16 Bits, in Byte Strobe Mode)

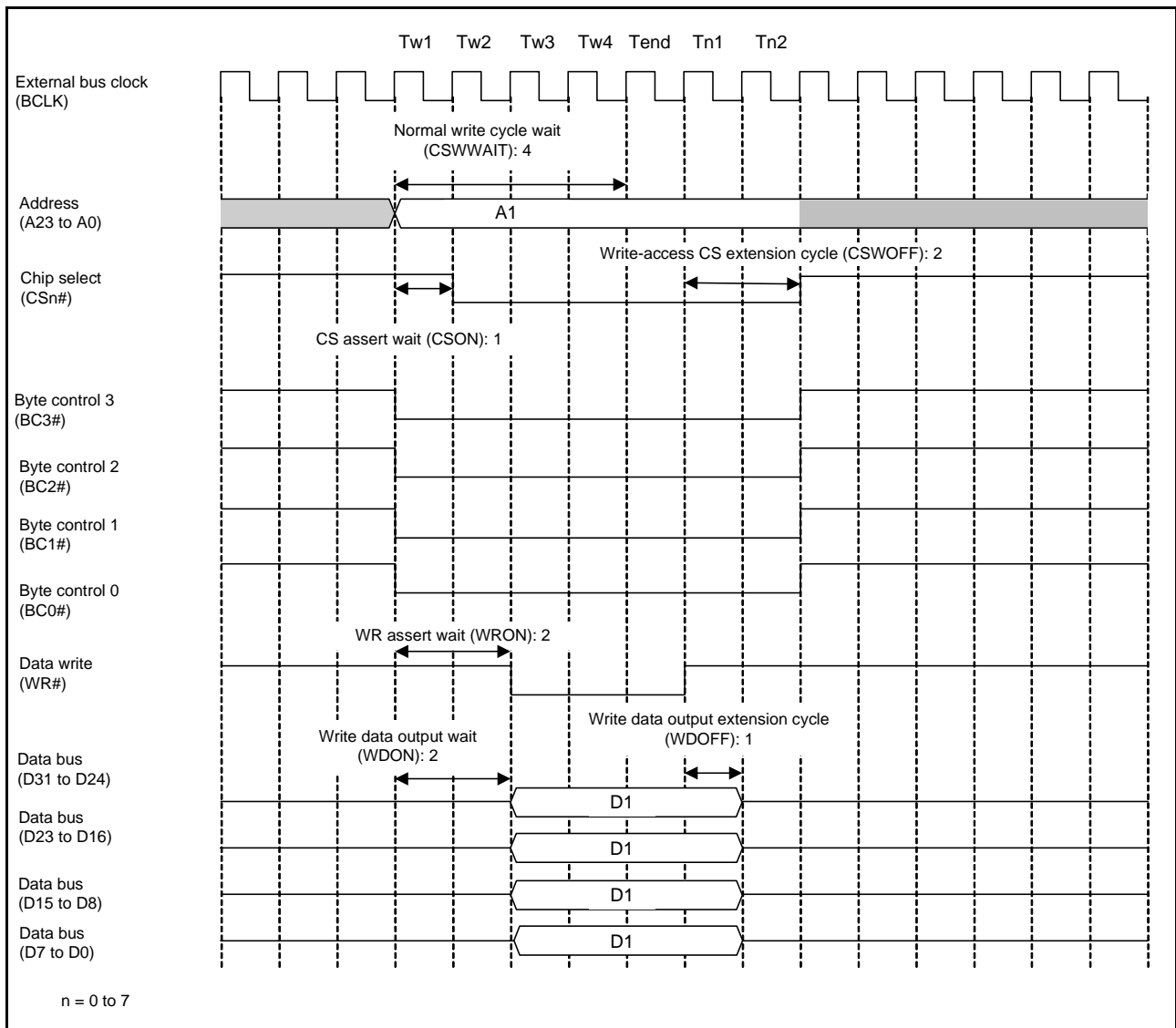


Figure 12.23 Example of Normal-Write Operation (when 32-Bit Bus Space is Accessed in 32 Bits, in Single Write Strobe Mode)

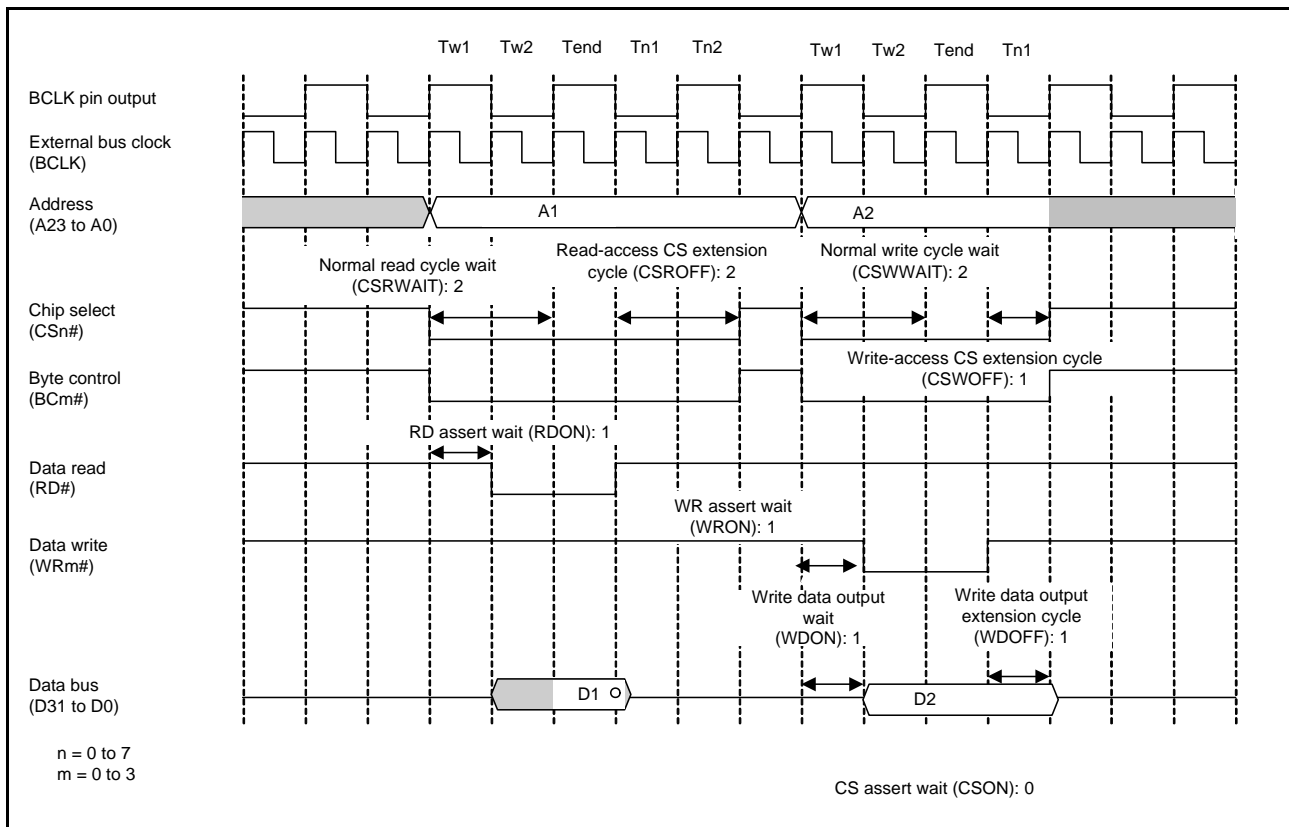


Figure 12.24 Example of Normal Access (when BCLK × 1/2 is Selected with the BCLK Pin Output Select Bit)

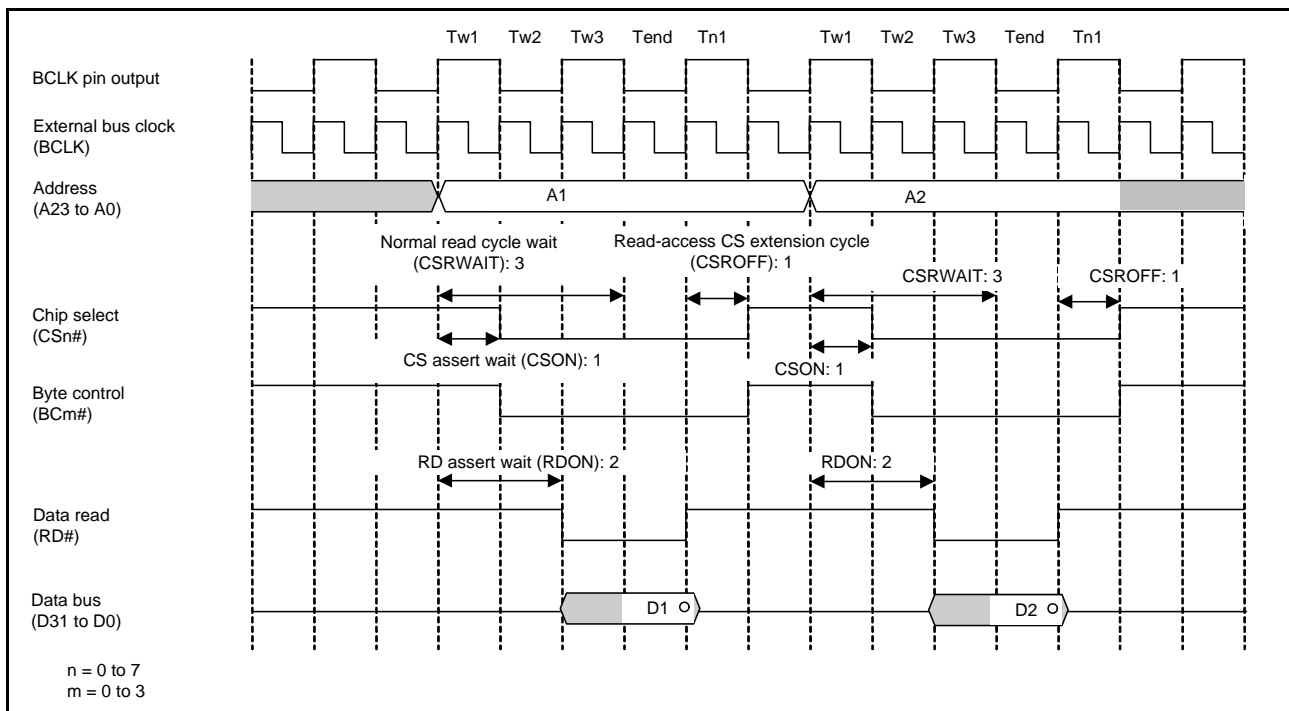


Figure 12.25 Example of Normal-Read Operation (when BCLK × 1/2 is Selected with the BCLK Pin Output Select Bit)

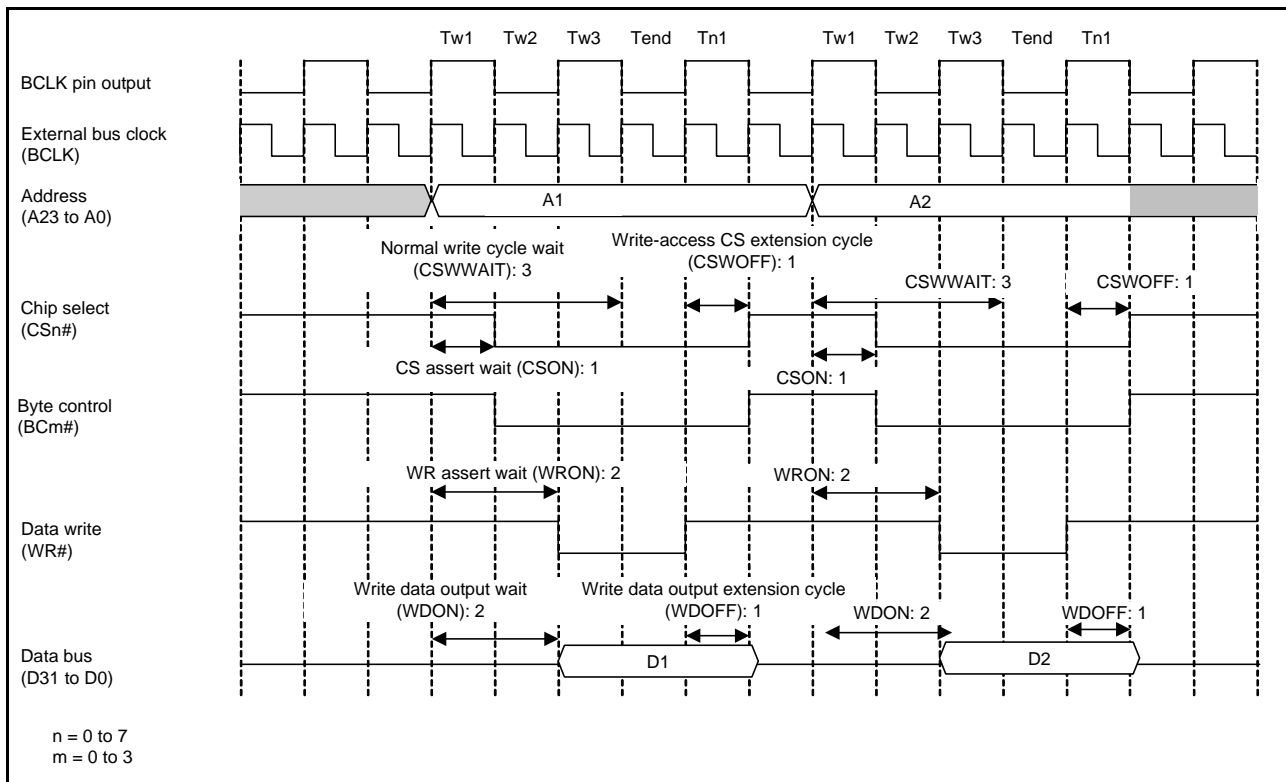


Figure 12.26 Example of Normal-Write Operation (when BCLK × 1/2 is Selected with the BCLK Pin Output Select Bit)

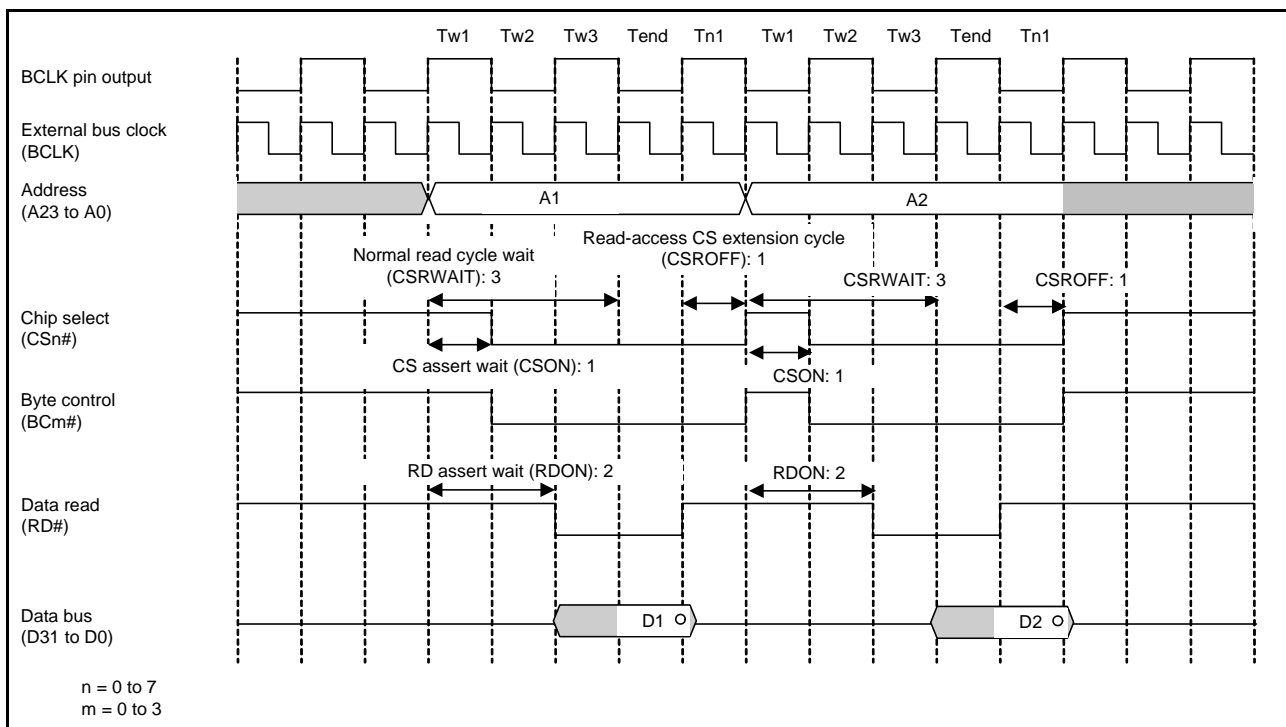


Figure 12.27 Example of Normal-Read Operation (when BCLK × 1/2 is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer)

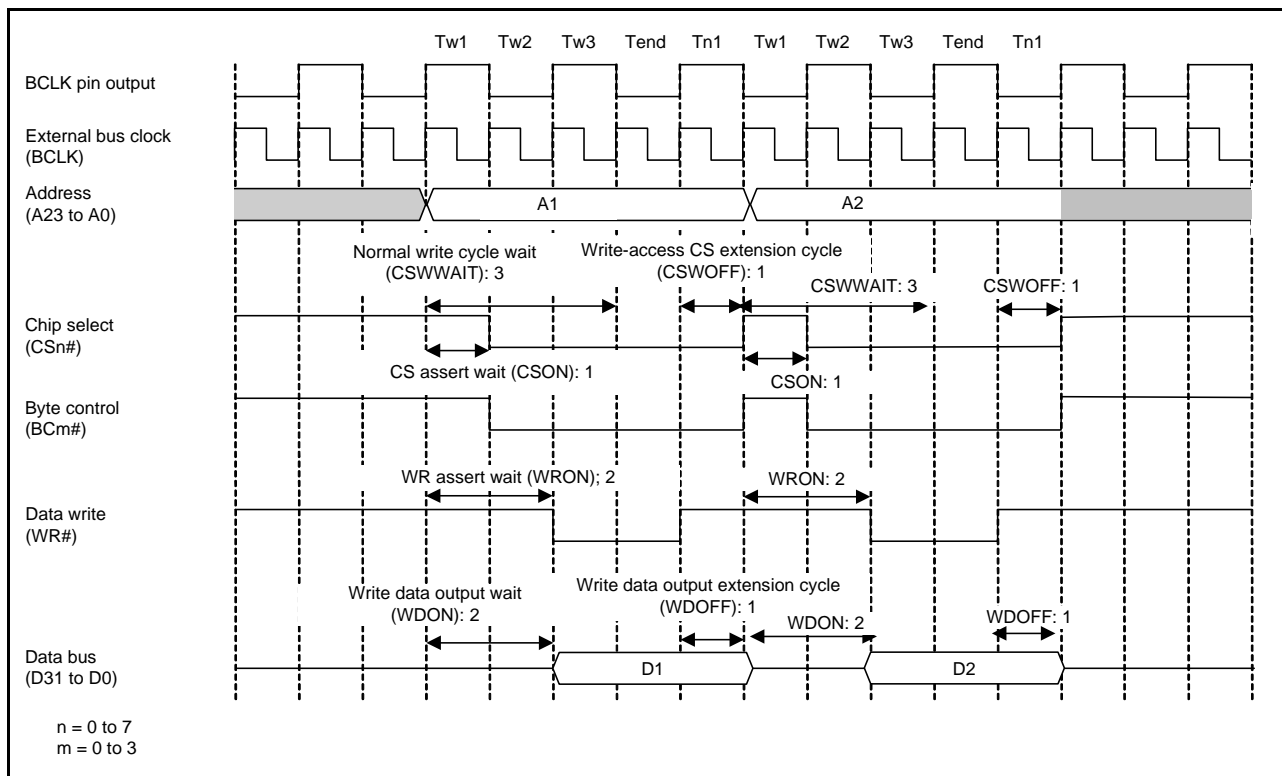


Figure 12.28 Example of Normal-Write Operation (when BCLK × 1/2 is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer)

(2) Page Access

When the PRENB and PWENB bits in CSnMOD are set to 1 to enable page-read and page-write access, respectively, the bus access for page access operations becomes page reading and writing. Specifically, page access can occur when two or more rounds of external bus access are required for a single transfer request from the bus master. However, normal access is made when the split accesses are not aligned or the access spreads over the 32-bit boundary. See figure 12.5 to figure 12.8 for the conditions under which page access occurs.

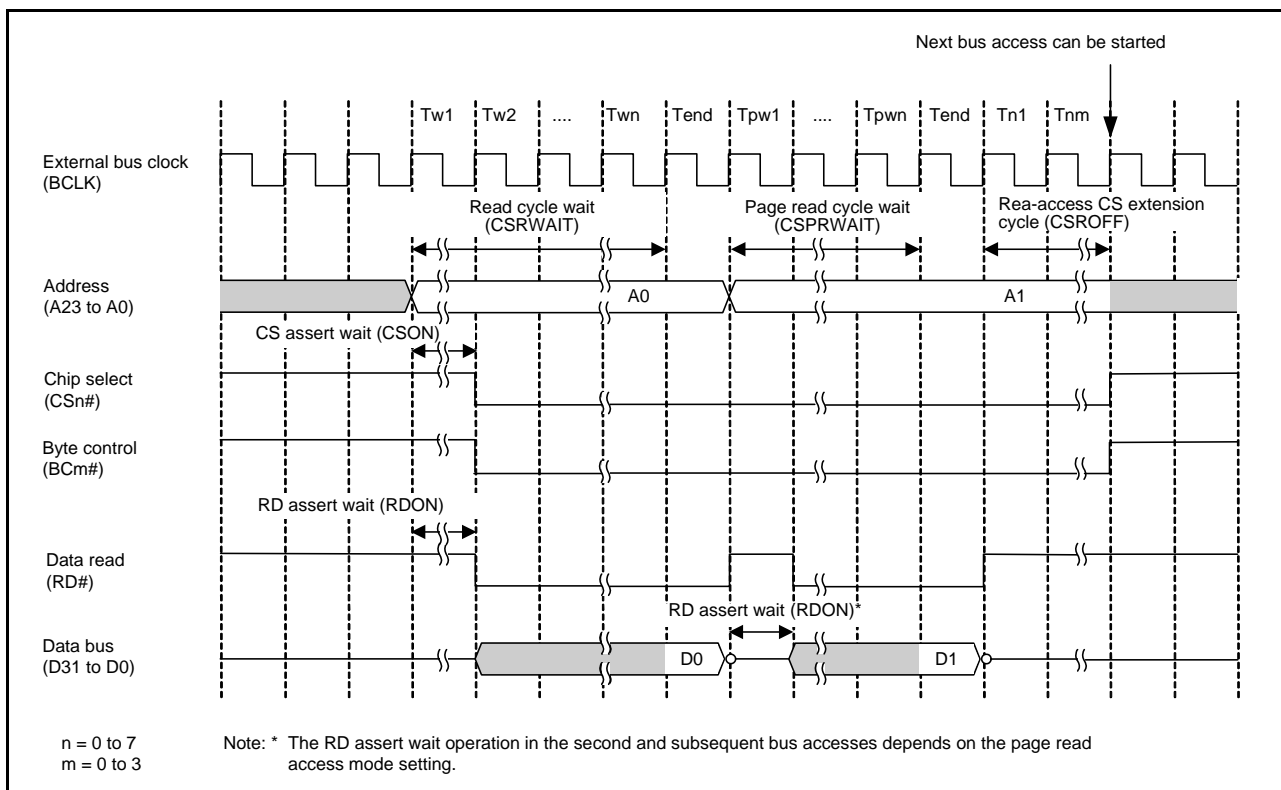


Figure 12.29 Page-Read Access Timing

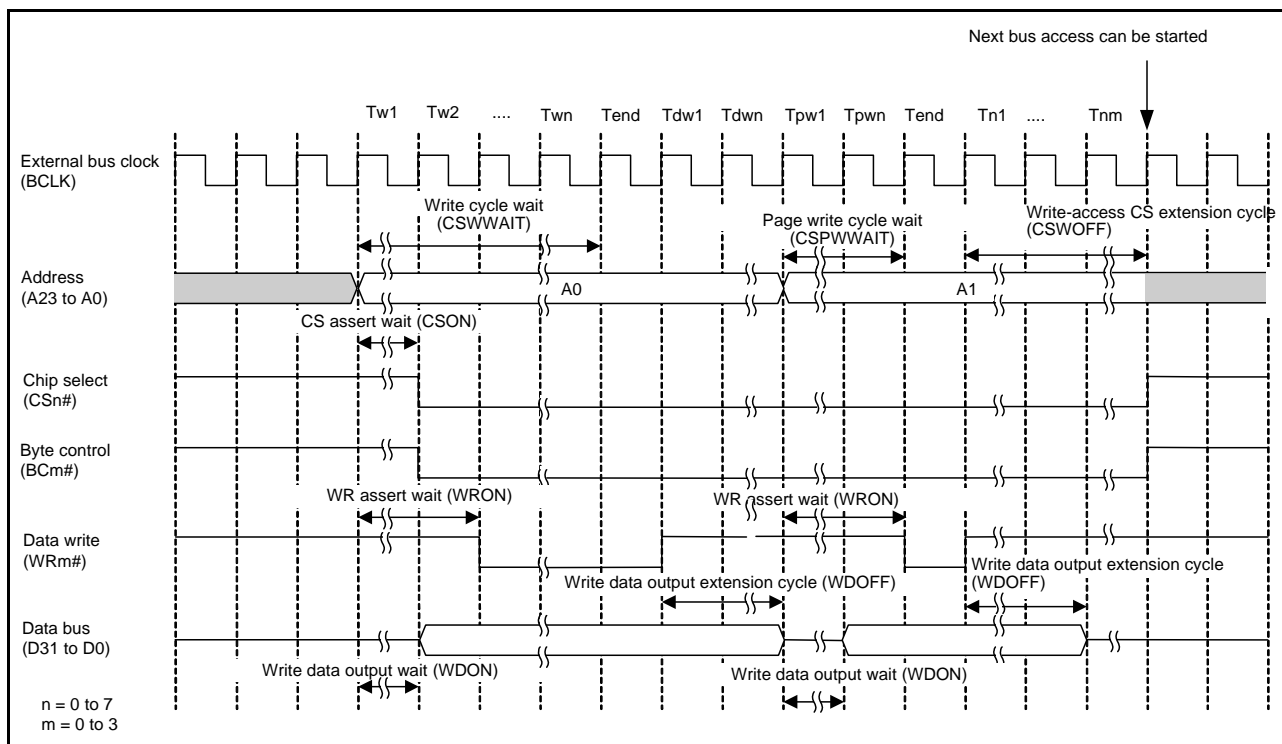


Figure 12.30 Page-Write Access Timing

Figure 12.31 and figure 12.32 depict examples of operations for access to a 16-bit bus space in 32 bits. The values of the wait control registers are example settings. In practice, the register settings will correspond to the specifications of connected devices.

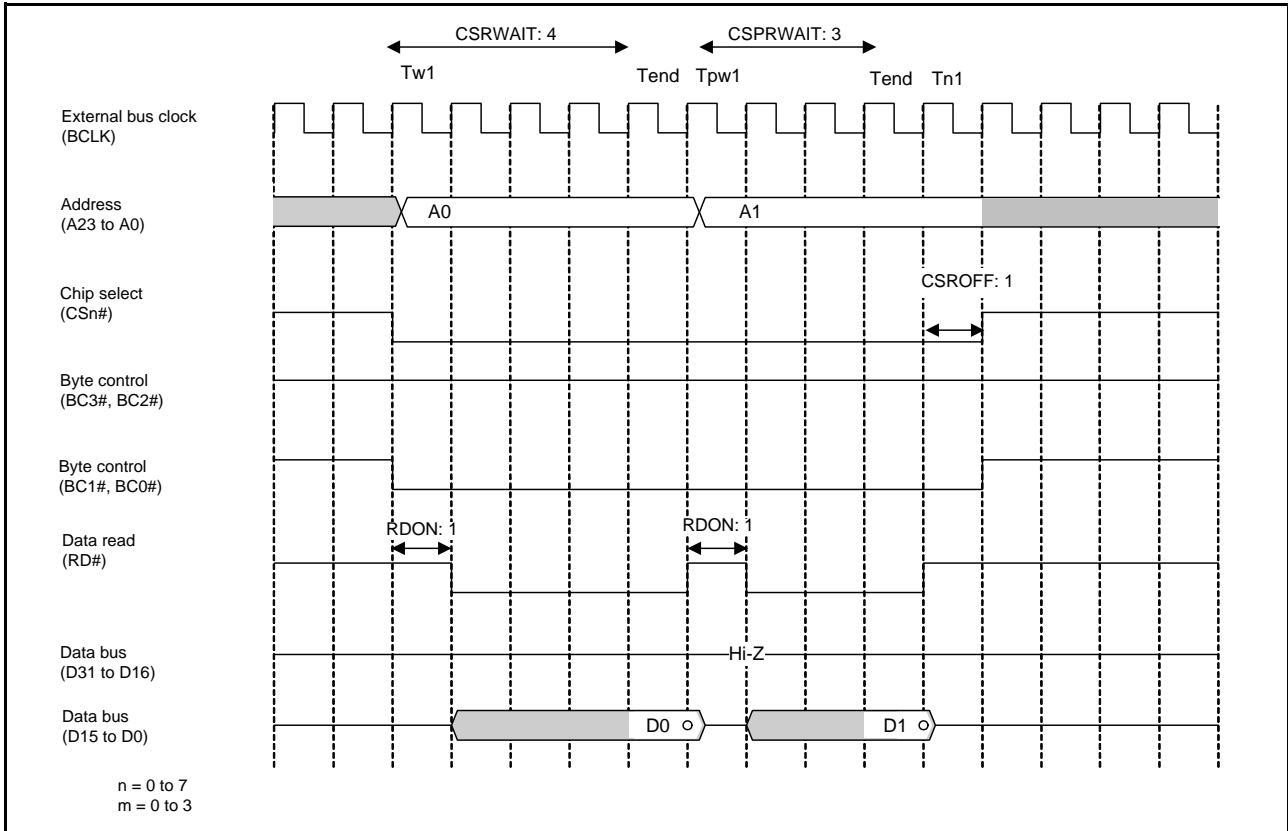


Figure 12.31 Example of Page-Read Access Operation (when 16-Bit Bus Space is Accessed in 32 Bits)

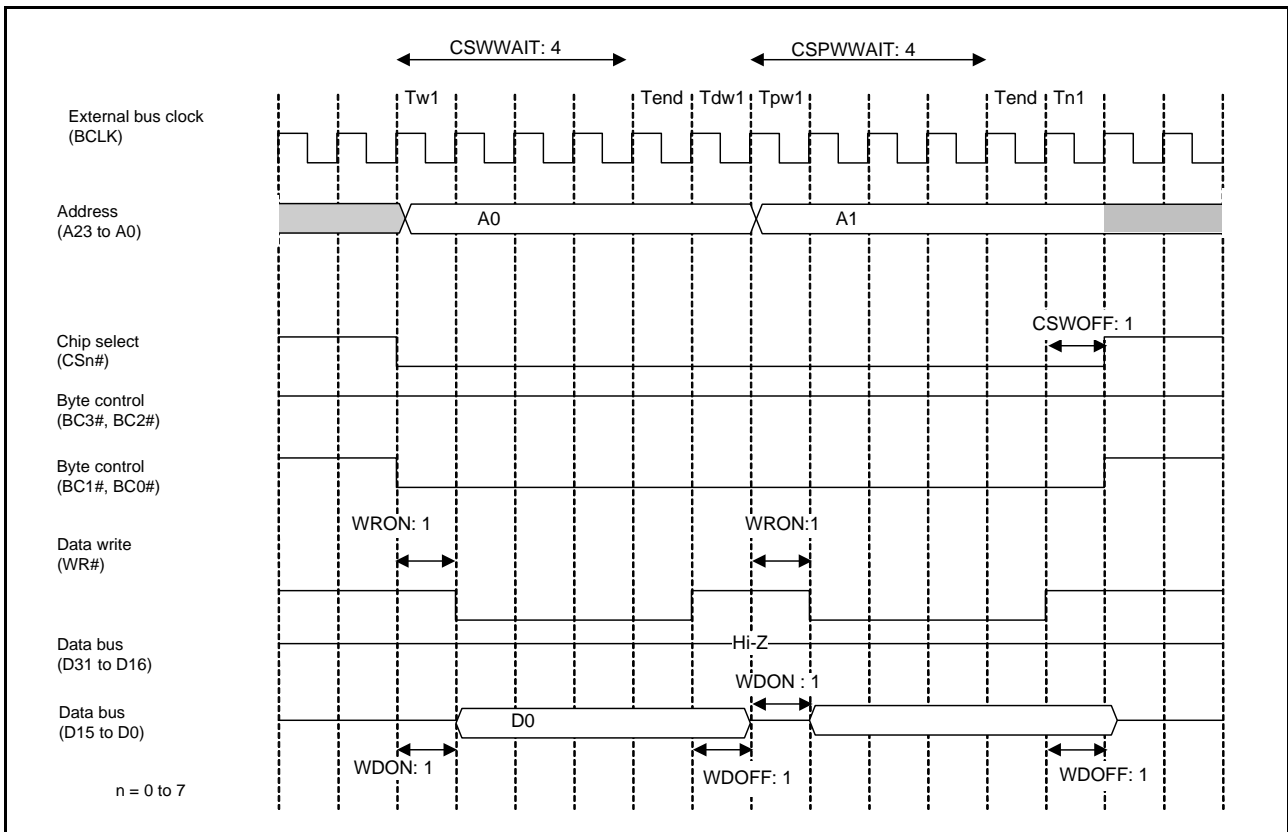


Figure 12.32 Example of Page-Write Access Operation (when 16-Bit Bus Space is Accessed in 32 Bits, in Single Write Strobe Mode)

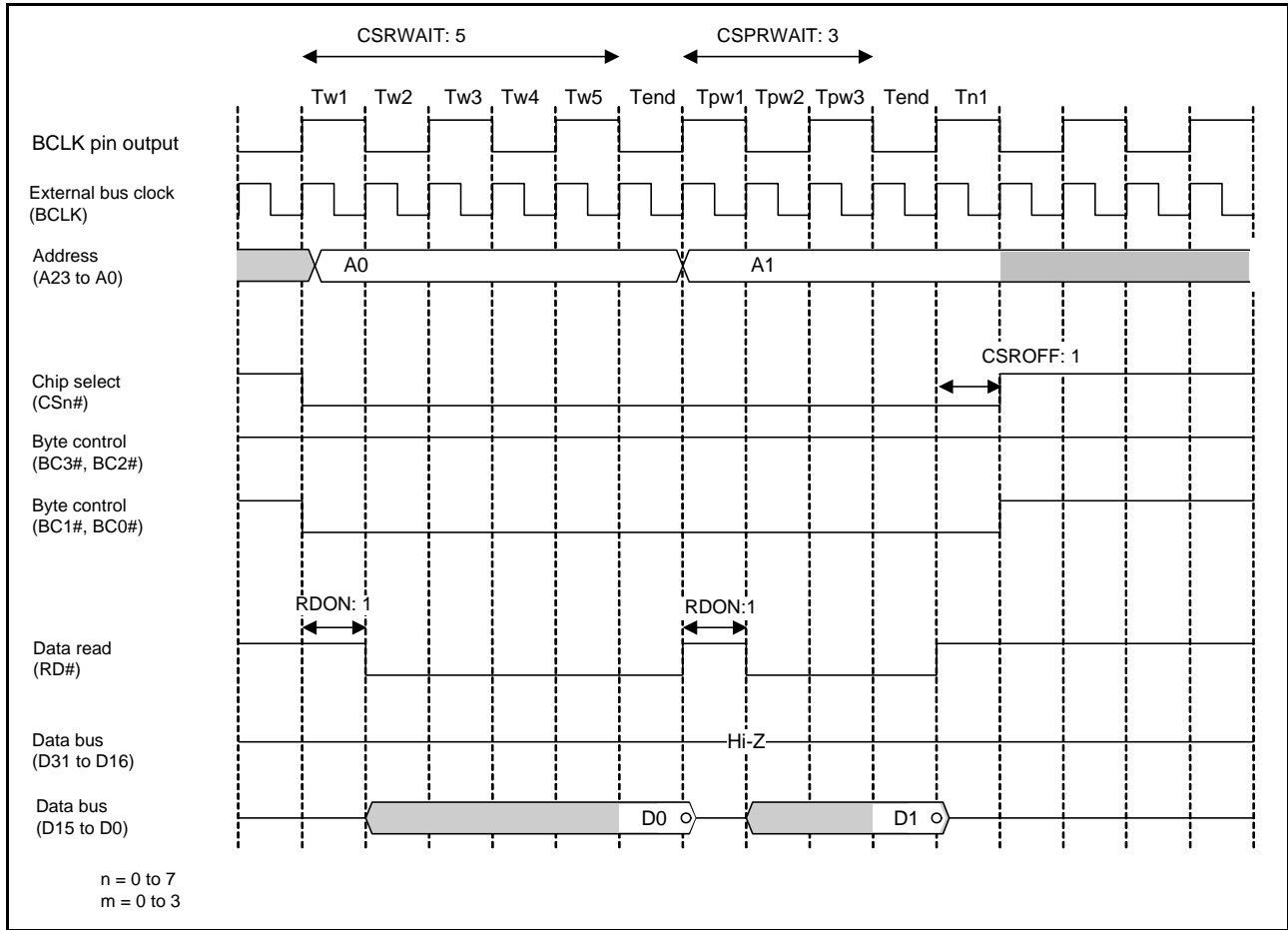


Figure 12.33 Example of Page Read Access Operation (BCLK Pin Output: BCLK = 1:2, when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer)

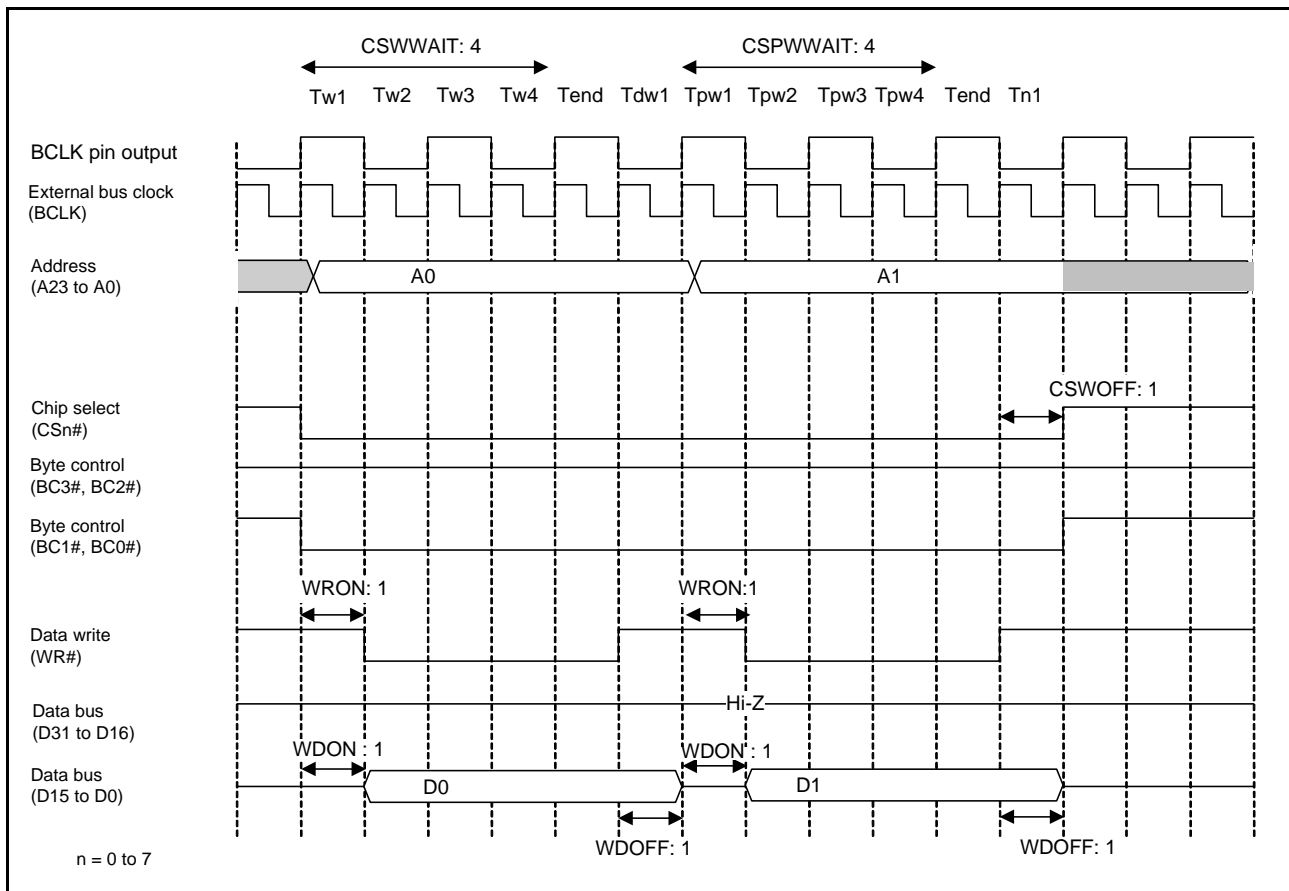


Figure 12.34 Example of Page Write Access Operation (BCLK Pin Output: BCLK = 1:2, when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode)

12.5.2 External Wait Function

Wait cycles can be extended by the WAIT# signal over the length of normal access cycle wait (specified by the CSRWAIT[4:0] and CSWAIT[4:0] bits in CSnWCR1) and page access cycle wait (specified by the CSPRWAIT[2:0] and CSPWWAIT[2:0] bits in CSnWCR1).

When external wait is enabled (the EWENB bit = 1 in CSnMOD), wait cycles are inserted while the WAIT# signal is held low. When external wait is disabled (the EWENB bit = 0 in CSnMOD), the WAIT# signal has no effect.

All wait cycles specified in CSnWCR1 are inserted independently of the WAIT# signal.

(1) Normal Access

Sampling of the WAIT# signal begins upon completion of the wait cycle (Tend) specified in CSnWCR1. The bus cycle is extended while the WAIT# signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT# signal becomes high.

(2) Page Access

The first access operation is the same as the normal access operation. Sampling of the WAIT# signal begins upon completion of the wait cycle (Tend) specified in the CSnWCR1 register. The bus cycle is extended while the WAIT# signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT# signal becomes high.

With respect to the second and subsequent read accesses, sampling of the WAIT# signal begins upon completion of the wait cycle of the page access (Tend). The wait cycle of the page access is extended while the WAIT# signal is held low, and ends (Tend) at the next cycle after the WAIT# signal becomes high.

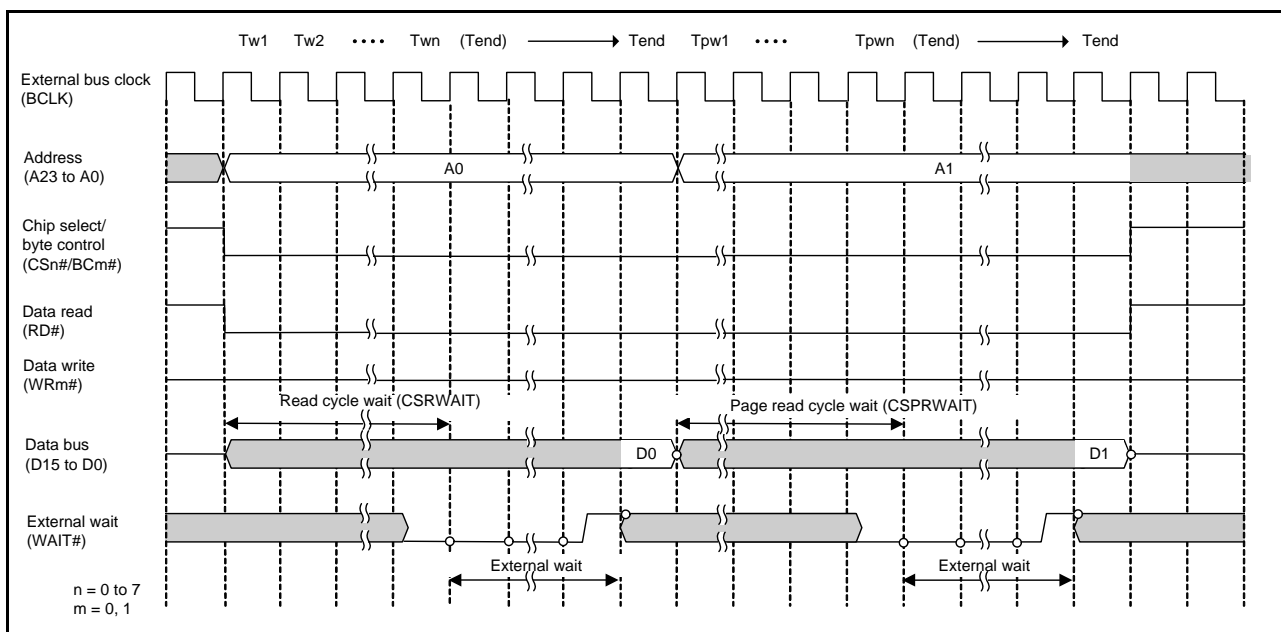


Figure 12.35 Example of External Wait Timing (Page-Read Access to 16-Bit Bus Space)

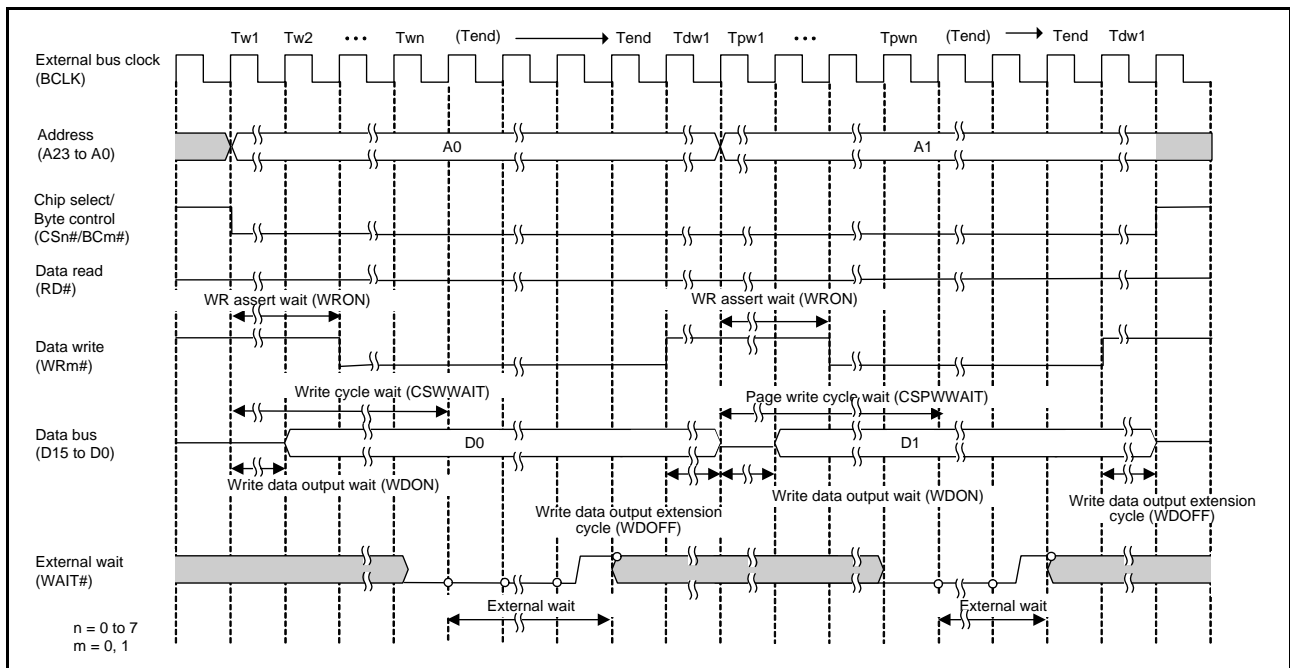


Figure 12.36 Example of External Wait Timing (Page-Write Access to 16-Bit Bus Space, in Byte Strobe Mode)

12.5.3 No Access State

When no external address space is accessed, CSn#, BCn#, WRn#, and RDn# signals are high and D31 to D0 are in the high-impedance state.

12.5.4 Insertion of Recovery Cycles

Recovery cycles can be inserted between consecutive rounds of external bus access. Conditions where recovery cycles can be inserted are described below.

Write access via the external bus follows read access via the external bus.

Read access to a different area follows read access via the external bus.

Read access via the external bus follows write access via the external bus.

Recovery cycles are not inserted between write access and subsequent write access.

The number of recovery cycles to be inserted after read cycles and write cycles can be set separately. The number of recovery cycles to be inserted after write cycles is set by the WRCV[3:0] bits in CSnREC for the write-accessed area in the previous bus cycle, and the number of recovery cycles to be inserted after read cycles is set by the RRCV[3:0] bits in CSnREC for the read-accessed area in the previous bus cycle. For example, when CS1 read access occurs after CS0 read access, the number of recovery cycles to be inserted between them is set by the RRCV[3:0] bits in CS0REC.

The recovery cycle starts at the end of the preceding bus cycle, i.e. when the CSn# signal (n = 0 to 7) is negated. A high-level period of the CSn# signal is inserted for the specified recovery cycle period starting from this point.

The CSn# signal for the next round of bus access is asserted immediately after the end of recovery cycles in the fastest case. Even if the next request for access to an external address space is generated during the recovery period, the next round of access over the external bus will start immediately after the end of recovery cycles.

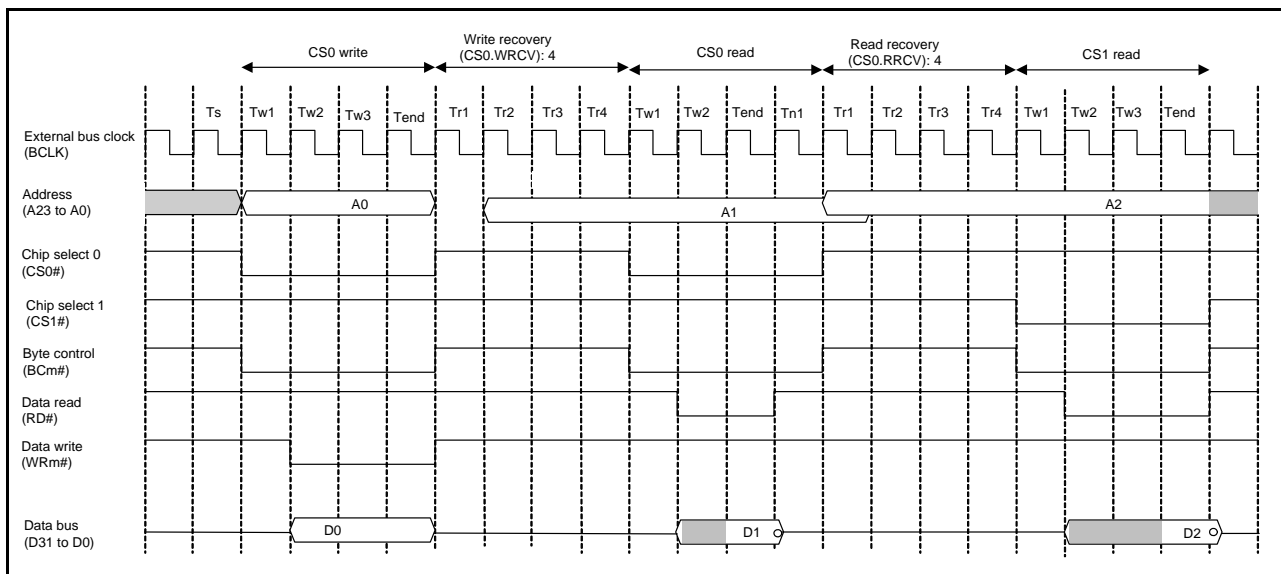


Figure 12.37 Example of Recovery Timing

12.5.5 Write Buffer Function

The internal main bus is released by writing data to the write buffer before the write access is completed, which allows the next round of bus access to start. However, if the following round of bus access is to an external address space or to a register of the external bus controller, it is suspended until the external bus operations already in progress are completed. Figure 12.38 shows an example of operation when the write-buffer function is in use. When this function is in use, if the next operation after an external write is internal access, the internal access (access to on-chip memory or a peripheral module) is executed in parallel with the external write, i.e. without waiting for completion of the latter operation.

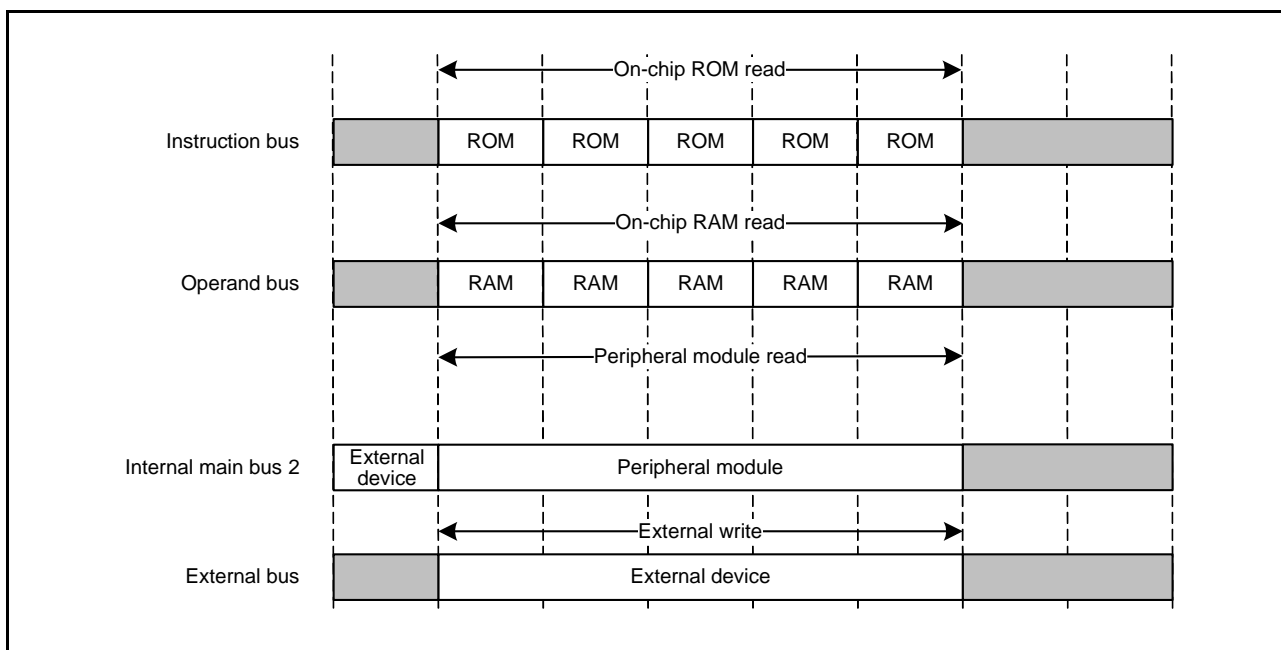


Figure 12.38 Example of Operation when the Write-Buffer Function is in Use

12.5.6 Notes on Usage

(1) Limitations at the Time of Normal and Page Access

Limitations that apply to various bits of CSn wait control register 1 (CSnWCNT1) and CSn wait control register 2 (CSnWCNT2) at the times of normal and page accesses are listed in table 12.11.

Even if the setting of the page-read access enable bit in the CSn mode register or the page-write access enable bit in the CSn mode register selects permission (CSnMOD.PRENB = 1 or CSnMOD.PWENB = 1), the first round of access for page access and the access that does not fall within the scope of page access are normal access operation, and thus limitations on normal access must be satisfied.

Table 12.11 Limitations at the Time of Normal and Page Access

Limitations at the Time of Normal Access		Limitations at the Time of Page Access	
Reading	Writing	Reading	Writing
CSON[2:0] ≤ CSRWAIT	1 ≤ WDON[2:0]	CSON[2:0] ≤ CSPRWAIT	1 ≤ WDON[2:0]
RDON[2:0] ≤ CSRWAIT	CSON[2:0] ≤ WDON[2:0]	RDON[2:0] ≤ CSPRWAIT	CSON[2:0] ≤ WDON[2:0]
CSON[2:0] ≤ RDON	CSON[2:0] ≤ CSWWAIT	CSON[2:0] ≤ RDON	CSON[2:0] ≤ CSPWWAIT
	WRON[2:0] ≤ CSWWAIT		WRON[2:0] ≤ CSPWWAIT
	WDON[2:0] ≤ CSWWAIT		WDON[2:0] ≤ CSPWWAIT
	WDOFF[2:0] ≤ CSWOFF		WDOFF[2:0] ≤ CSWOFF
	WDON[2:0] ≤ WRON		WDON[2:0] ≤ WRON
	CSON[2:0] ≤ WRON		CSON[2:0] ≤ WRON

(2) Limitations when BCLK × 1/2 is Selected through BCLK Pin Output Select Bit

When BCLK × 1/2 is selected through the BCLK pin output select bit, the external bus access cycle starts at the rising edge of the BCLK pin output. However, when two or more external bus access cycles are generated for a single transfer request from a bus master, the second or later external bus access cycle may start at the falling edge of the BCLK pin output depending on the wait cycle settings. Make appropriate register settings according to the specifications of the device to be connected.

(3) Limitation when a Pin is Multiplexed between A0 and BC0# Functions

When the A0 and BC0# pin functions share the same pin, single write strobe mode can be set only for the 16-bit and 32-bit bus spaces. In the 8-bit bus space, setting the single write strobe mode is prohibited.

(4) Limitations on EXDMAC Single Address Transfer Mode

- During the EXDMAC transfer in single address mode, the EDACK signal can be negated one cycle before the RD# signal is negated for read access or one cycle after the WR# signal is negated for write access through the settings of the EDACKn pin negate wait bit in the EXDMA output set register (EDMOMD.DACKW). Here, the CS# signal assertion and negation timing should be set so that the EDACK signal is enabled while the CS# signal is asserted. table 12.12 and table 12.13 show the limitations on the CSnWCR1 and CSnWCR2 register setting during the EXDMAC transfer in single address mode.
- To enable the EDACK signal output during the EXDMAC transfer in single address mode, the external wait function should be disabled (EWENB bit in CSnMOD = 0).
- When the external data read continuous assertion mode is specified (CSnMOD.PRMOD = 1) for page read access, the EXDMAC transfer in single address mode is prohibited; if such an attempt is made, correct operation is not guaranteed.

Table 12.12 Limitations on EXDMAC Single Address Transfer Mode (EDMOMD.DACKW = 0)

Limitations at the Time of Normal Access		Limitations at the Time of Page Access	
Reading	Writing	Reading	Writing
CSON[2:0] ≤ CSRWAIT	CSON[2:0] ≤ CSWWAIT	CSON[2:0] ≤ CSPRWAIT	CSON[2:0] ≤ CSPWWAIT
RDON[2:0] ≤ CSRWAIT	WRON[2:0] ≤ CSWWAIT	RDON[2:0] ≤ CSPRWAIT	WRON[2:0] ≤ CSPWWAIT
CSON[2:0] ≤ RDON	WDON[2:0] ≤ CSWWAIT	CSON[2:0] ≤ RDON	WDON[2:0] ≤ CSPWWAIT
1 ≤ RDON	WDOFF[2:0] ≤ CSWOFF	1 ≤ RDON	WDOFF[2:0] ≤ CSWOFF
	WDON[2:0] ≤ WRON		WDON[2:0] ≤ WRON
	CSON[2:0] ≤ WRON		CSON[2:0] ≤ WRON
	1 ≤ WRON		1 ≤ WRON

Table 12.13 Limitations on EXDMAC Single Address Transfer Mode (EDMOMD.DACKW = 1)

Limitations at the Time of Normal Access		Limitations at the Time of Page Access	
Reading	Writing	Reading	Writing
CSON[2:0] ≤ CSRWAIT	CSON[2:0] ≤ CSWWAIT	CSON[2:0] ≤ CSPRWAIT	CSON[2:0] ≤ CSPWWAIT
RDON[2:0] < CSRWAIT	WRON[2:0] ≤ CSWWAIT	RDON[2:0] < CSPRWAIT	WRON[2:0] ≤ CSPWWAIT
CSON[2:0] ≤ RDON	WDON[2:0] ≤ CSWWAIT	CSON[2:0] ≤ RDON	WDON[2:0] ≤ CSPWWAIT
1 ≤ RDON	WDOFF[2:0] ≤ CSWOFF	1 ≤ RDON	WDOFF[2:0] ≤ CSWOFF
	WDON[2:0] ≤ WRON		WDON[2:0] ≤ WRON
	CSON[2:0] ≤ WRON		CSON[2:0] ≤ WRON
	1 ≤ WRON		1 ≤ WRON
	1 ≤ WDOFF		1 ≤ WDOFF

(5) Prohibition of Access that Spans Areas of Address Space

Single access that spans two areas of the address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.

Single access that spans two areas of the address space is also prohibited in the EXDMAC block transfer in single address mode or cluster transfer, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single access in the EXDMAC block transfer in single address mode or cluster transfer.

(6) Restrictions on RMPA and String-Manipulation Instructions

- Although the external space has a per-area ending-switching facility (only for data), the allocation of data to be handled by RMPA or string-manipulation instructions to an area where the endian differs from that of the chip is prohibited, and operation is not guaranteed if this restriction is not observed. If data to be handled by RMPA or string-manipulation instructions are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.
- The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

(7) Restriction on Instruction Code

- When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

12.6 SDRAM Area Controller Operation

The following sections describe how the SDRAM area controller (SDRAMC) is enabled and the SDRAM bus width is set, which is followed by the description of SDRAMC operations including read, write, auto-refresh, self-refresh, initialization sequence, and mode register setting.

12.6.1 Enabling/Disabling SDRAM Access and Setting SDRAM Bus Width

SDRAM access can be enabled or disabled using the SDC control register (SDCCR). The SDRAM bus width can also be set using SDCCR.

Even when the operation of the SDRAM address space is disabled, refresh operation is available as long as self-refresh or auto-refresh operation is enabled.

12.6.2 No Access State

When no external address space is accessed, SDCS#, WEn#, RAS#, and CAS# signals are high.

12.6.3 Insertion of Recovery Cycles

When access to the SDRAM area follows access to the CS area, data recovery cycles are inserted for the CS area controller (CSC). If the number of recovery cycles for the CSC is 0, the ACT command for the next SDRAM access is issued immediately after negation of CSn# signal at the earliest. If the number of recovery cycles is not 0, the ACT command is issued two cycles after the specified recovery cycle period elapsed after negation of CSn# signal at the earliest. Since no data conflicts can occur during access to the SDRAM area, there is no need to set data recovery cycles for the SDRAM (fixed to zero cycle).

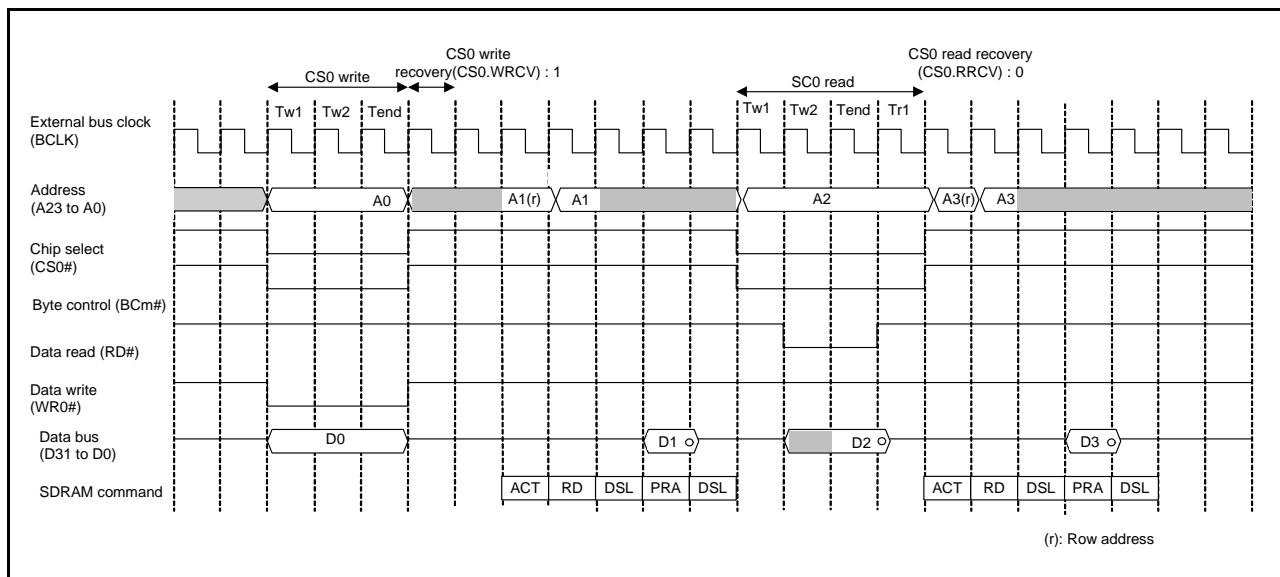


Figure 12.39 Example of Recovery Timing (for SDRAM Access)

12.6.4 Write Buffer Function

In write access, the internal main bus is released by writing data to the write buffer before the write access is completed, which allows the next round of bus access to start. However, if the following round of bus access is to an external address space or to a register of the external bus controller, it is suspended until the external bus operations already in progress are completed.

12.6.5 SDRAM Commands

The SDRAMC issues a command for each bus cycle to control SDRAM. Commands are defined by combination of SDCS#, RAS#, CAS#, WE#, CKE, and other signals.

Table 12.14 lists the commands issued by the SDRAMC.

Table 12.14 List of SDRAMC Commands

Name	Abbreviation	Command	SDCS#	RAS#	CAS#	WE#	CKE		BA1	BA0
							n - 1	n		
DESL	DSL	Device deselect	H	x	x	x	H	x	x	x
ACTV	ACT	Bank active	L	L	H	H	H	x	V	V
READ	RD	Read	L	H	L	H	H	x	V	V
WRIT	WRI	Write	L	H	L	L	H	x	V	V
PALL	PRA	All bank precharge	L	L	H	L	H	x	x	x
REF	RFA	Auto-refresh	L	L	L	H	H	x	x	x
MRS	MRS	Mode register set	L	L	L	L	H	x	L	L
SELF	RFS	Self-refresh entry	L	L	L	H	H	L	x	x
SELFx	RFX	Self-refresh end	H	x	x	x	L	H	x	x

[Legend] H: High level, L: Low level, V: Valid, x: Don't care. (High level or low level)
n: Command issue cycle, n - 1: One cycle before the command is issued.

12.6.6 Conditions for Setting SDRAMC Registers

SDRAMC registers should be modified only when all the corresponding conditions are satisfied as shown in table 12.15.

Table 12.15 Conditions for Register Modification

Function/Operation	Registers	Conditions
Self-refresh	SDSELF*1	SDRAM access is disabled. (SDCCR.EXENB = 0*2) Auto-refresh operation is enabled. (SDRFEN.RFEN = 1)
Auto-refresh	SDRFCR	Self-refresh operation is disabled. (SDSELF.SFEN = 0)
	SDRFEN	Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Initialization sequence	SDIR*1	The SDICR has not been set yet. SDRAM access is disabled. (SDCCR.EXENB = 0*2) Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) Self-refresh operation is disabled. (SDSELF.SFEN = 0)
	SDICR*1	SDRAM access is disabled. (SDCCR.EXENB = 0*2) Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Address register	SDADR	SDRAM access is disabled. (SDCCR.EXENB = 0*2) Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Timing register	SDTR	During self-refresh operation (SDSELF.SFEN = 1) or SDRAM access is disabled. (SDCCR.EXENB = 0*2) Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Mode register	SDMOD*1	SDRAM access is disabled. (SDCCR.EXENB = 0*2) Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Access mode register	SDAMOD	SDRAM access is disabled. (SDCCR.EXENB = 0*2) Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) Self-refresh operation is disabled. (SDSELF.SFEN = 0)

Note 1. Before modification, be sure to confirm that all the status bits in SDSR are 0.

Note 2. After writing 0 to the EXENB bit, confirm that the EXENB bit is cleared to 0.

12.6.7 Self-Refresh

Transition to or recovery from self-refresh mode can be controlled using the SDRAM self-refresh control register (SDSELF).

Immediately before transition to self-refresh mode, auto-refresh operation is performed. In self-refresh mode, the CKE signal is low. Immediately after recovery from self-refresh mode, the auto-refresh cycle is started.

Figure 12.40 and figure 12.41 show timing examples of transition to self-refresh mode and recovery from self-refresh mode, respectively.

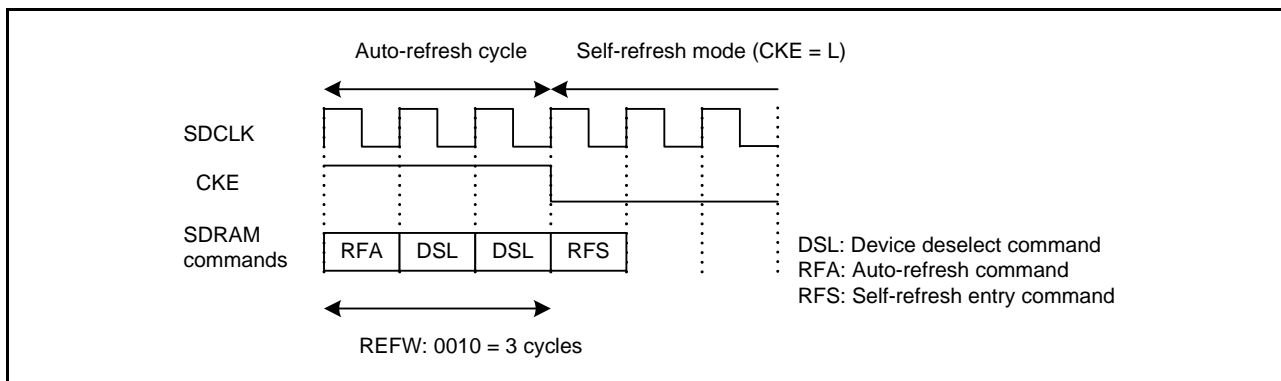


Figure 12.40 Timing Example of Transition to Self-Refresh Mode (when SDRFCR.REFW[3:0] = 0010b: 3 Cycles)

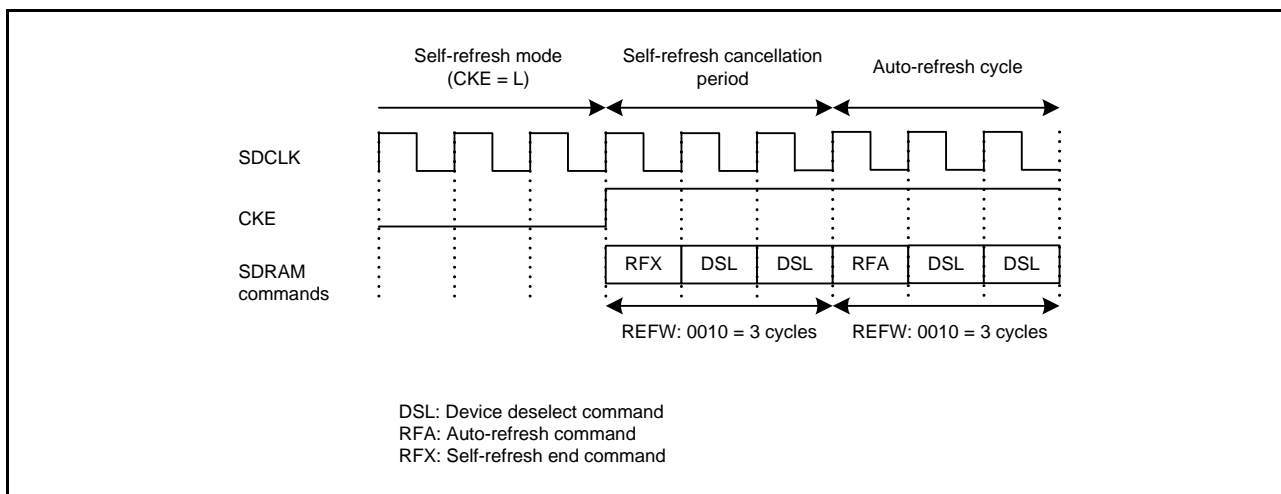


Figure 12.41 Timing Example of Recovery from Self-Refresh Mode

1. Self-Refresh in All-Module-Clock Stop Mode

When causing transition to self-refresh mode in all-module-clock stop mode, first cause transition to self-refresh mode according to the procedure shown in section 12.6.12.2, Procedure for Transition to and Recovery from Self-Refresh Mode, and then make settings to cause transition to all-module-clock stop mode.

After canceling all-module-clock stop mode, follow the procedure shown in section 12.6.12.2, Procedure for Transition to and Recovery from Self-Refresh Mode.

For details of transition and cancellation of all-module-clock stop mode, refer to section 9, Low Power Consumption.

2. Self-Refresh in Software Standby Mode

When causing transition to self-refresh mode in software standby mode, first cause transition to self-refresh mode according to the procedure shown in section 12.6.12.2, Procedure for Transition to and Recovery from Self-Refresh Mode, and then make settings to cause transition to software standby mode. In software standby mode, set the output port enable bit (OPE) in the standby control register (SBYCR) to 1 to hold the output state of the address bus and bus control signals.

After canceling software standby mode, follow the procedure shown in section 12.6.12.2, Procedure for Transition to and Recovery from Self-Refresh Mode.

For details of transition and cancellation of software standby mode, refer to section 9, Low Power Consumption.

3. Self-Refresh in Deep Software Standby Mode

Transition to deep software standby mode is performed via software standby mode. On transition to deep software standby mode from software standby mode, the state of pins remains unchanged. Therefore, transition to self-refresh mode in deep software standby mode can be made according to the same procedure as that in software standby mode.

In deep software standby mode, however, additional setting is necessary to cause transition to self-refresh mode; it is necessary to set the I/O port keep bit (IOKEEP) in the deep software standby control register (DPSBYCR) to 1.

Since the SDRAMC is internally reset by an internal reset signal when deep software standby mode is canceled, the SDRAM control registers need to be set again. After canceling software standby mode, follow the procedure shown below to cancel self-refresh mode. Figure 12.42 shows self-refresh timing in deep software standby mode.

For details of transition and cancellation of deep software standby mode, refer to section 9, Low Power Consumption.

1. In deep software standby mode, the CKE signal output remains low according to the IOKEEP setting in DPSBYCR.
2. Start clock supply to SDRAMC.
3. Set the SDRAM control registers (SDCMOD, SDAMOD, SDADR, and SDTR) again, which have been initialized by an internal reset upon transition to deep software standby mode, and then enable auto-refresh operation (RFEN bit in SDRFEN = 1).
4. Check that all the status bits in SDSR are cleared to 0 and set the SFEN bit in SDSELF to 1 to set self-refresh mode again.
5. Modify port settings for the SDRAM interface according to the procedure below.
 - (1)Set the enable bits for the SDRAM pins (MDSDE and DQM1E in PF6BUS) to 1 to set the ports for SDRAM again.
 - (2)Set the enable bit for the SDCLK pin (SDCLKE in PF6BUS) to 1 to enable SDCLK pin output again.
 - (3)Clear the IOKEEP bit in DPSBYCR to 0 to release the I/O ports from the held state.
6. Clear the PSTOP0 bit in SCKCR to 0 to start clock supply to the SDRAM via the SDCLK pin.
7. Check that all the status bits in SDSR are cleared to 0 and set the SFEN bit in SDSELF to 0 to cancel self-refresh mode.

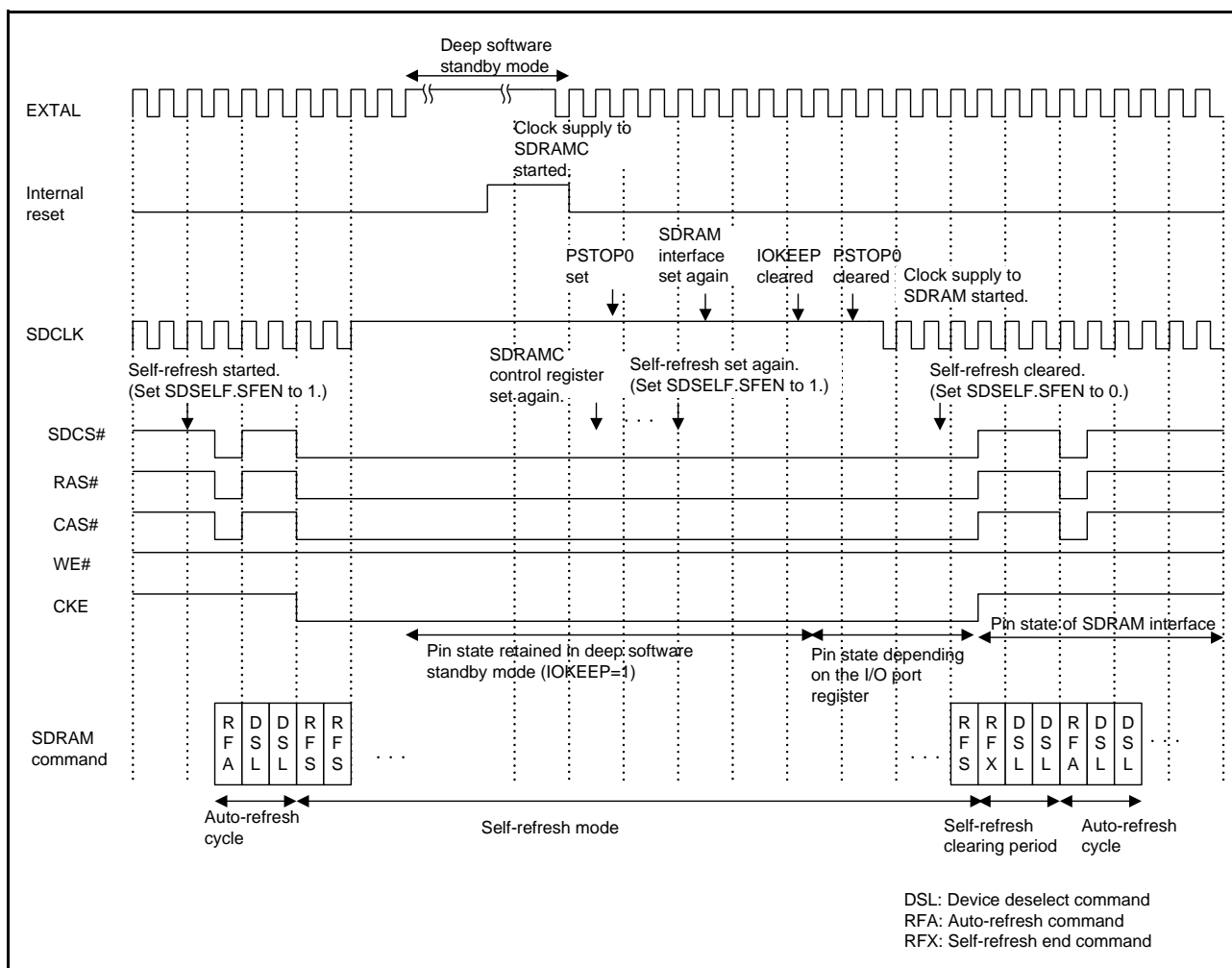


Figure 12.42 Timing Example of Self-Refresh Cycle (in Deep Software Standby Mode)

12.6.8 Auto-Refresh

The auto-refresh cycle can be started by setting the auto-refresh operation enable bit (RFEN) in the SDRAM auto-refresh control register (SDRFEN) to 1. Once the cycle is started, refresh requests are generated at fixed intervals determined by the refresh counter to start the auto-refresh cycle. However, since refresh requests are not accepted during read/write access, the auto-refresh cycle may be suspended. If an auto-refresh request is issued during consecutive access to the SDRAM, the auto-refresh cycle starts after bus access in response to a single transfer request from the bus master is completed.

If an SDRAM access and a refresh request are generated at the same time, the refresh request takes precedence. A CS area access and a refresh request can be made at the same if the SDCS#, RAS#, CAS#, WE#, and CKE signals, which are necessary for issuing the refresh command, are exclusively provided for SDRAM access.

When the RFEN bit in SDRFEN is set to 1 again after the auto-refresh cycle is started, a refresh request is generated. However, if a request is made during read/write access, a request is actually generated when access is completed.

The refresh counter is halted during self-refresh operation. After recovery from self-refresh mode, the auto-refresh cycle is started and the counter value is reset thus resuming the counter operation.

Figure 12.43 shows an example of the timing of an auto-refresh cycle. Figure 12.44 and figure 12.45 show examples of operation when an auto-refresh request is generated during single access and continuous access, respectively.

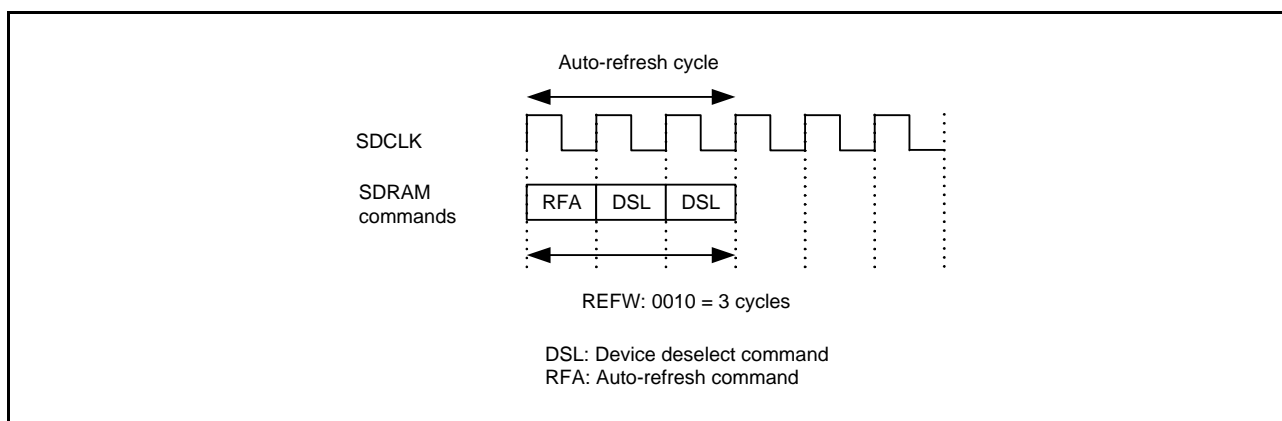


Figure 12.43 Timing Example of Auto-Refresh Cycle (1)

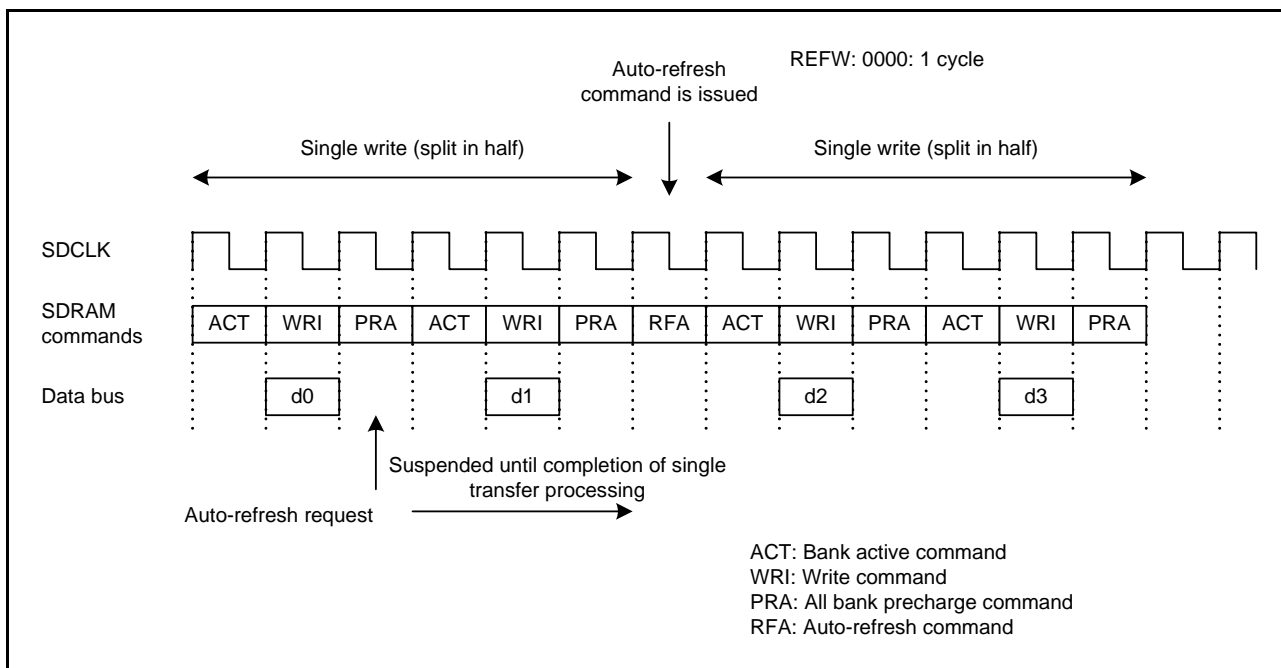


Figure 12.44 Timing Example of Auto-Refresh Cycle (2)
(Auto-Refresh Request is Made during Single Access)

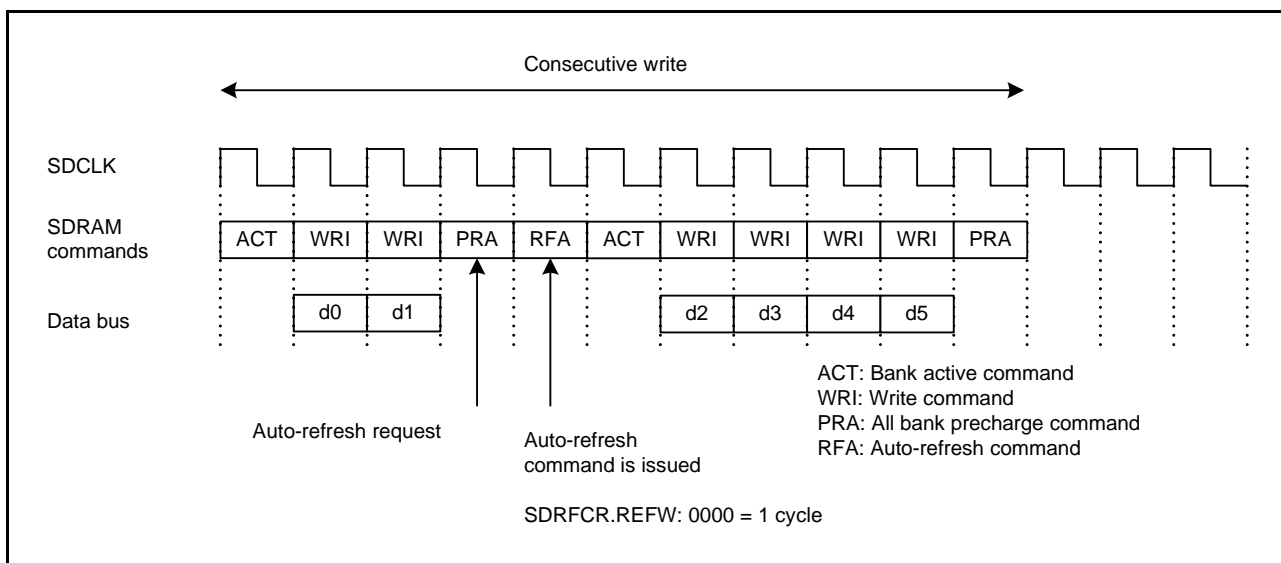


Figure 12.45 Timing Example of Auto-Refresh Cycle (3)
(Auto-Refresh Request is Made during Continuous Access)

12.6.9 Initialization Sequencer

The SDRAMC has a sequencer to issue SDRAM initialization commands. After a reset, the initialization sequence must be activated without fail; the operation is not guaranteed if the SDRAM is not initialized.

The SDRAM initialization sequencer issues a precharge-all-bank command followed by auto-refresh commands n times ($n = 1$ to 15). The SDRAM initialization sequence timing can be set using the SDRAM initialization register (SDIR). The SDRAM initialization sequence can be activated using the SDRAM initialization sequence control register (SDICR). These registers should be set only when the conditions listed in table 12.15, Conditions for Register Modification are satisfied.

Figure 12.46 shows a timing example of the SDRAM initialization sequence. When the ARFC[3:0] bits in SDIR are set so that auto-refresh operation is performed two or more times, auto-refresh cycles are repeated in the initialization sequence accordingly.

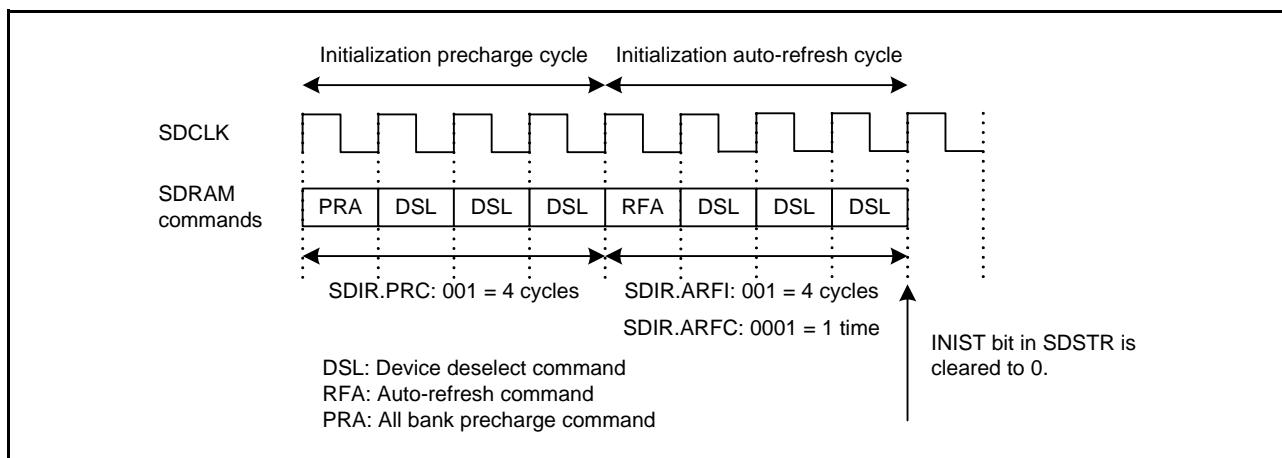


Figure 12.46 Timing Example of SDRAM Initialization Sequence

12.6.10 Read/Write Access

The SDRAMC controls read/write access in the following two modes.

- Single access mode: the row address is output each time data is accessed
- Consecutive access mode: when the same row address is accessed consecutively, only the column address is changed after the row address is output, enabling quick data access.

Consecutive SDRAM access is enabled by setting the continuous access enable bit (BE) in SDRAM access mode register (SDAMOD) to 1 in EXDMAC cluster transfer or block transfer in single address mode.

If the data size for a single transfer by the EXDMAC is less than the width of the external bus, and when bus access for a single transfer request ends once, in the same way as for non-aligned access, operation with consecutive access becomes possible.

When the above condition is not satisfied, the setting for consecutive-access mode is prohibited, and operation is not guaranteed if the setting is made.

Furthermore, setting the SDRAMC column-latency setting bits (CL[2:0]) in SDTR to 1 (CL = 1) in consecutive-access mode is prohibited, and operation is not guaranteed if this setting is made.

When the BE bit in SDAMOD is 0, single access is used in both cluster transfer for the EXDMAC and block transfer in single address mode.

(1) Single Access

Figures 12.47 and 12.48 show timing examples of single read, whereas figure 12.49 shows that of single write. The specific access timing depends on the SDRAM timing register (SDTR) settings. For details, refer to section 12.6.12.3, Timing Register Settings and Access Timing.

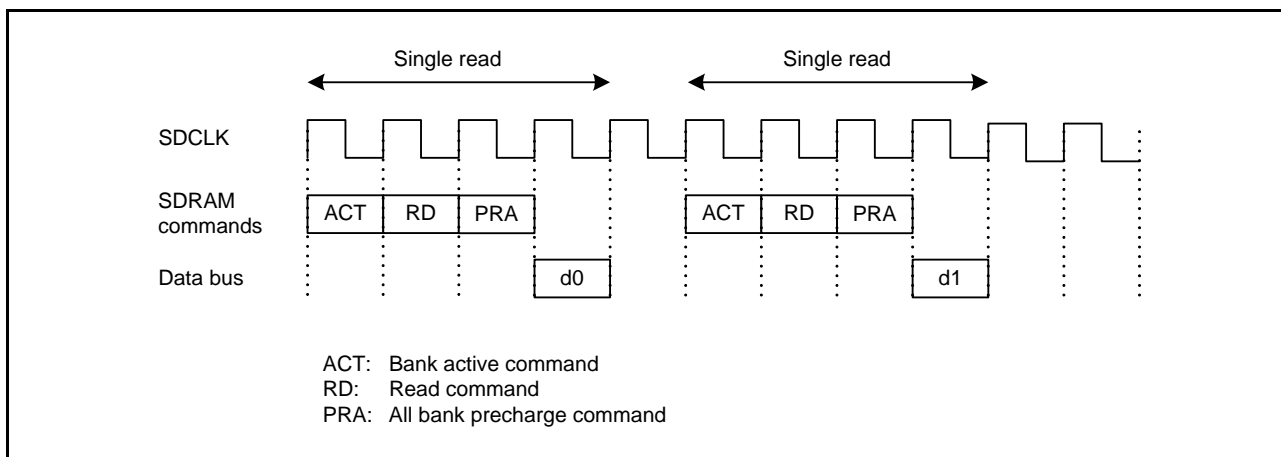


Figure 12.47 Timing Example of Single Read (SDTR.CL[2:0] = 010b: 2 Cycles)

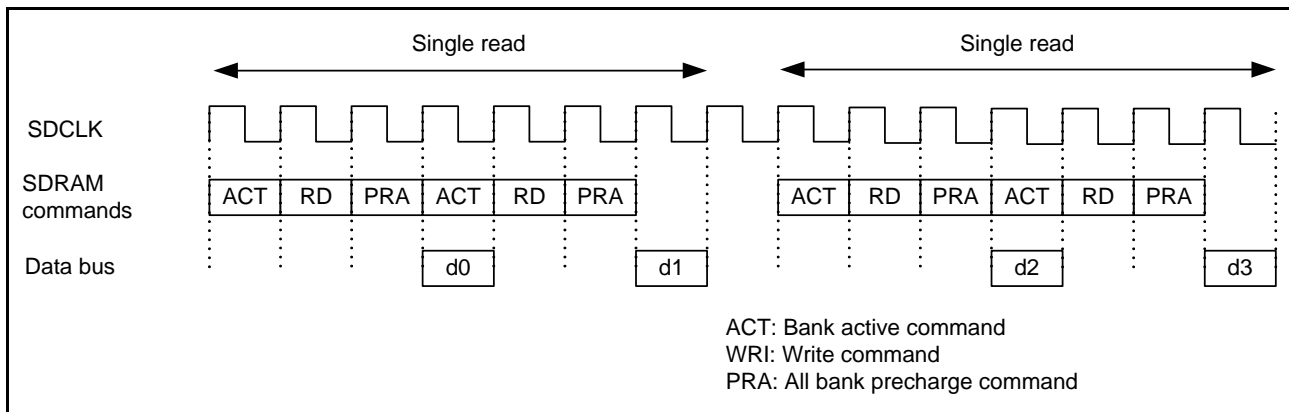


Figure 12.48 Timing Example of Single Read (Cluster Transfer by EXDMAC or Block Transfer in Single Address Mode with SDAMOD.BE = 0 and SDTR.CL[2:0] = 010b: 2 Cycles)

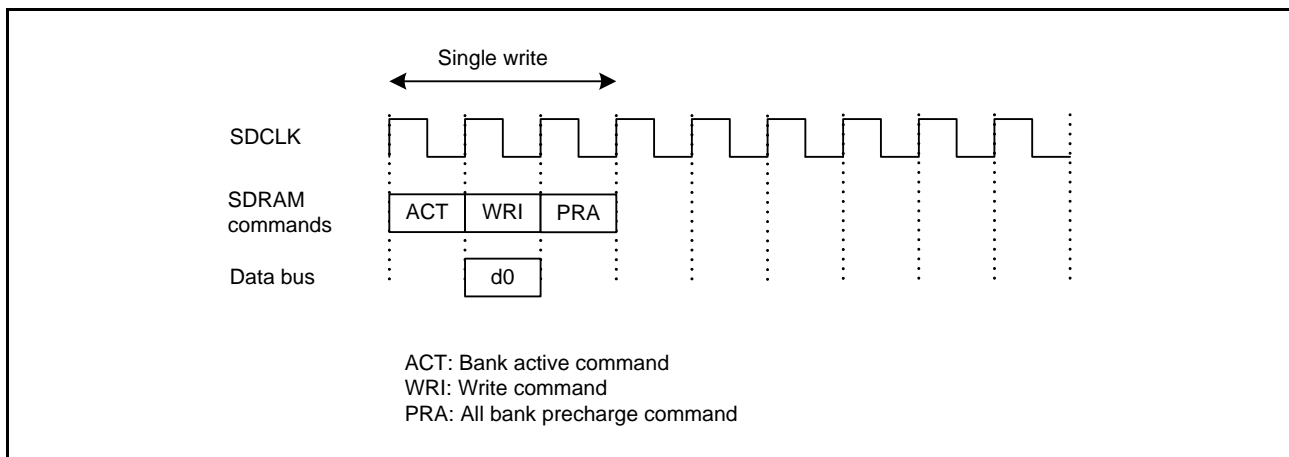


Figure 12.49 Timing Example of Single Write (when the Shortest Timing is Set)

(2) Consecutive Access

Figure 12.50 and figure 12.51 show timing examples of consecutive read and consecutive write for four data, respectively.

When the SDRAM row address changes during transfer, the pertinent row is automatically deactivated or activated appropriately. Figure 12.52 shows a timing example of consecutive write in which the row address changes.

The specific access timing depends on the SDRAM timing register (SDTR) settings. For details, refer to section 12.6.12.3, Timing Register Settings and Access Timing.

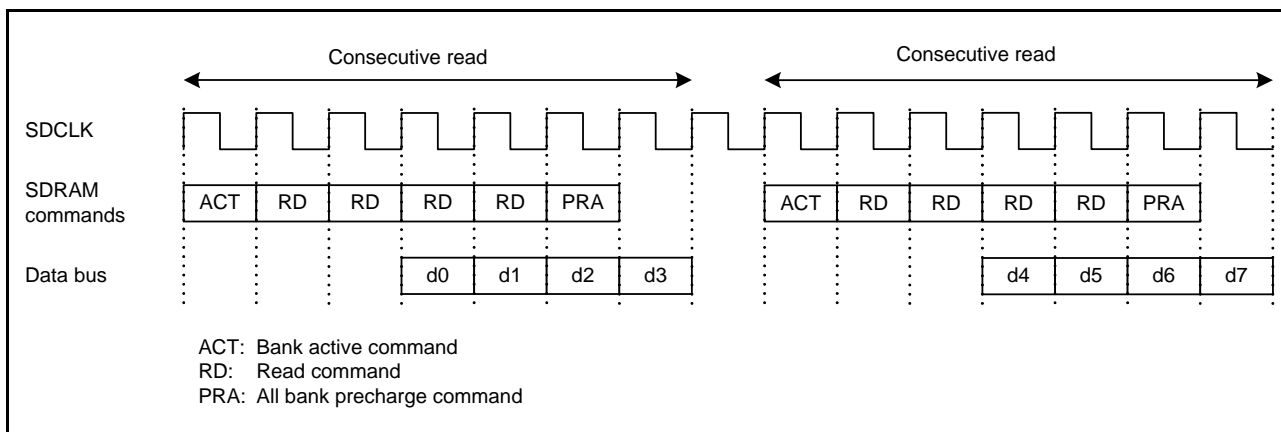


Figure 12.50 Timing Example of Consecutive Read (SDAMOD.BE = 1 and SDTR.CL[2:0] = 010b: 2 Cycles)

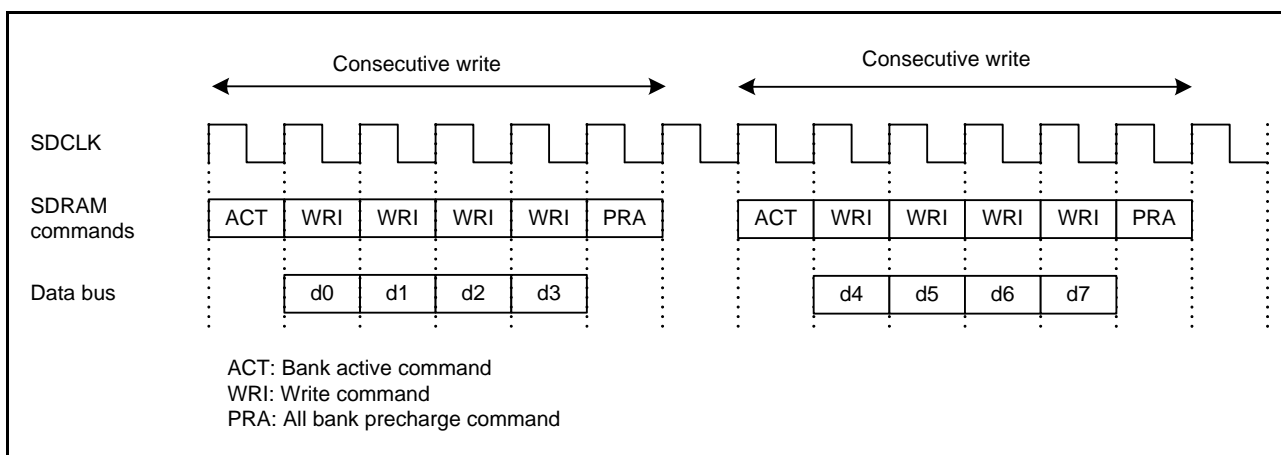


Figure 12.51 Timing Example of Consecutive Write (SDAMOD.BE = 1, when the Earliest Timing is Set)

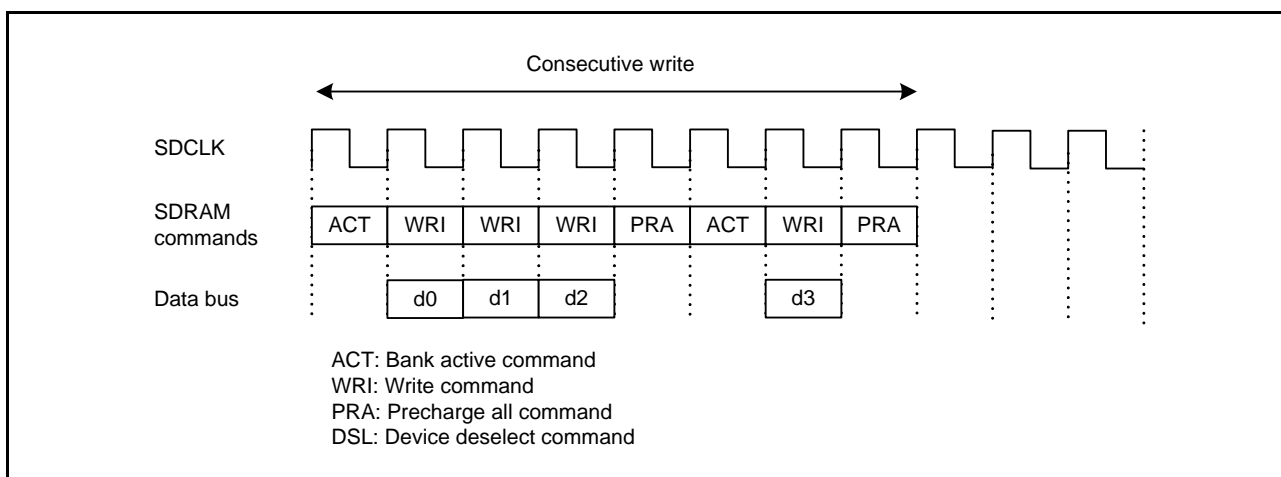


Figure 12.52 Timing Example of Consecutive Write (SDAMOD.BE = 1, when the Earliest Timing is Set) with More Than One Row Address Accessed)

12.6.11 Setting Mode Register

Setting the SDRAM mode register (SDMOD) allows the mode register set command to be issued to SDRAM and the value set in the MR[14:0] bits in SDMOD to be output to the lower bits of the address; specifically, to the A14 to A0 for 8-bit bus width, A15 to A1 for 16-bit bus width, and A16 to A2 for 32-bit bus width. Therefore, set the SDCCR.BSIZE[1:0] bits before setting the mode register, to determine the data bus width of the SDRAM.

Figure 12.53 shows the mode register setting timing.

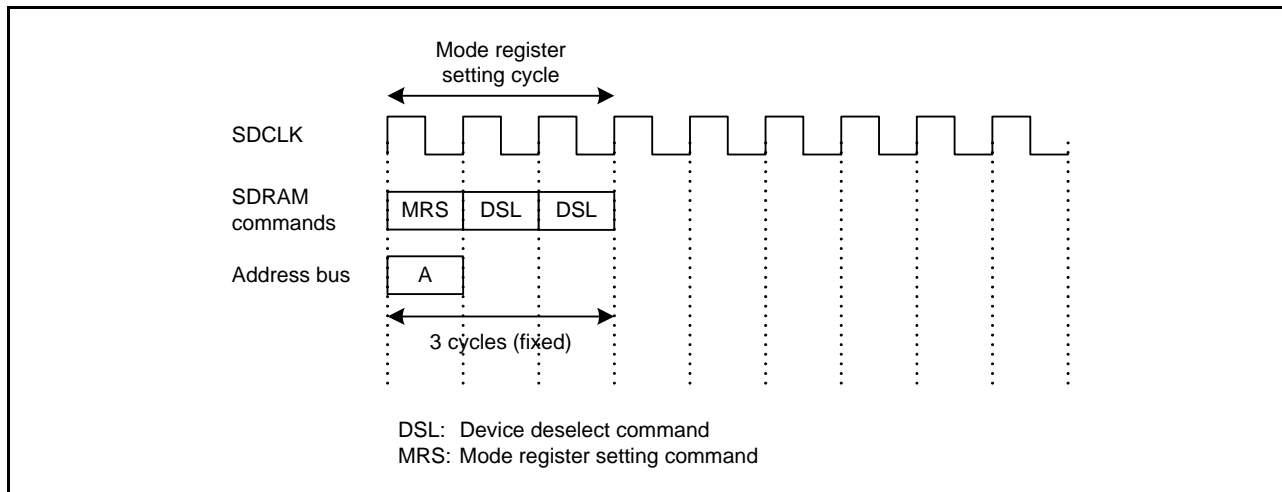


Figure 12.53 Mode Register Setting Timing

12.6.12 SDRAMC Setting Examples

This section describes the SDRAMC setting procedure, timing register setting examples, and procedure for transition to and recovery from self-refresh mode.

12.6.12.1 SDRAMC Access Procedure

Figure 12.54 shows the SDRAMC setting procedure.

The shown specifications including a power-up sequence may be different from that from the specifications of the SDRAM actually used; the system should be designed after reviewing the specifications of the SDRAM.

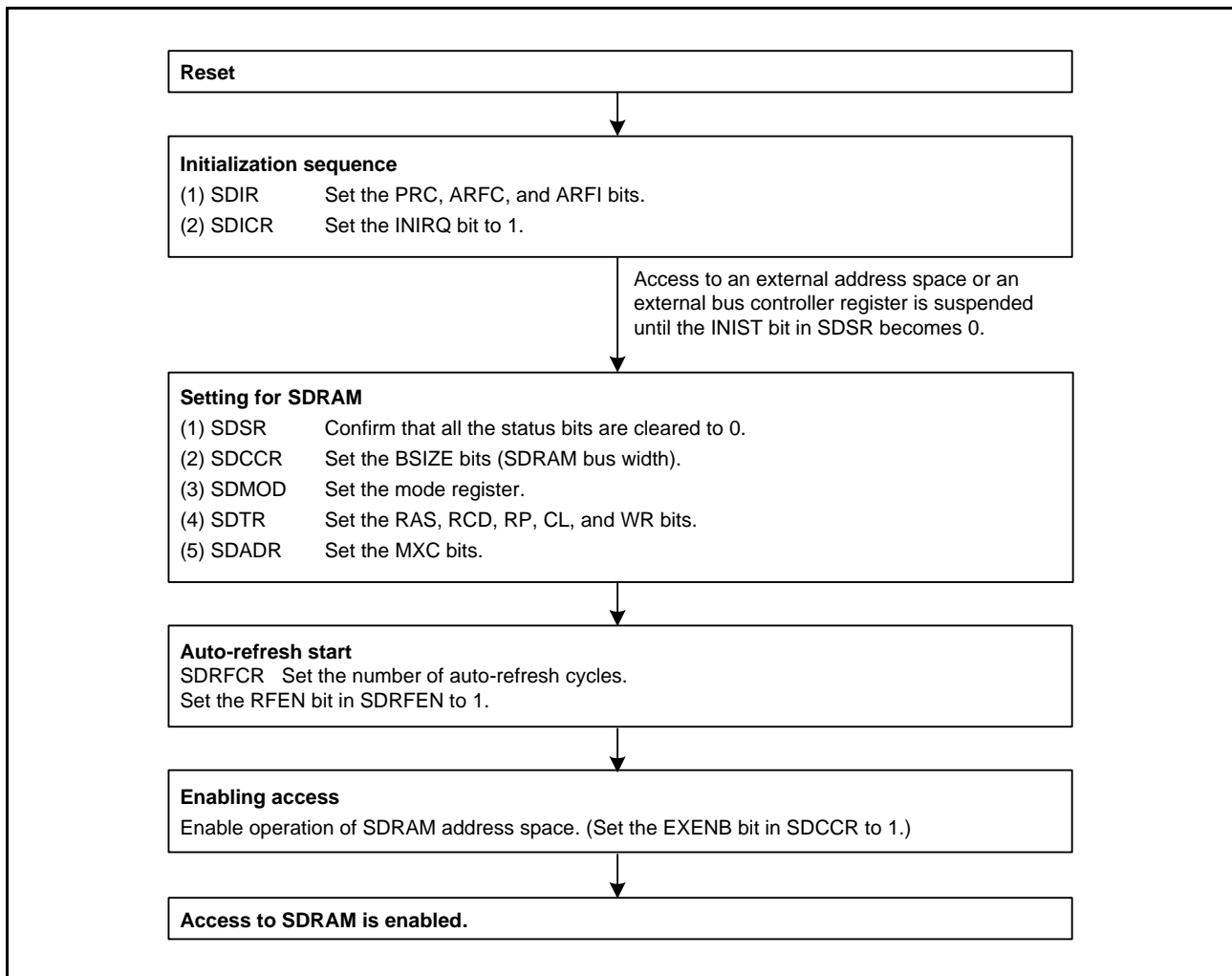


Figure 12.54 SDRAMC Setting Procedure

12.6.12.2 Procedure for Transition to and Recovery from Self-Refresh Mode

Figure 12.55 shows the procedure for transition to and recovery from self-refresh mode.

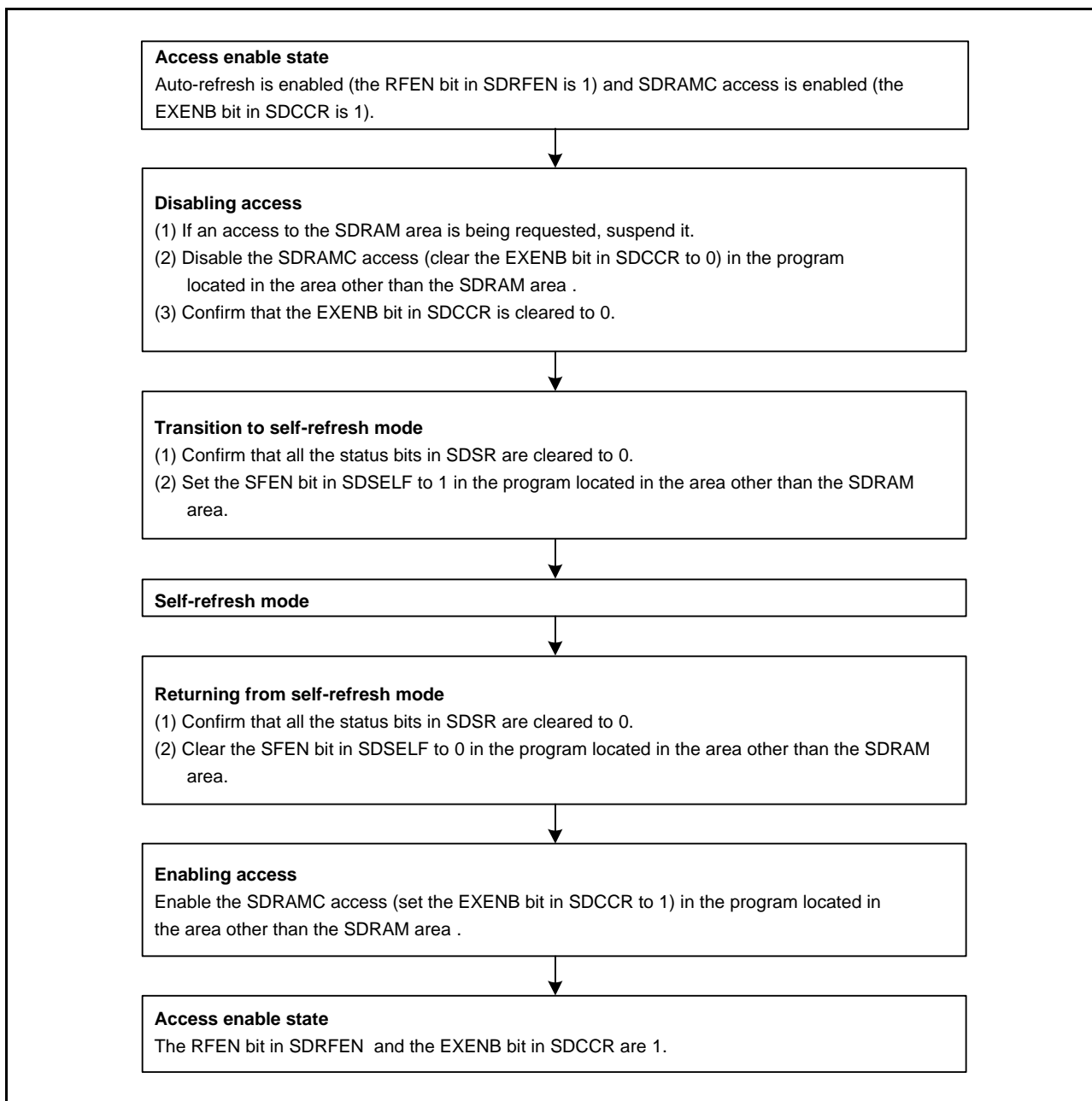


Figure 12.55 Procedure for Transition to and Recovery from Self-Refresh Mode

Note: Transition to and recovery from self-refresh mode requires SDRAM access to be disabled. Accordingly, transition to and recovery from self-refresh mode cannot be made during SDRAM access. The instructions below should be followed in programming.

- Before making transition to self-refresh mode, disable the access to the SDRAM area.
- During transition to self-refresh mode, self-refresh operation, and recovery from self-refresh mode, do not allow any operand access or instruction fetch (including prefetch) to the SDRAM area to be generated.

Figure 12.56 shows the procedure for transition to and recovery from self-refresh mode in deep software standby mode.

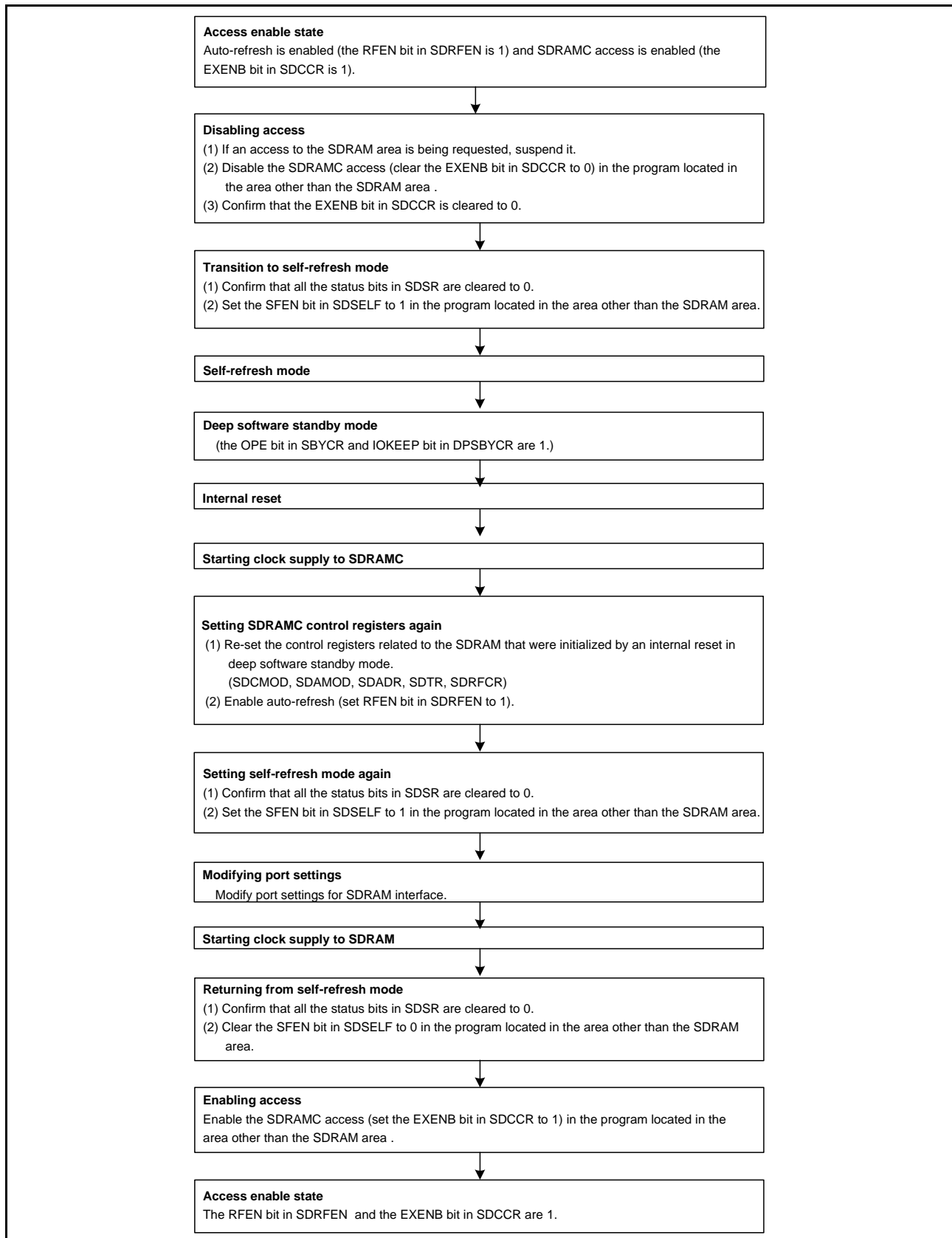


Figure 12.56 Procedure for Transition to and Recovery from Self-Refresh Mode in Deep Software Standby Mode

12.6.12.3 Timing Register Settings and Access Timing

This section describes the relationship between the read/write timing and the settings of the SDRAM timing register (SDTR).

(1) Single Read Timing Examples

Figure 12.57 to figure 12.61 show the relationship between the single read timing and the SDTR register settings. table 12.16 shows the correspondence between the figures and the SDTR register settings.

During read access, the next bus access is enabled two cycles after the read data becomes valid at the earliest. However, if two or more accesses occur for one transfer request, the next bus access is enabled one cycle after the read data becomes valid at the earliest, as shown in figure 12.61.

Table 12.16 Correspondence between Figure 12.57 to Figure 12.59 and STDR Register Settings (Single Read Timing)

Figure No.	RAS[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	CL[2:0] Settings	Number of Cycles
Figure 12.57	010	3	00	1	001	2	010	2
Figure 12.58	000	1	01	2	001	2	010	2
Figure 12.59	000	1	01	2	001	2	011	3
Figure 12.60 Figure 12.61	010	3	00	1	000	1	010	2

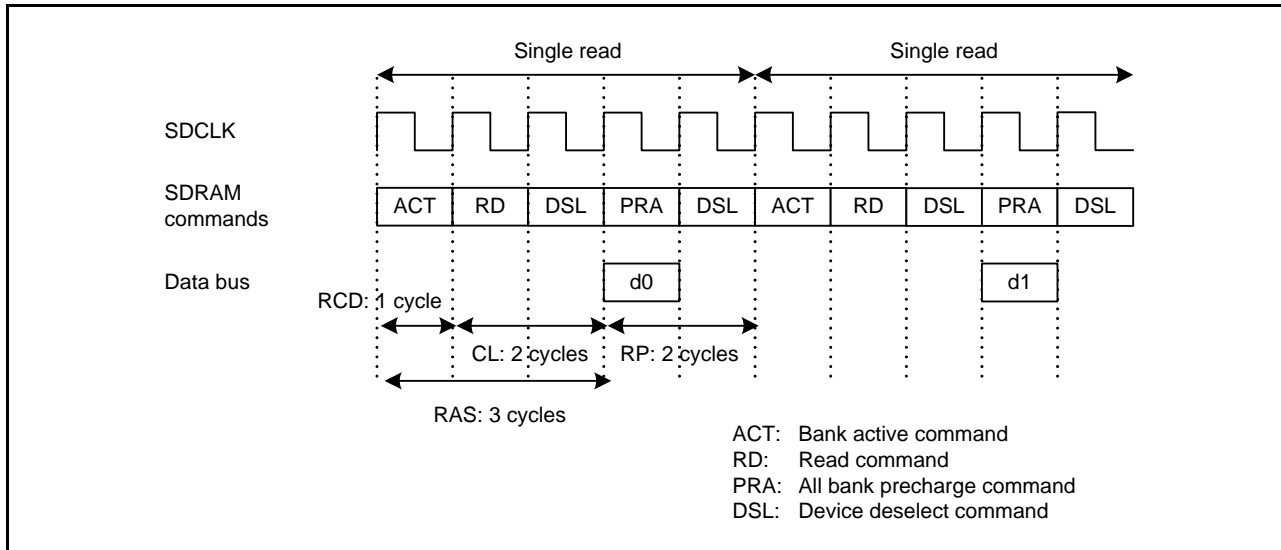


Figure 12.57 Timing Example of Single Read (1)

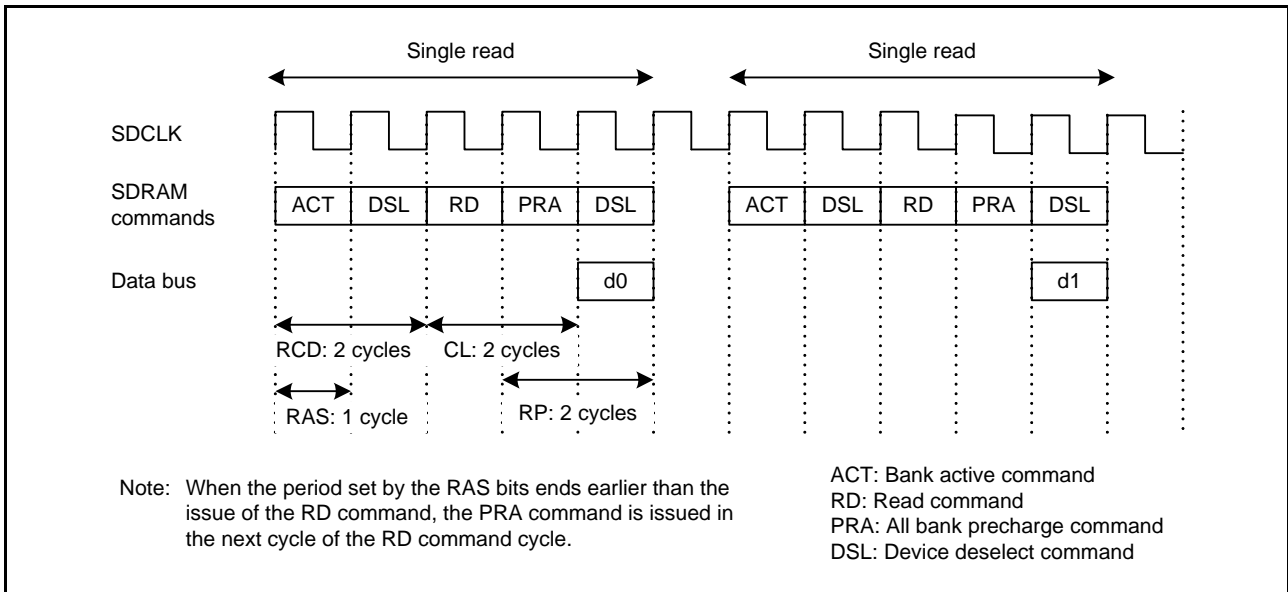


Figure 12.58 Timing Example of Single Read (2)

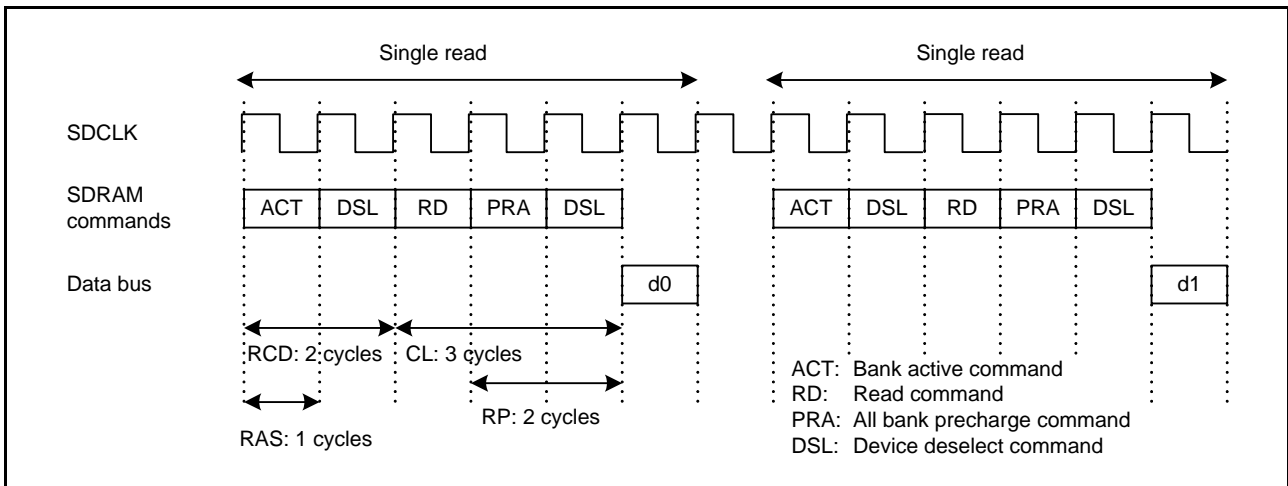


Figure 12.59 Timing Example of Single Read (3)

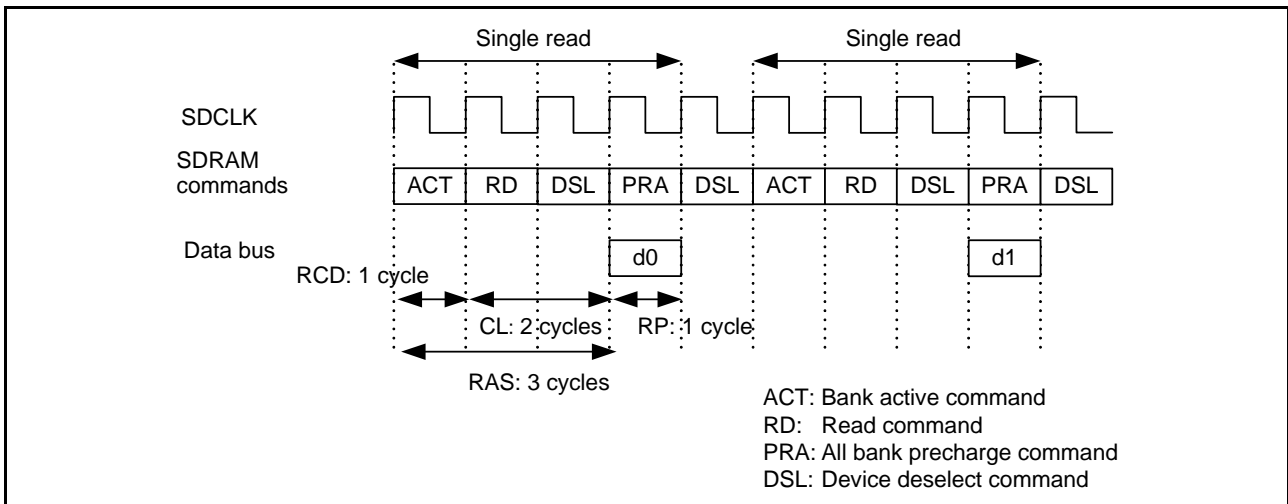


Figure 12.60 Timing Example of Single Read (4)

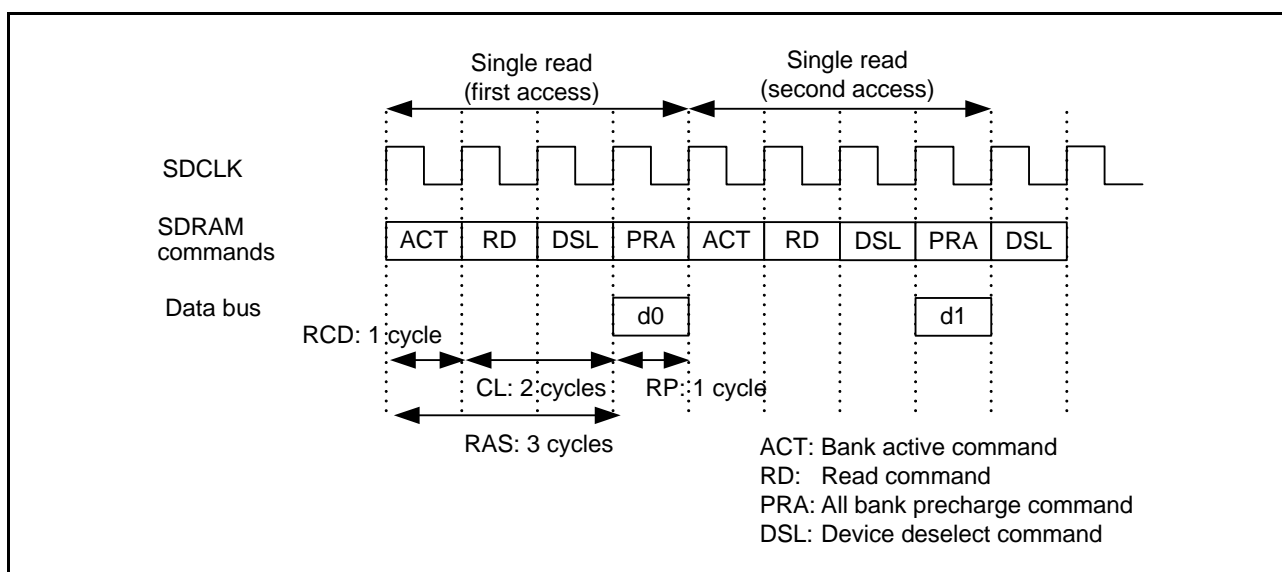


Figure 12.61 Timing Example of Single Read (5) (Two Bus Accesses Occur for One Transfer Request)

(2) Single Write Timing Examples

Figure 12.62 to figure 12.66 show the relationship between the single write timing and the SDTR register settings. Table 12.17 shows the correspondence between the figures and the SDTR register settings.

Table 12.17 Correspondence between Figure 12.60 to Figure 12.62 and STDR Register Settings (Single Write Timing)

Figure No.	RAS[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	WR Settings	Number of Cycles
Figure 12.62	010	3	00	1	001	2	0	1
Figure 12.63	000	1	01	2	001	2	0	1
Figure 12.64	000	1	01	2	001	2	1	2
Figure 12.65, Figure 12.66	010	3	00	0	000	2	0	1

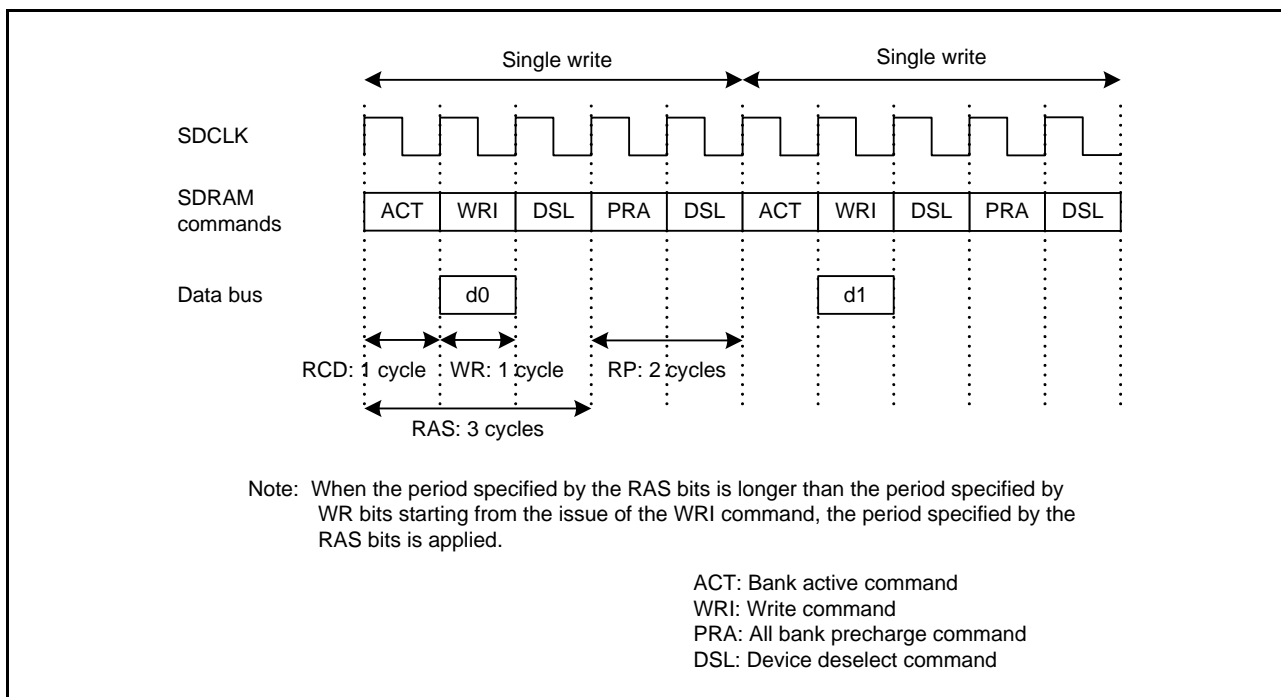


Figure 12.62 Timing Example of Single Write (1)

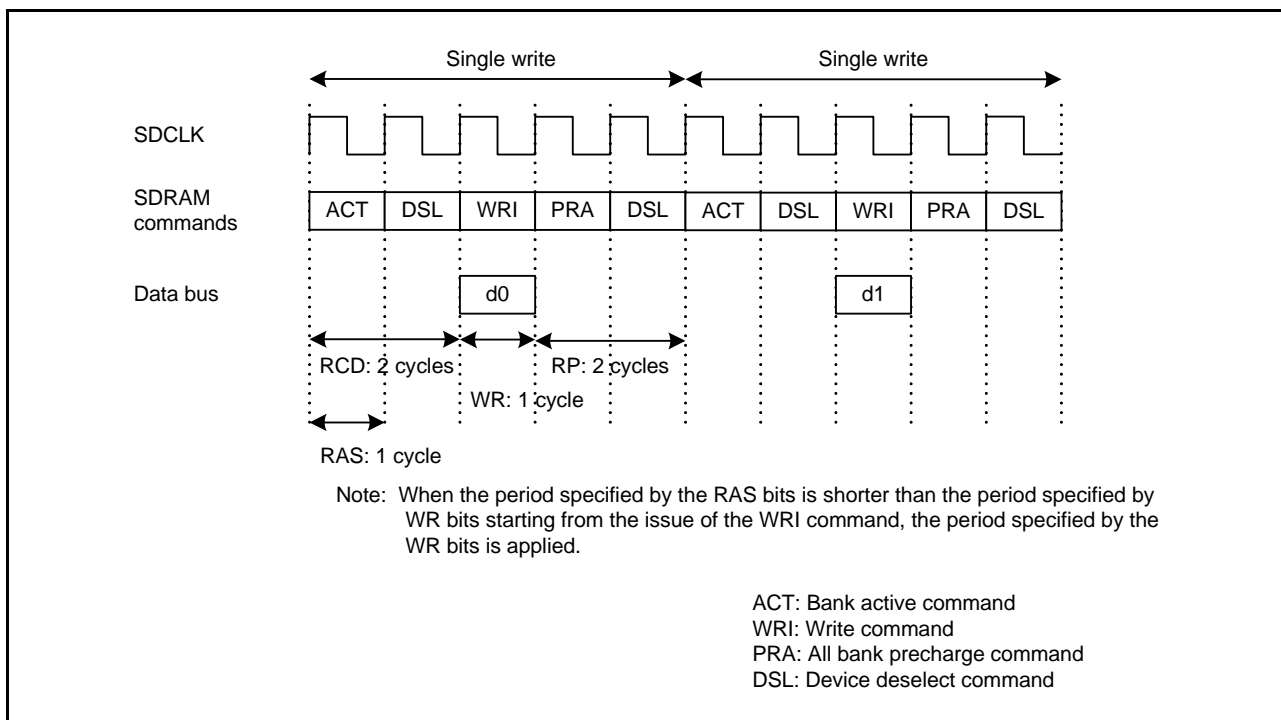


Figure 12.63 Timing Example of Single Write (2)

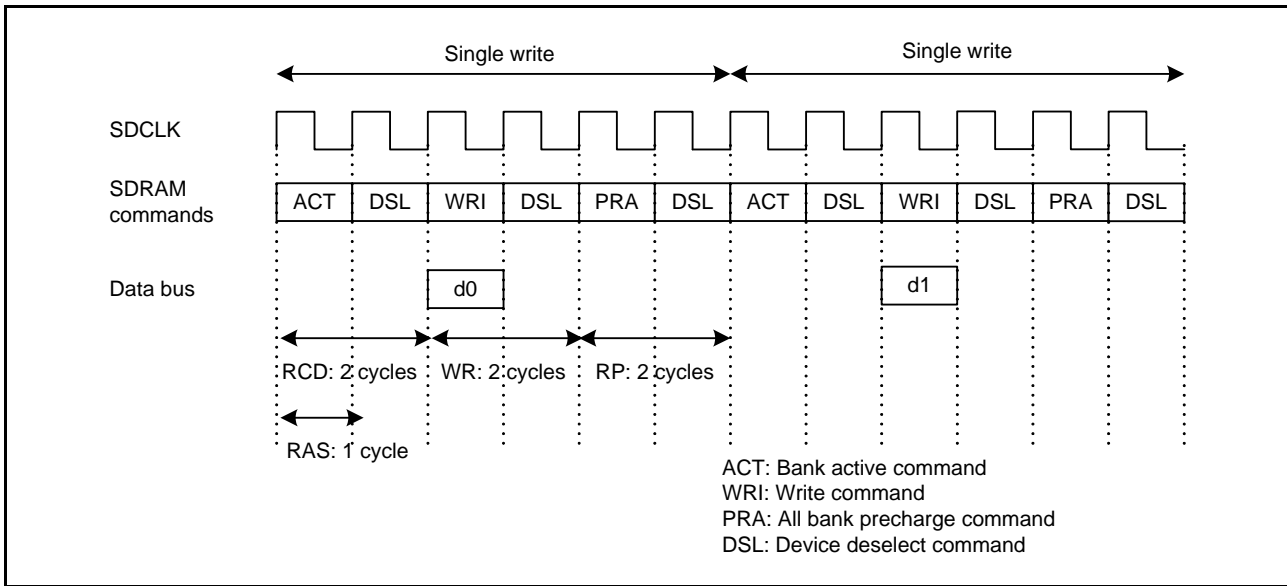


Figure 12.64 Timing Example of Single Write (3)

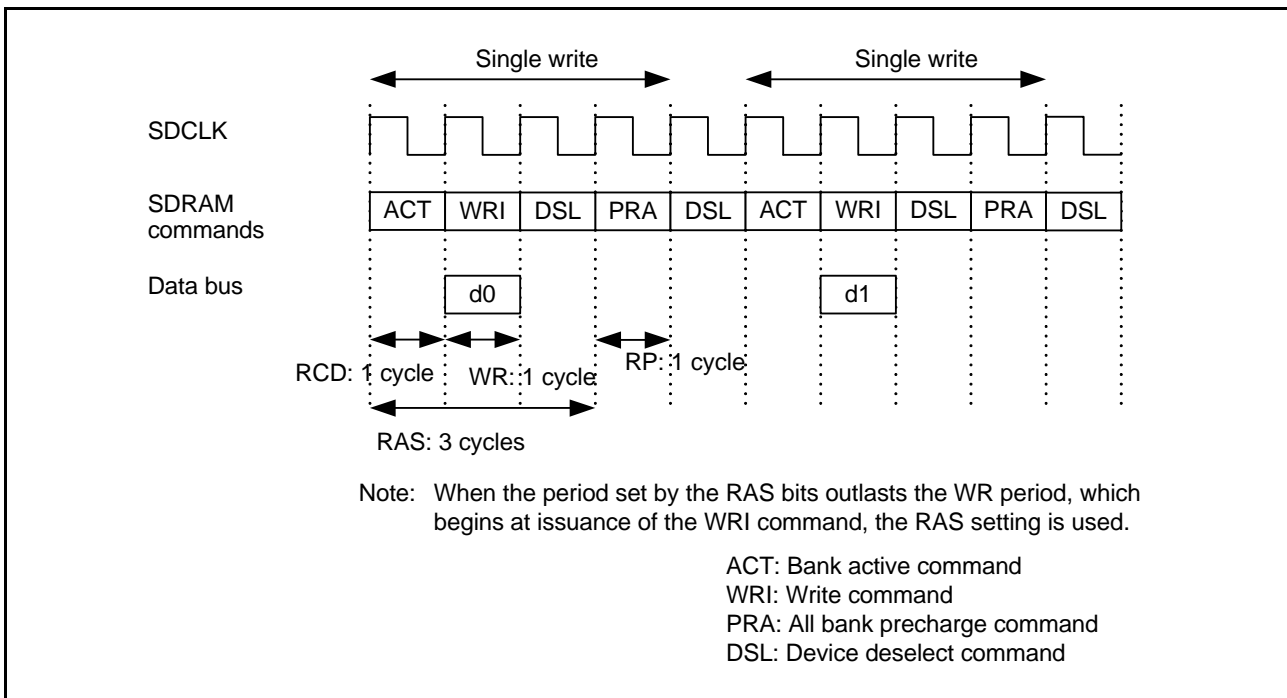


Figure 12.65 Timing Example of Single Write (4)

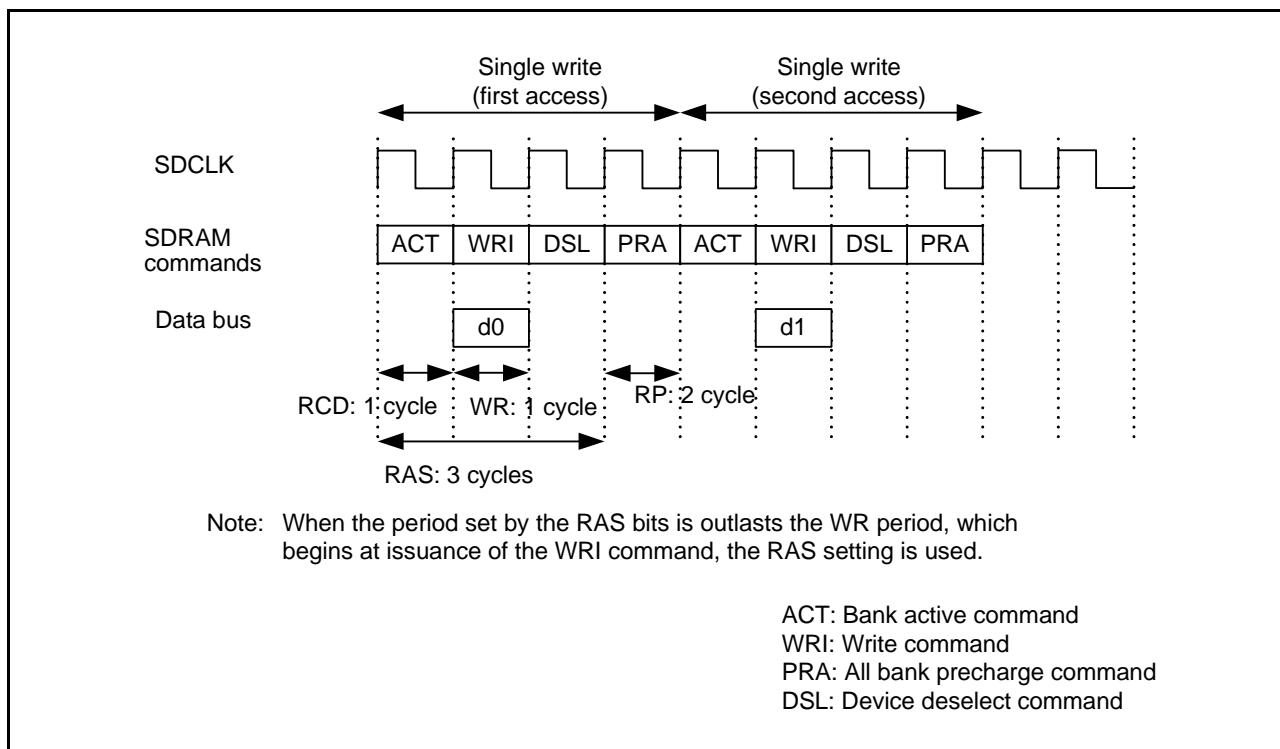


Figure 12.66 Timing Example of Single Write (5) (Two Bus Accesses Occur for One Transfer Request)

(3) Consecutive Read Timing Examples

Figure 12.67 to figure 12.69 show the relationship between the consecutive read timing for four data and the SDTR register settings. Table 12.18 shows the correspondence between the figures and the SDTR register settings.

Table 12.18 Correspondence between Figure 12.63 to Figure 12.65 and STDR Register Settings (Consecutive Read Timing)

Figure No.	RAS[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	CL[2:0] Settings	Number of Cycles
Figure 12.67	010	3	00	1	001	2	010	2
Figure 12.68	000	1	01	2	001	2	010	2
Figure 12.69	000	1	01	2	001	2	011	3

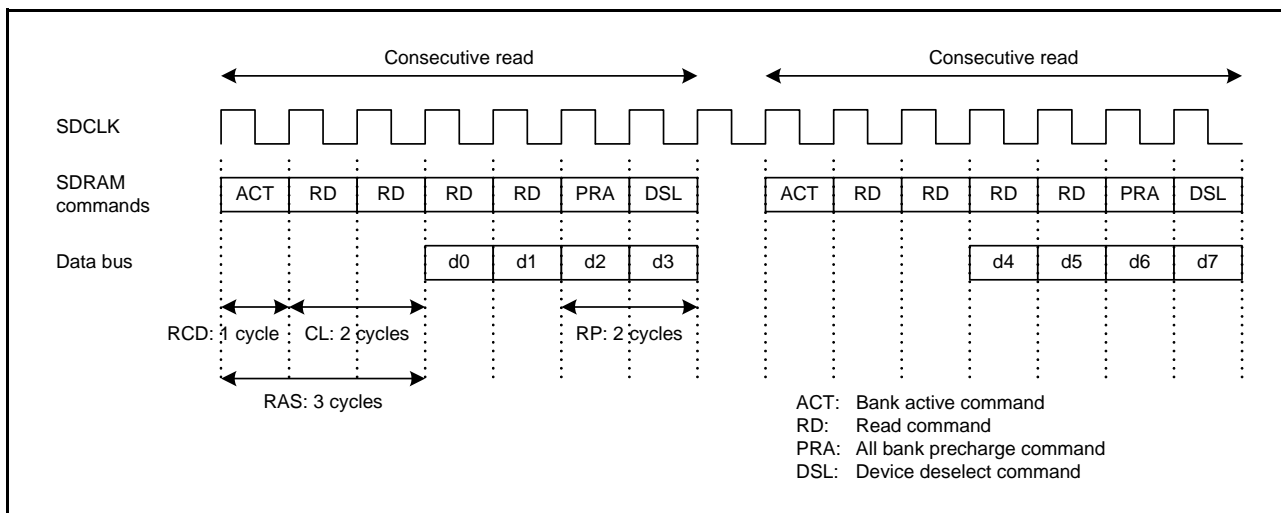


Figure 12.67 Timing Example of Consecutive Read (1)

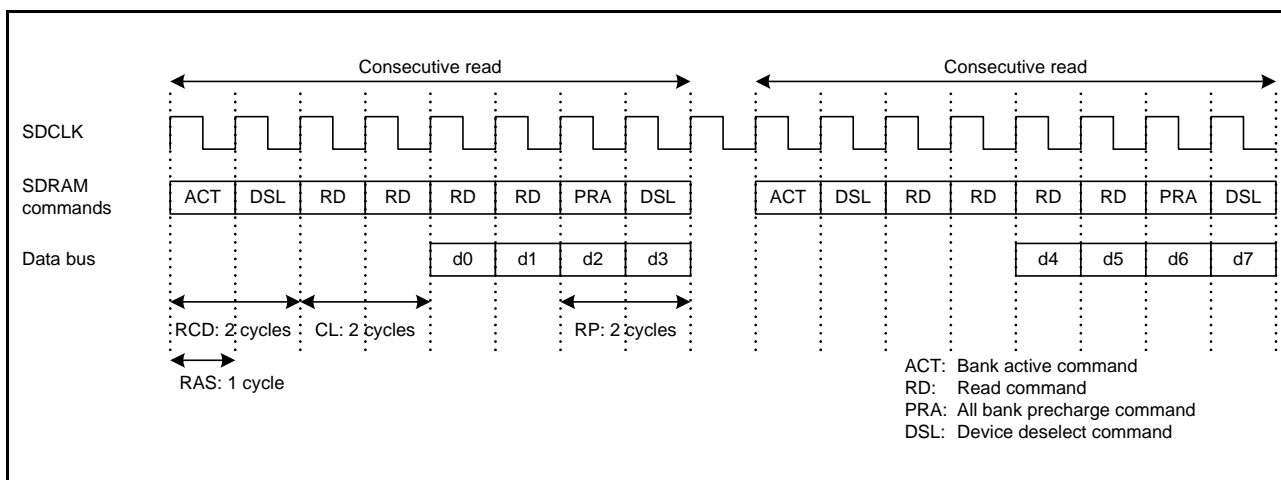


Figure 12.68 Timing Example of Consecutive Read (2)

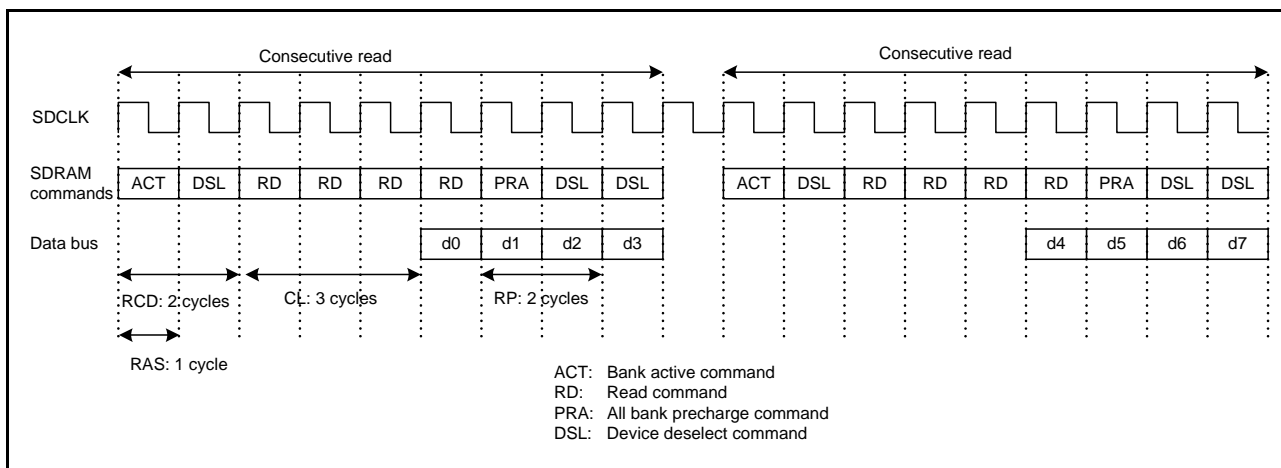


Figure 12.69 Timing Example of Consecutive Read (3)

(4) Consecutive Write Timing Examples

Figure 12.70 to figure 12.72 show the relationship between the consecutive write timing for four data and the SDTR register settings. Table 12.19 shows the correspondence between the figures and the SDTR register settings.

Table 12.19 Correspondence between Figure 12.66 to Figure 12.68 and STDR Register Settings (Consecutive Write Timing)

Figure No.	RAS[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	WR Settings	Number of Cycles
Figure 12.70	010	3	00	1	001	2	0	1
Figure 12.71	000	1	01	2	001	2	0	1
Figure 12.72	000	1	01	2	001	2	1	2

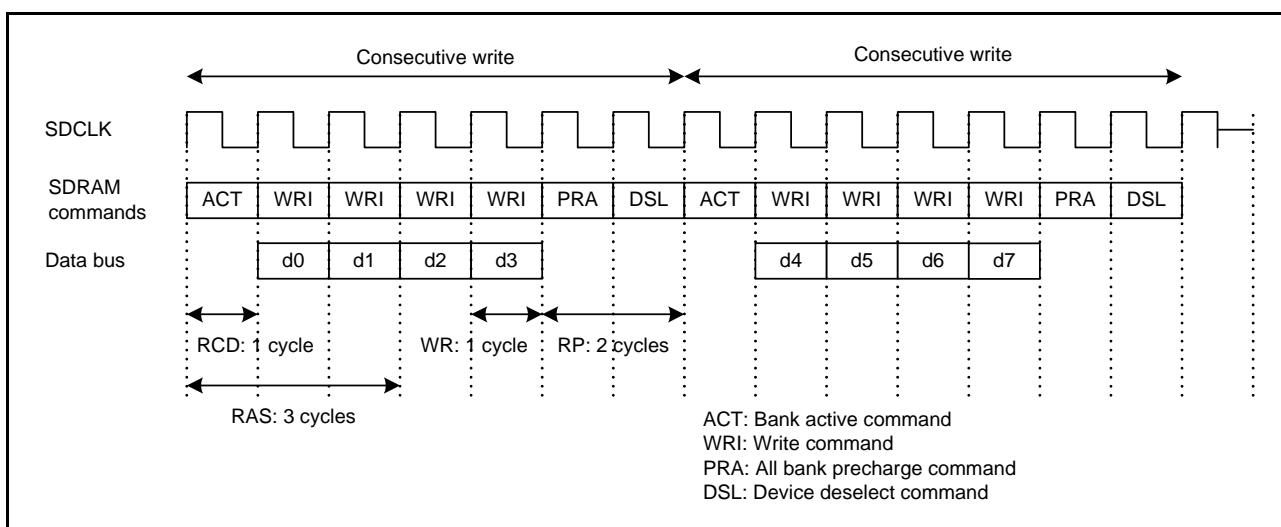


Figure 12.70 Timing Example of Consecutive Write (1)

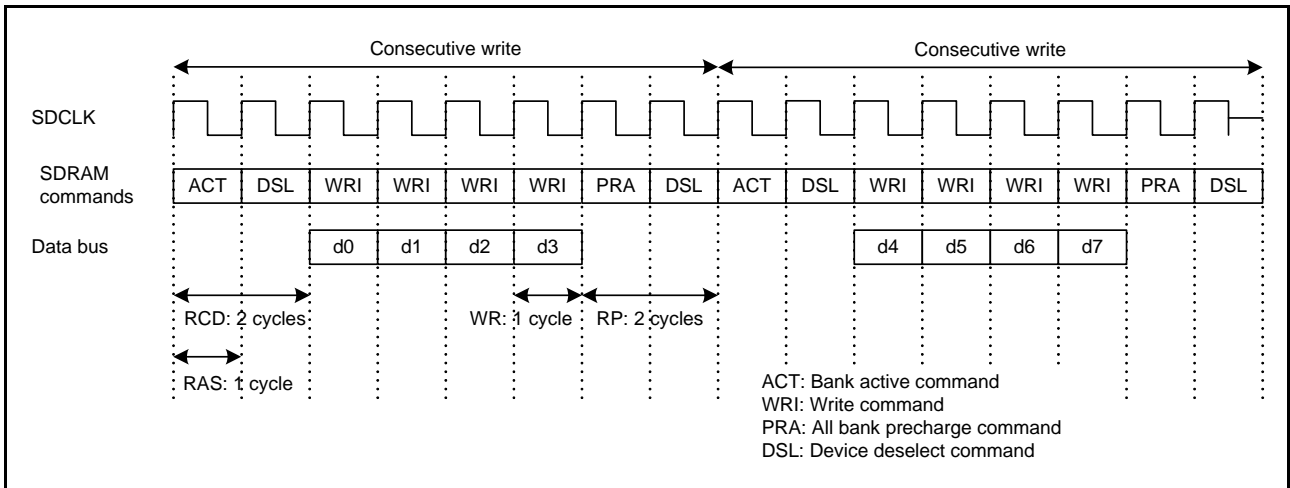


Figure 12.71 Timing Example of Consecutive Write (2)

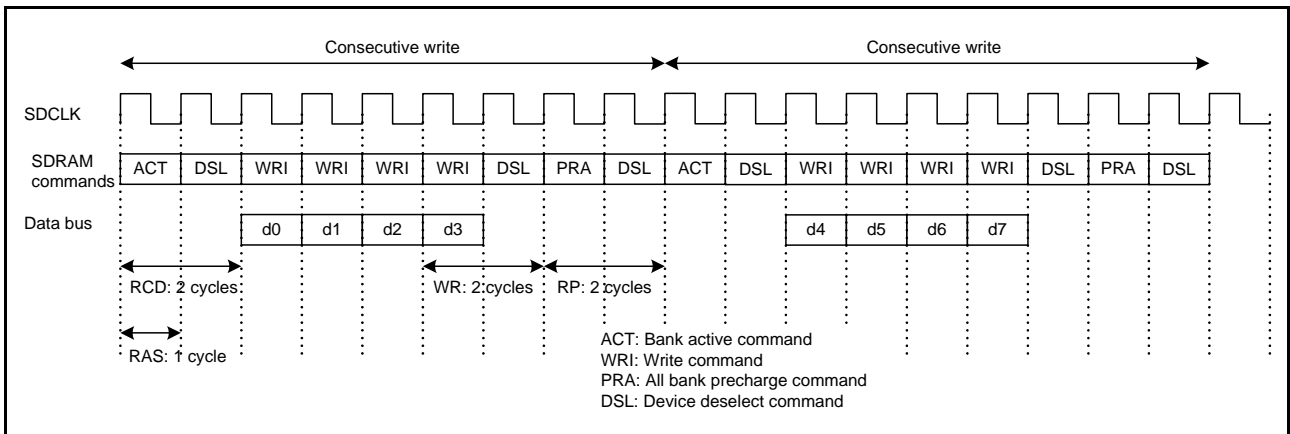


Figure 12.72 Timing Example of Consecutive Write (3)

12.6.13 Address Multiplexing

In the SDRAM space, row and column addresses are multiplexed. The size of the shift in a row address should be specified for address multiplexing by the address multiplex select bits (SDADR.MXC[1:0]) in the SDRAM address register (SDADR). Moreover, in the SDRAM space, the address precharge select command (precharge-sel) is output to the upper bits of column addresses. Table 12.20 shows the relationship between the SDADR.MXC[1:0] settings and the shift amount.

Table 12.20 Address Multiplexing

MXC [1:0]	Shift Amount	Data Bus Width	Address	Address Pins External to the Microcomputer																		
				A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
00	8 bits	8 bits	Row	A26	A25	A24	A23	A22	A21	A20	A19	A18*	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
			Column	A26	A25	A24	A23	A22	A21	A20	A19	P	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		16 bits	Row	A26	A25	A24	A23	A22	A21	A20	A19*	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
			Column	A26	A25	A24	A23	A22	A21	A20	P	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		32 bits	Row	A26	A25	A24	A23	A22	A21	A20*	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
			Column	A26	A25	A24	A23	A22	A21	P	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
01	9 bits	8 bits	Row	-	A26	A25	A24	A23	A22	A21	A20	A19*	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
			Column	-	A26	A25	A24	A23	A22	A21	A20	P	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		16 bits	Row	-	A26	A25	A24	A23	A22	A21	A20*	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
			Column	-	A26	A25	A24	A23	A22	A21	P	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		32 bits	Row	-	A26	A25	A24	A23	A22	A21*	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
			Column	-	A26	A25	A24	A23	A22	P	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
10	10 bits	8 bits	Row	-	-	A26	A25	A24	A23	A22	A21	A20*	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	-	-	A26	A25	A24	A23	A22	A21	P	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		16 bits	Row	-	-	A26	A25	A24	A23	A22	A21*	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	-	-	A26	A25	A24	A23	A22	P	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		32 bits	Row	-	-	A26	A25	A24	A23	A22*	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	-	-	A26	A25	A24	A23	P	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
11	11 bits	8 bits	Row	-	-	-	A26	A25	A24	A23	A22	A21*	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	-	-	-	A26	A25	A24	A23	A10	P	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		16 bits	Row	-	-	-	A26	A25	A24	A23	A22*	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	-	-	-	A26	A25	A24	A11	P	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		32 bits	Row	-	-	-	A26	A25	A24	A23*	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	-	-	-	A26	A25	A12	P	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Notes P: Precharge setting command (Precharge-sel) is output.
 *: When the PALL command is issued, Precharge-sel = 1 (High) is output. When the Active command is issued, the corresponding address is output.

12.6.14 Examples for Connecting with SDRAMs

12.6.14.1 32-Bit Bus Space

Figure 12.73 shows an example for connecting to two 512-Mbit SDRAMs with 13-bit row address, 10-bit column address and 16-bit bus.

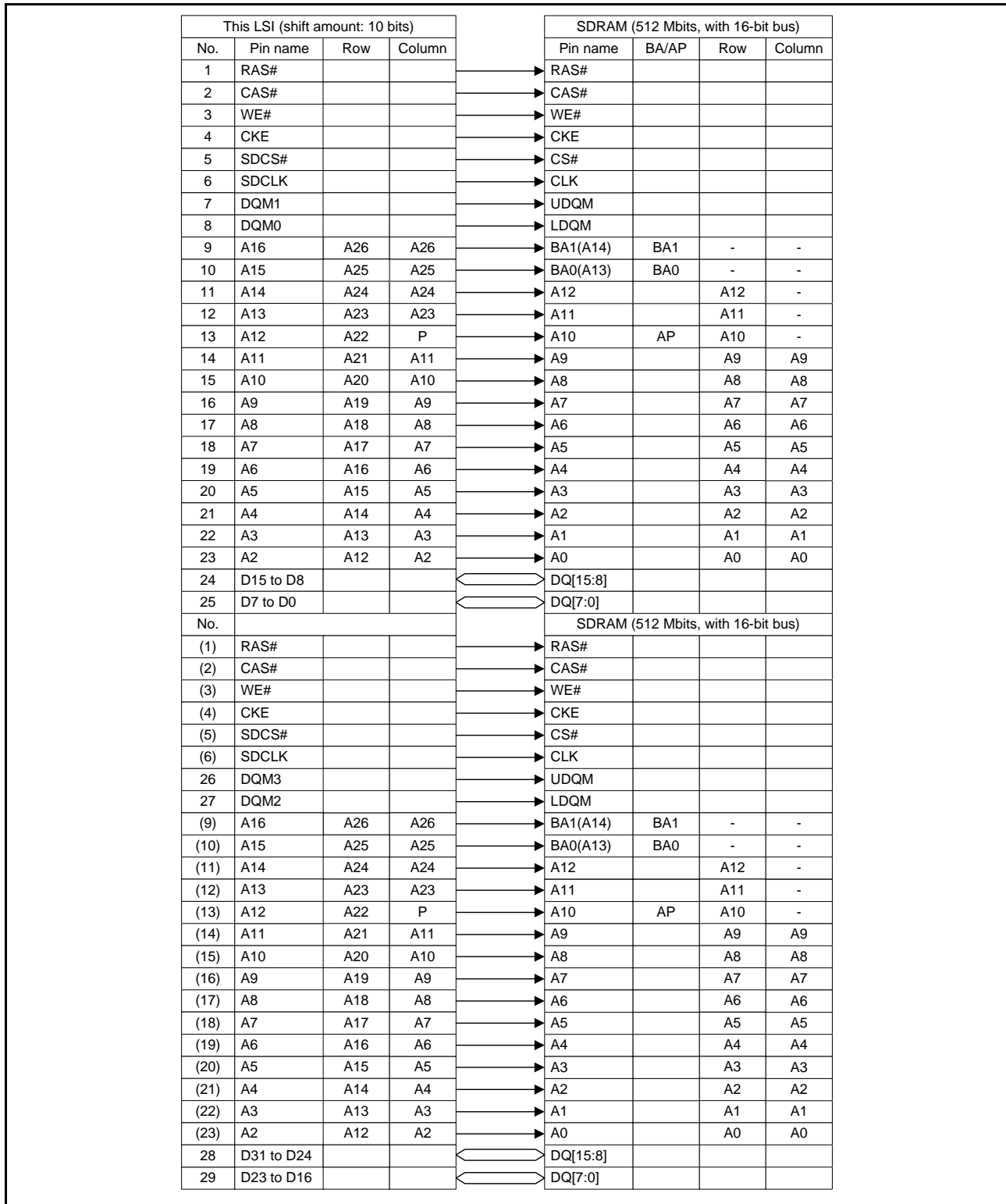


Figure 12.73 SDRAM Connection Example (512-Mbit × 2, with 16-Bit Bus)

Figure 12.74 shows an example for connecting to a 256-Mbit SDRAM with 12-bit row address, 9-bit column address and 32-bit bus.

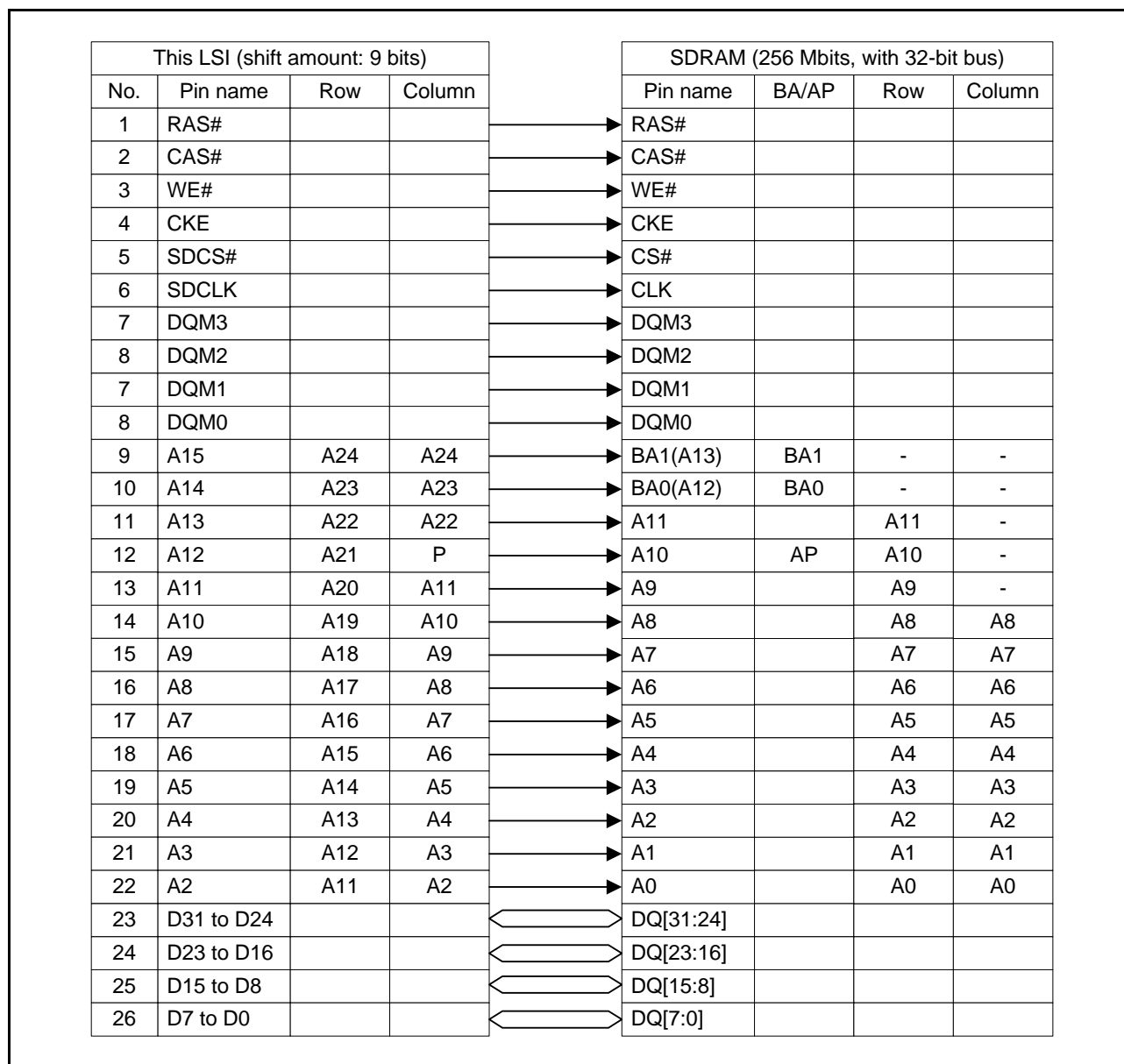


Figure 12.74 SDRAM Connection Example (256-Mbit × 1, with 32-Bit Bus)

Figure 12.75 shows an example for connecting to two 128-Mbit SDRAMs with 12-bit row address, 9-bit column address and 16-bit bus.

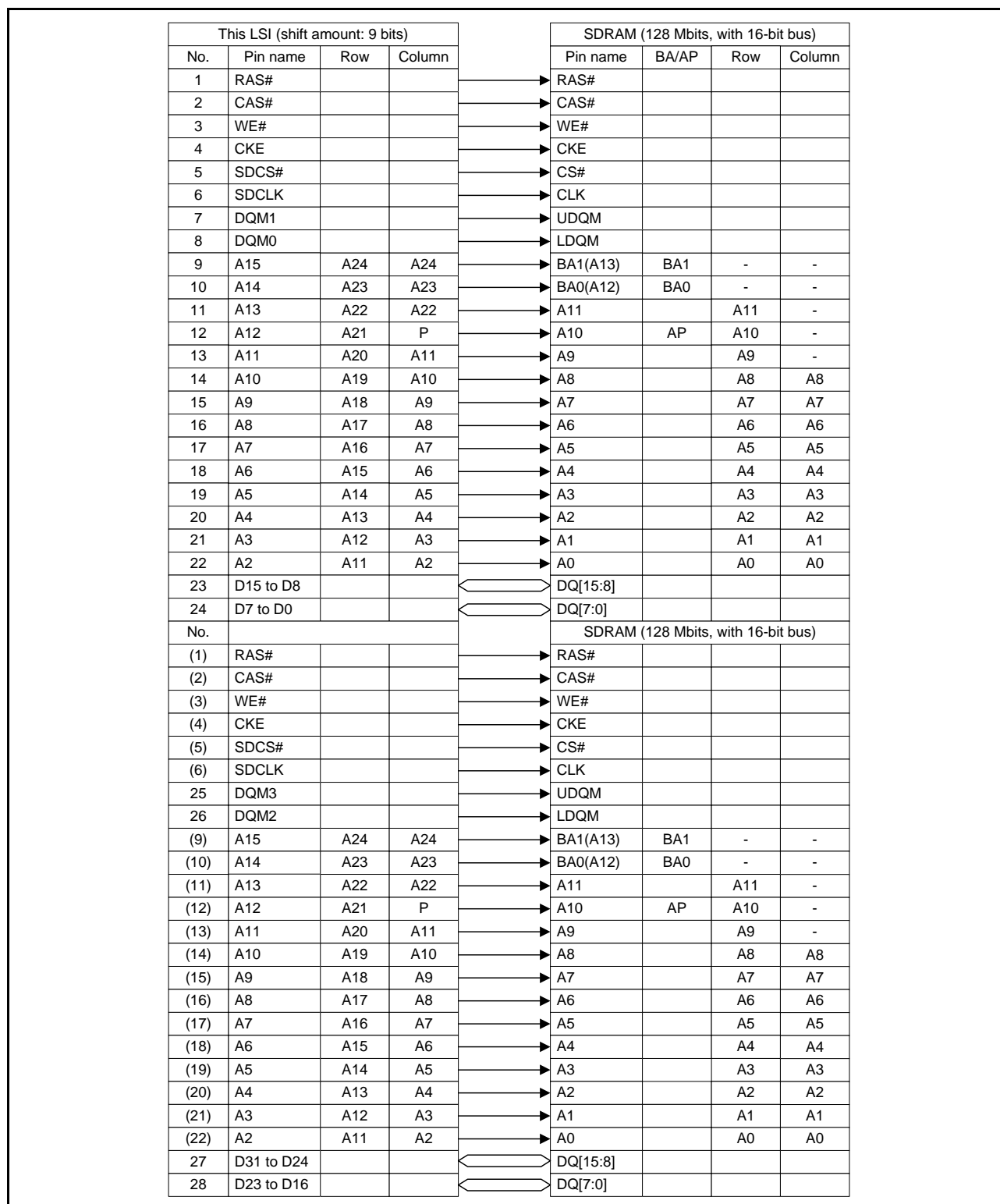


Figure 12.75 SDRAM Connection Example (128-Mbits x 2, with 16-Bit Bus)

12.6.14.2 16-Bit Bus Space

Figure 12.76 shows an example for connecting to two 512-Mbit SDRAMs with 13-bit row address, 11-bit column address and 8-bit bus.

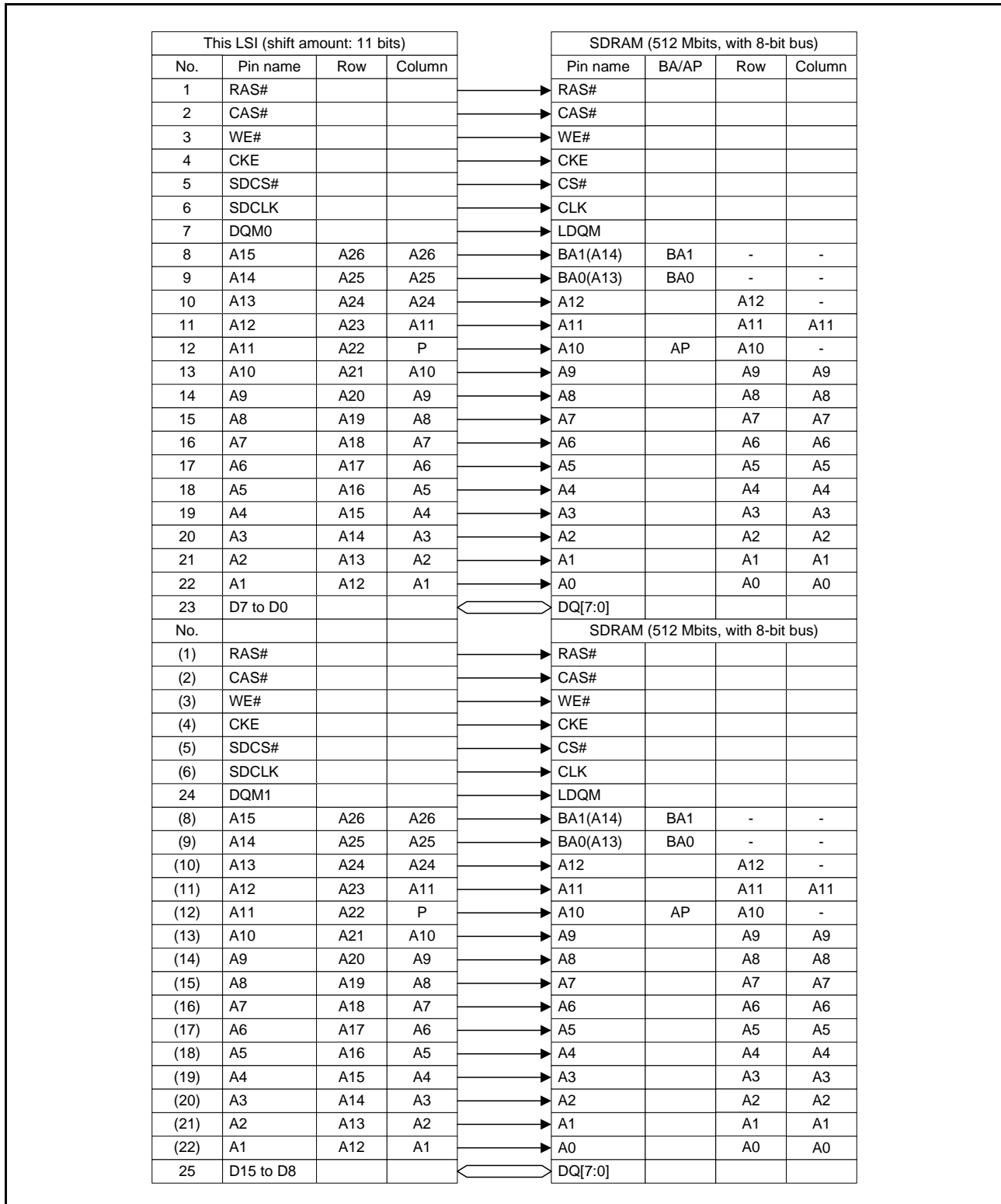


Figure 12.76 SDRAM Connection Example (512-Mbit x 2, with 8-Bit Bus)

Figure 12.77 shows an example for connecting to a 512-Mbit SDRAM with 13-bit row address, 10-bit column address and 16-bit bus.

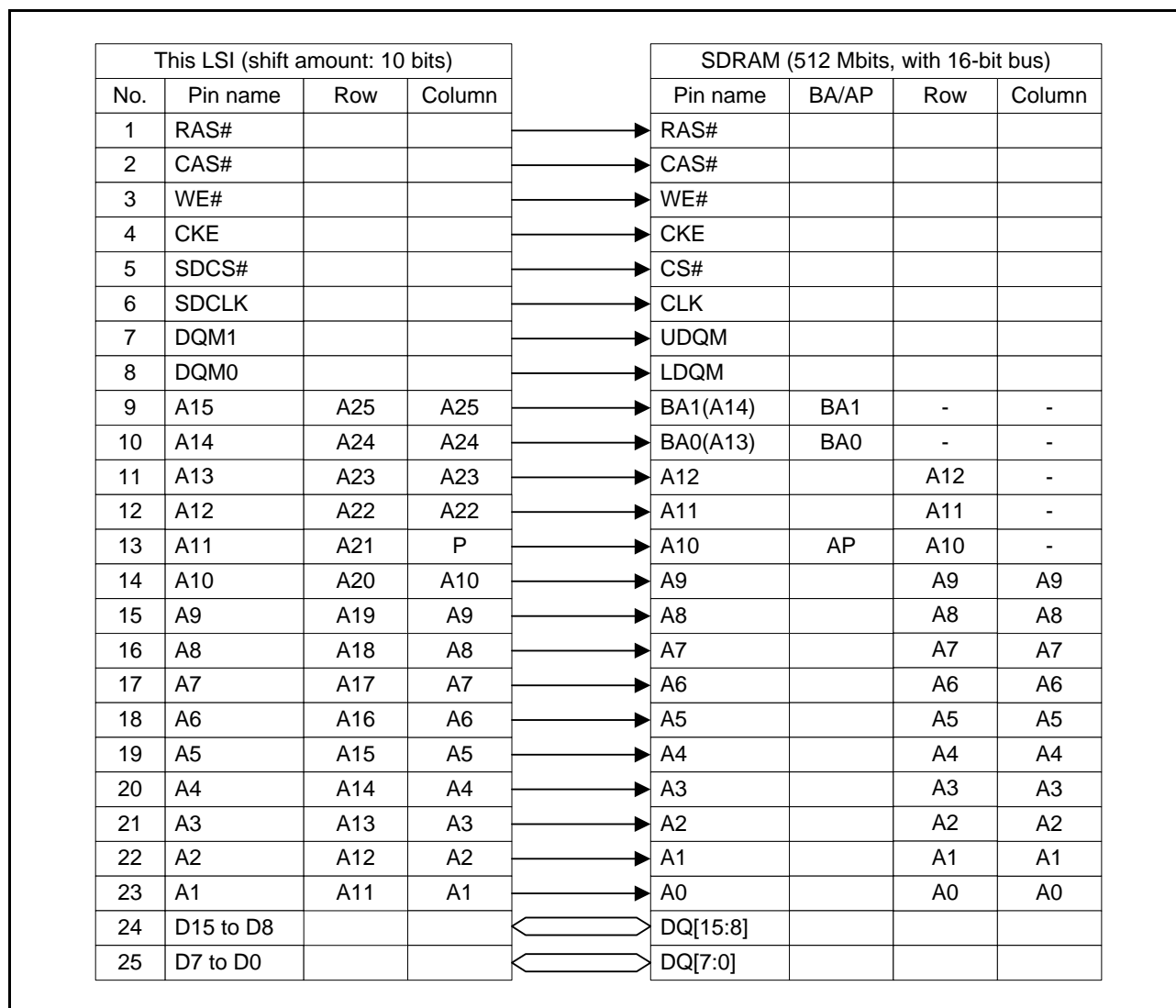


Figure 12.77 SDRAM Connection Example (512-Mbit x 1, with 16-Bit Bus)

Figure 12.78 shows an example for connecting to a 256-Mbit SDRAM with 13-bit row address, 9-bit column address and 16-bit bus.

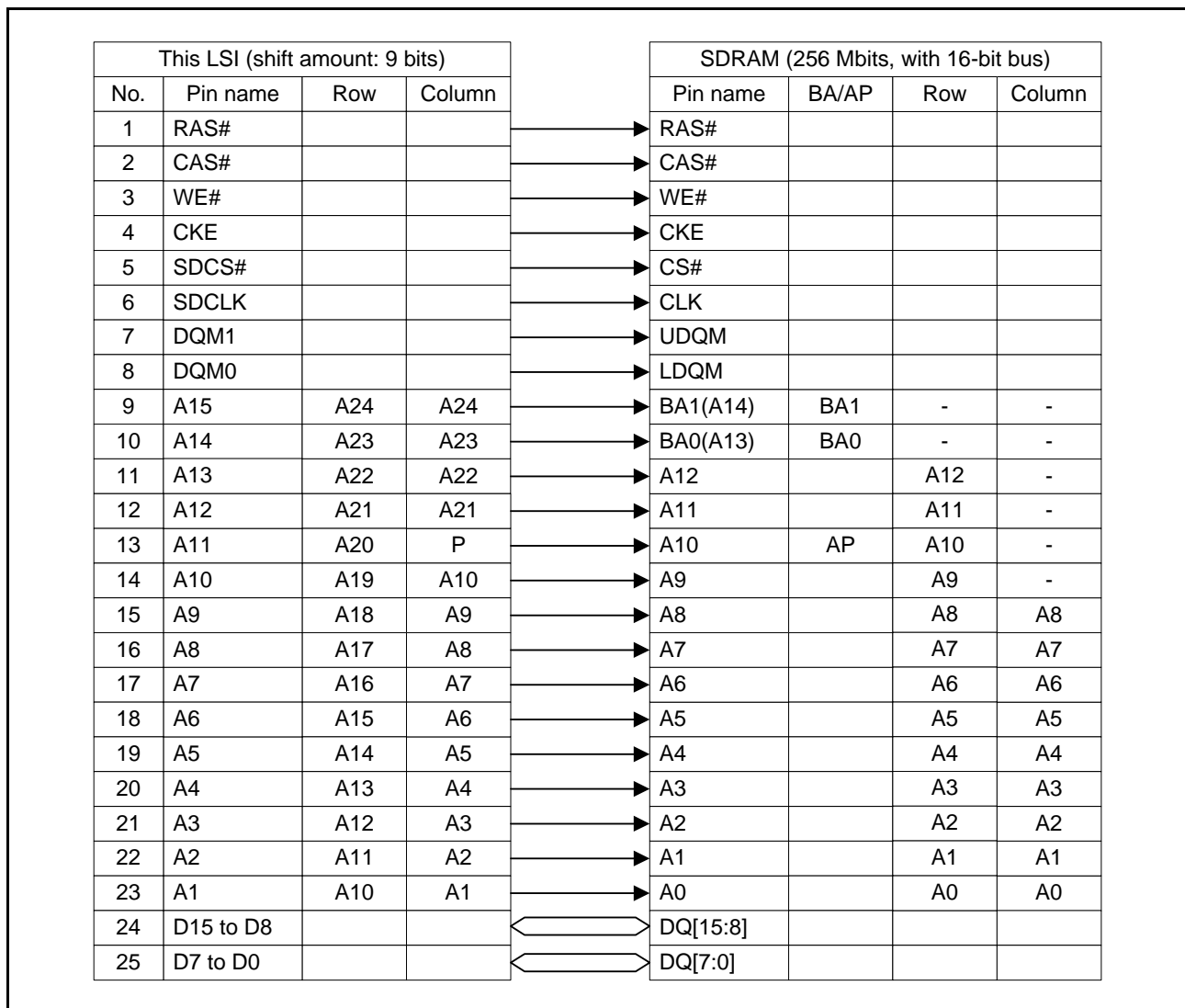


Figure 12.78 SDRAM Connection Example (256-Mbit x 1, with 16-Bit Bus)

12.6.15 Restrictions

(1) Prohibition of Access that Spans Areas of External Address Space

Single access that spans two areas of the external address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.

Single access that spans two areas of the address space is also prohibited in the EXDMAC block transfer in single address mode or cluster transfer, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single access in the EXDMAC block transfer in single address mode or cluster transfer.

(2) Low Power Consumption State

In all-module clock stop mode, software standby mode, and deep software standby mode, auto-refresh operation is not available since the clock supply to SDRAMC is stopped. To retain the data in the SDRAM when the SDRAM is externally connected, use the self-refresh function. For the procedure for transition to and recovery from self-refresh mode, see section 12.6.7, Self-Refresh.

(3) Consecutive-Access Mode

For block transfer or cluster transfer by the EXDMAC in single-address mode, the setting $CL = 1$ is prohibited, and operation is not guaranteed if this setting is made.

(4) Setting the SDRAM Timing Register

Set the RAS[2:0] bits in the SDRAM timing register (SDTR) to a value less than or equal to the sum of the row-column latency (SDTR.RCD[1:0]) and column latency (SDTR.CL[2:0]) settings. Operation is not guaranteed if this condition is not satisfied.

(5) Restrictions on RMPA and String-Manipulation Instructions

- Although the external space has a per-area ending-switching facility (only for data), the allocation of data to be handled by RMPA or string-manipulation instructions to an area where the endian differs from that of the chip is prohibited, and operation is not guaranteed if this restriction is not observed. If data to be handled by RMPA or string-manipulation instructions are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.
- The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

(6) Restriction on Instruction Code

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

12.7 Bus Error Monitoring Section

The bus-error monitoring section monitors the individual areas for bus errors, and generates an interrupt when it detects a bus error.

There are two types of bus error: illegal address access and timeout. Illegal address access is the detection of illegal access to an area, and time-out is the detection of a bus-access operation not being completed within 768 cycles.

12.7.1 Types of Bus Error

There are two types of bus error: illegal address access and timeout.

Illegal address access is the detection of illegal access to an area, and time-out is the detection of a bus-access operation not being completed within 768 cycles.

12.7.1.1 Illegal Address Access

When the illegal address access detection enable bit (IGAEN) in the bus error monitoring enable register (BEREN) is set to 1, access of the following types leads to illegal address access errors.

- Access to areas of external space for which operation has been disabled (CSnCNT.EXENB = 0, SDCCR.EXENB = 0)
- With respect to areas other than those described above, access to illegal address ranges

The address ranges where access will lead to illegal address access errors are indicated in table 12.21.

12.7.1.2 Timeout

When the timeout detection enable bit (TOEN) in the bus error monitoring enable register (BEREN) is set to 1, bus access that is not completed within 768 cycles leads to a timeout error.

- CS areas (CS0 to CS7): Bus access is not completed (the WAIT# signal is not negated) within 768 external bus clock (BCLK) cycles from the start of the access.

When a timeout error occurs, the bus access is forcibly terminated, and thus operation cannot be guaranteed. Once a timeout error occurs, accesses from the bus master are rejected for 256 BCLK cycles. If multiple external bus accesses are generated with a single request from the bus master during the transfer, the bus accesses cannot be stopped by a timeout. At this time, timeout errors may occur repeatedly.

Note: In the RX62N/RX621, timeout errors occur only for access to CS areas.

12.7.2 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 IEN register in the ICU can specify whether to generate an interrupt in the case of a bus error.

12.7.3 Conditions Leading to Bus Errors

Table 12.21 lists the types of bus errors for each area in the respective address space.

If an illegal address access error or timeout is detected when no bus error has occurred (bus error status register n (BERSRn; n = 1 or 2) is cleared), the detected error is reflected on the BERSRn. Once a bus error occurs, no subsequent bus errors are reflected on the register unless the register is cleared.

If bus errors are simultaneously caused by two or more bus masters, error information of only one bus master is reflected: CPU, bus arbitration section, and EDMAC. Once a bus error occurs, the status is retained until BERSRn is cleared.

Table 12.21 Types of Bus Errors

Address	Type of Area		Type of Error			
			Illegal Address Access		Timeout	
	On-chip ROM Mode Enabled	On-chip ROM Mode Disabled	On-chip ROM Mode Enabled	On-chip ROM Mode Disabled	On-chip ROM Mode Enabled	On-chip ROM Mode Disabled
0000 0000h to 0007 FFFFh	On-chip RAM*		—	—	—	—
0008 0000h to 0009 0FFFh	Peripheral I/O registers		—	—	—	—
0009 1000h to 0009 FFFFh			✓	—	—	—
000A 0000h to 000A 00FFh			—	—	—	—
000A 0100h to 000A 01FFh			✓	—	—	—
000A 0200h to 000A 02FFh			—	—	—	—
000A 0300h to 000A 03FFh			✓	—	—	—
000A 0400h to 000A 041Fh			—	—	—	—
000A 0420h to 000B FFFFh			✓	—	—	—
000C 0000h to 000C 043Fh			—	—	—	—
000C 0440h to 000D FFFFh			✓	—	—	—
000E 0000h to 000F FFFFh			✓	—	—	—
0010 0000h to 0011 FFFFh	Data flash*	Reserved area	—	✓	—	—
0012 0000h to 007F 7FFFh	Reserved area		✓	—	—	—
007F 8000h to 007F 9FFFh	FCU RAM		—	—	—	—
007F A000h to 007F BFFFh	Reserved area		✓	—	—	—
007F C000h to 007F C4FFh	Peripheral I/O registers		—	—	—	—
007F C500h to 007F FBFFh	Reserved area		✓	—	—	—
007F FC00h to 007F FFFFh	Peripheral I/O registers		—	—	—	—
0080 0000h to 00DF FFFFh	Reserved area		—	—	—	—
00E0 0000h to 00FF FFFFh	On-chip ROM* (dedicated area for writing)		—	—	—	—
0100 0000h to 07FF FFFFh	External address space (CS1 to CS7)		[IA]*1	—	[TO]	—
0800 0000h to 0FFF FFFFh	SDRAM area		[IA]*2	—	—	—
1000 0000h to 7FFF FFFFh	Reserved area		✓	—	—	—
8000 0000h to FFFF FFFFh	On-chip ROM* (dedicated area for reading)	Reserved area	—	✓	—	—
FF00 0000h to FF7F FFFFh		External address space (CS0)	—	[IA]*1	—	[TO]

[Legend] —: A bus error is not produced.

✓: A bus error is produced.

[IA]*1: Access to this area leads to detection of a bus error if operation for this area is disabled (CSnCR.EXENB = 0; n = 0 to 7).

[IA]*2: Access to this area leads to detection of a bus error if operation for this area is disabled (SDCCR.EXENB = 0)

[TO]: Bus access not being completed within 768 cycles leads to detection of a bus error.

Note: * The capacity of the on-chip RAM, data flash, and on-chip ROM differs depending on the product. For details, see section 37, RAM, section 38, ROM (Flash Memory for Code Storage), and section 39, Data Flash Memory (Flash Memory for Data Storage).

13. Memory-Protection Unit (MPU)

13.1 Overview

The RX CPU incorporates a memory-protection unit that checks the addresses of CPU access to the overall address space (0000 0000h to FFFF FFFFh).

Access-control information can be set for up to eight regions, and permission for access to each region is in accord with this information. The default response to the detection of access to a region where permission has not been set is the generation of a memory-protection error.

The supported access-control information for the individual regions consists of permission to read, permission to write, and permission to execute. This access-control information is effective when the processor mode of the CPU is user mode. Memory protection is not applied when the CPU is in supervisor mode.

Table 13.1 lists the specifications of the memory-protection unit, and Figure 13.1 shows a block diagram of the memory-protection unit.

Table 13.1 Specifications of Memory Protection

Specifications	Description
Region to be covered by memory protection and processor mode	0000 0000h to FFFF FFFFh (in user mode) No memory protection in supervisor mode
Number of regions	8
Page size (smallest unit of protection)	16 bytes
Specifying addresses of individual regions	Setting the page numbers where regions start and end
Setting to make memory protection effective or ineffective in individual regions	A V (valid) bit in each region-n end page number register (REPAGEn) makes the settings effective or ineffective for the corresponding region (n = 0 to 7).
Access-control information settings for individual regions	Instruction execution: Permission to execute Operand access: Permission to read, permission to write
Start of memory-protection operations	After the memory-protection unit has been enabled, access monitoring start-ing up with the transition to user mode.
Memory-protection error processing	Generation of access exceptions
Addresses where memory-protection errors are generated	Address in instruction execution: The PC value is preserved on the stack. Address in operand access: The address is stored in the data memory-protection error address register (MPDEA).
Determining the reasons for memory-protection errors	The memory-protection error status register (MPESTS) holds indicators of the reason.
Background region setting	Access-control information can be set for the background region (the whole address space).
Processing where regions overlap	The access-control information for access to an overlap between regions is the logical OR of the attributes for the given regions, and permission is given priority.

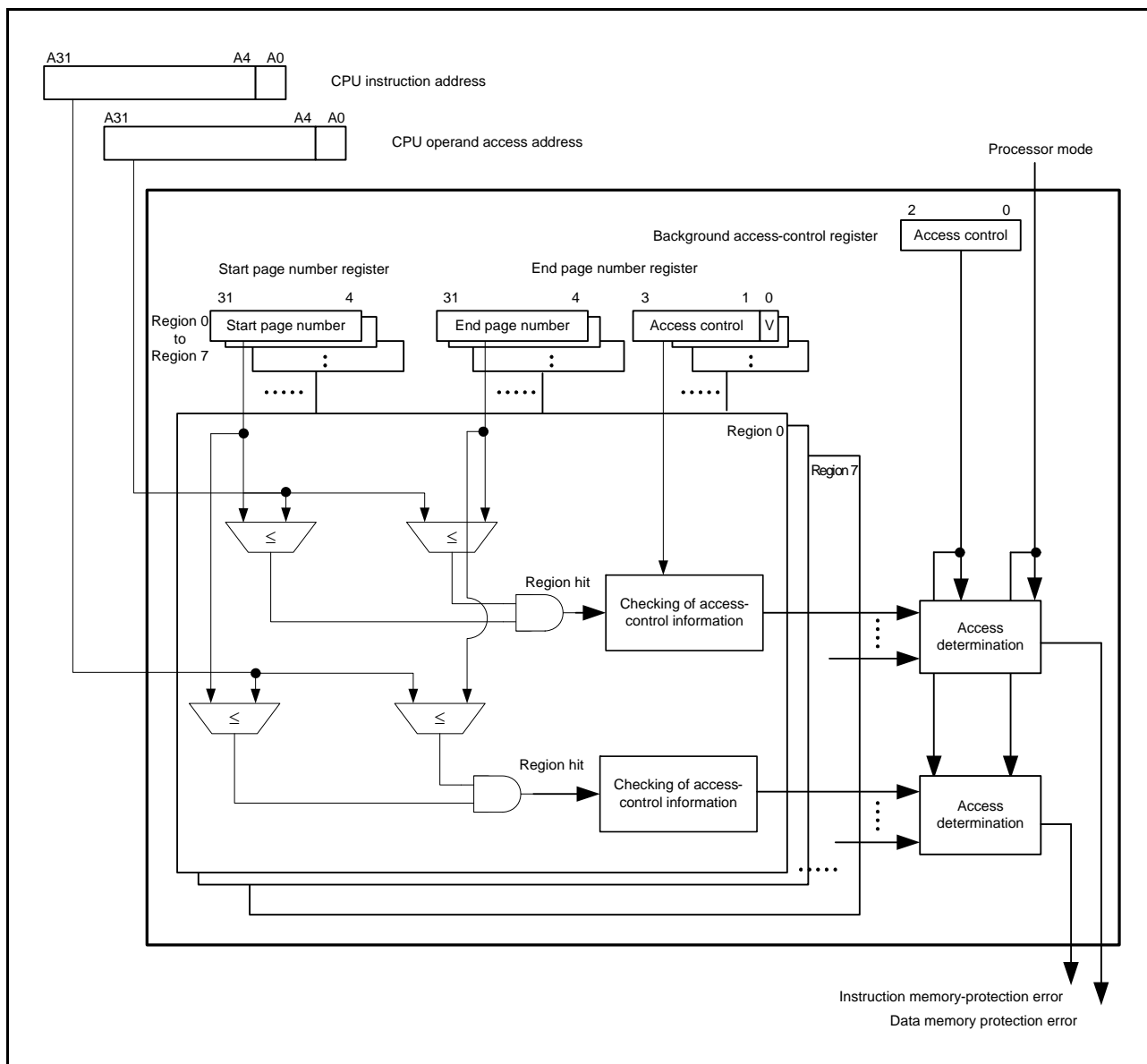


Figure 13.1 Block Diagram of the Memory-Protection Unit

13.1.1 Types of Access Control

There are three types of access control information: permission for instruction execution, permission to read operands, and permission to write operands. Violations of these types of access control are only detected when programs are running in user mode. Violations are not detected when programs are running in supervisor mode.

13.1.2 Regions for Access Control

Up to eight regions for access control are definable. Settings of the range of memory for each access-control region are made in the corresponding region-n start page number register (RSPAGEn) and region-n end page number register (REPAGEn), where n = 0 to 7.

The minimum unit for control of access is the "page", by which the address space is divided into 16-byte units. The 28 higher-order bits ([31:4]) of the address [31:0] bits correspond to the page number.

The REPAGEn register specifies the access-control information for each area and whether the area is enabled or not.

13.1.3 Background Region

"Background region" refers to the whole address space (0000 0000h to FFFF FFFFh). Access-control information for the background region is set in the background-region access-control register (MPBAC). In contrast to the access-control information for the eight individual regions, protection information for the background region is effective as long as memory protection is enabled (the MPEN bit in the MPEN register is 1).

13.1.4 Overlap between Regions

In cases of overlap between multiple regions, the access-control information becomes the logical OR of the access-control bits for the overlapping regions (including the background region), with permission given priority.

13.1.5 Instructions and Data that Span Regions

Operations in response to the detection of memory-protection errors when instructions or data span regions for which different access-control settings have been made are undefined. Ensure that instructions and data do not span regions for which different access-control settings have been made.

13.2 Register Descriptions

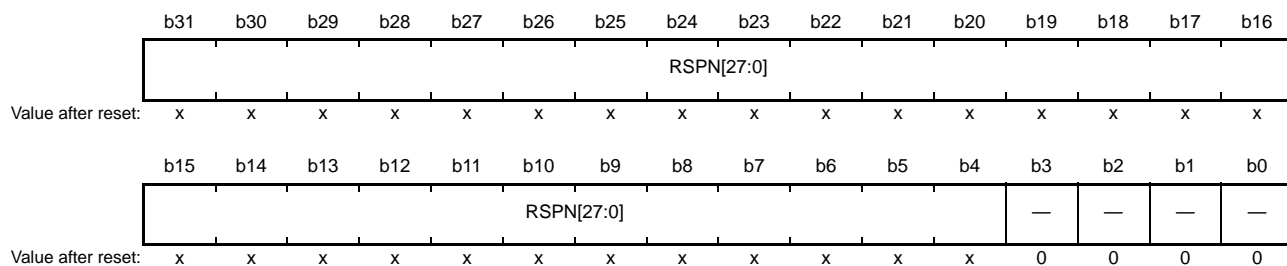
Table 13.2 lists the registers of the interrupt control unit.

Table 13.2 Registers of the Memory-Protection Unit

Register Name	Symbol	Value after Reset	Address	Access Size
Region 0 start page-number register	RSPAGE0	xxxx xxx0h	0008 6400h	32
Region 0 end page-number register	REPAGE0	xxxx xxx0h	0008 6404h	32
Region 1 start page-number register	RSPAGE1	xxxx xxx0h	0008 6408h	32
Region 1 end page-number register	REPAGE1	xxxx xxx0h	0008 640Ch	32
Region 2 start page-number register	RSPAGE2	xxxx xxx0h	0008 6410h	32
Region 2 end page-number register	REPAGE2	xxxx xxx0h	0008 6414h	32
Region 3 start page-number register	RSPAGE3	xxxx xxx0h	0008 6418h	32
Region 3 end page-number register	REPAGE3	xxxx xxx0h	0008 641Ch	32
Region 4 start page-number register	RSPAGE4	xxxx xxx0h	0008 6420h	32
Region 4 end page-number register	REPAGE4	xxxx xxx0h	0008 6424h	32
Region 5 start page-number register	RSPAGE5	xxxx xxx0h	0008 6428h	32
Region 5 end page-number register	REPAGE5	xxxx xxx0h	0008 642Ch	32
Region 6 start page-number register	RSPAGE6	xxxx xxx0h	0008 6430h	32
Region 6 end page-number register	REPAGE6	xxxx xxx0h	0008 6434h	32
Region 7 start page-number register	RSPAGE7	xxxx xxx0h	0008 6438h	32
Region 7 end page-number register	REPAGE7	xxxx xxx0h	0008 643Ch	32
Memory-protection enable register	MPEN	0000 0000h	0008 6500h	32
Background access control register	MPBAC	0000 0000h	0008 6504h	32
Memory-protection error status-clearing register	MPECLR	0000 0000h	0008 6508h	32
Memory-protection error status register	MPESTS	0000 0000h	0008 650Ch	32
Data memory-protection error address register	MPDEA	xxxx xxxh	0008 6514h	32
Region search address register	MPSA	xxxx xxxh	0008 6520h	32
Region search operation register	MPOPS	0000h	0008 6524h	16
Region invalidation operation register	MPOPI	0000h	0008 6526h	16
Instruction-hit region register	MHITI	0000 0000h	0008 6528h	32
Data-hit region register	MHITD	0000 0000h	0008 652Ch	32

13.2.1 Region-n Start Page Number Register (RSPAGEn) (n = 0 to 7)

Addresses: RSPAGE0 0008 6400h, RSPAGE1 0008 6408h, RSPAGE2 0008 6410h, RSPAGE3 0008 6418h
 RSPAGE4 0008 6420h, RSPAGE5 0008 6428h, RSPAGE6 0008 6430h, RSPAGE7 0008 6438h



x: Undefined

Bit	Symbol	Bit Name	Function	R/W
b3 to b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b31 to b4	RSPN[27:0]	Region-Start Page Number	Page number where the region starts, for use in region determination	R/W

RSPN[27:0] Bits (Region-Start Page Number)

These bits specify the page number where the region starts.

13.2.2 Region-n End Page Number Register (REPAGEn) (n = 0 to 7)

Addresses: REPAGE0 0008 6404h, REPAGE1 0008 640Ch, REPAGE2 0008 6414h, REPAGE3 0008 641Ch
 REPAGE4 0008 6424h, REPAGE5 0008 642Ch, REPAGE6 0008 6434h, REPAGE7 0008 643Ch



x: Undefined

Bit	Symbol	Bit Name	Function	R/W
b0	V	Valid Bit	0: Region setting invalid 1: Region setting valid	R/W
b3 to b1	UAC[2:0]	Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	REPN[27:0]	Region-End Page Number	Page number where the region ends, for use in region determination	R/W

V Bit (Valid Bit)

This bit enables or disables the settings for the corresponding region.
 This bit is cleared to 0 when the region invalidation operation register (MPOPI) invalidates all access-controlled areas.

UAC[2:0] Bits (Access Control Bits in User Mode)

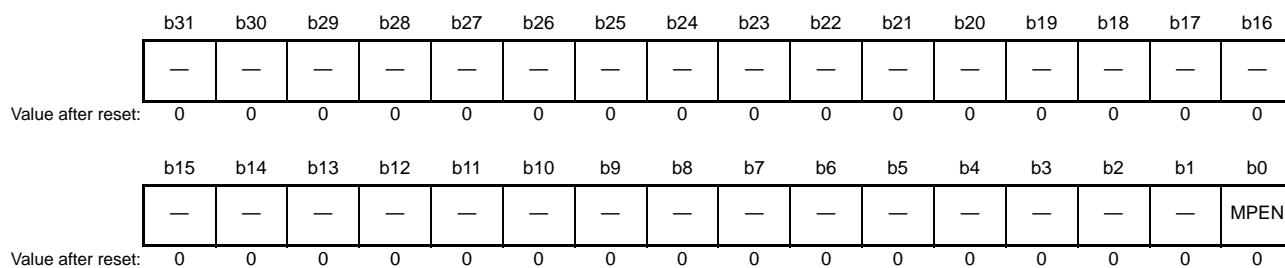
These bits specify the access control in user mode.

REPN[27:0] Bits (Region-End Page Number)

These bits specify the page number where the region ends.
 Specify a value that is greater than or equal to the page number where the corresponding region starts. The page specified by the region-end page number is part of the target region for memory protection.

13.2.3 Memory-Protection Enable Register (MPEN)

Address: 0008 6500h



Bit	Symbol	Bit Name	Function	R/W
b0	MPEN	Memory-Protection Enable	1: The memory protection is enabled. 0: The memory protection is disabled.	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should always be 0	R/W

MPEN Bit (Memory-Protection Enable)

This bit enables or disables the memory protection.

After 1 has been written to this bit, address checking for memory protection by the CPU starts on the execution of a branch instruction (RTE and RTFI) that shifts operation to the user mode.

13.2.4 Background Access Control Register (MPBAC)

Address: 0008 6504h



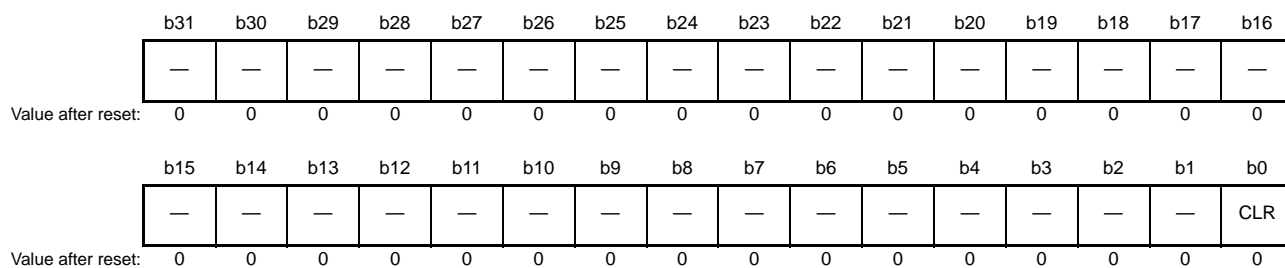
Bit	Symbol	Bit Name	Function	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b3 to b1	UBAC[2:0]	Background Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	—	Reserved	The read value is 0. The write value should always be 0.	R/W

UBAC[2:0] Bits (Background Access Control Bits in User Mode)

These bits specify the background access control in user mode.

13.2.5 Memory-Protection Error Status-Clearing Register (MPECLR)

Address: 0008 6508h



Bit	Symbol	Bit Name	Function	R/W
b0	CLR	Error Status-Clearing	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: The DRW, DA, and IA bits in the MPESTS are cleared to 0.	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W

CLR Bit (Error Status-Clearing)

This bit clears the data read/write bit (DRW), the data memory-protection error generated bit (DA), and the instruction memory-protection error generated bit (IA) in the memory-protection error status register (MPESTS) to 0.

13.2.6 Memory-Protection Error Status Register (MPESTS)

Address: 0008 650Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DRW	DA	IA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	IA	Instruction Memory-Protection Error Generated Bit	0: No instruction memory-protection error was generated. 1: Instruction memory-protection error was generated.	R
b1	DA	Data Memory-Protection Error Generated Bit	0: No data memory-protection error was generated. 1: Data memory-protection error was generated.	R
b2	DRW	Data Read/Write Bit	0: Data were read. 1: Data were written.	R
b31 to b3	—	Reserved	The read value is 0. The write value should always be 0.	R/W

IA Bit (Instruction Memory-Protection Error Generated Bit)

This bit indicates the state of memory-protection error generation by instruction execution.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

DA Bit (Data Memory-Protection Error Generated Bit)

This bit indicates the state of memory-protection error generation by operand access.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

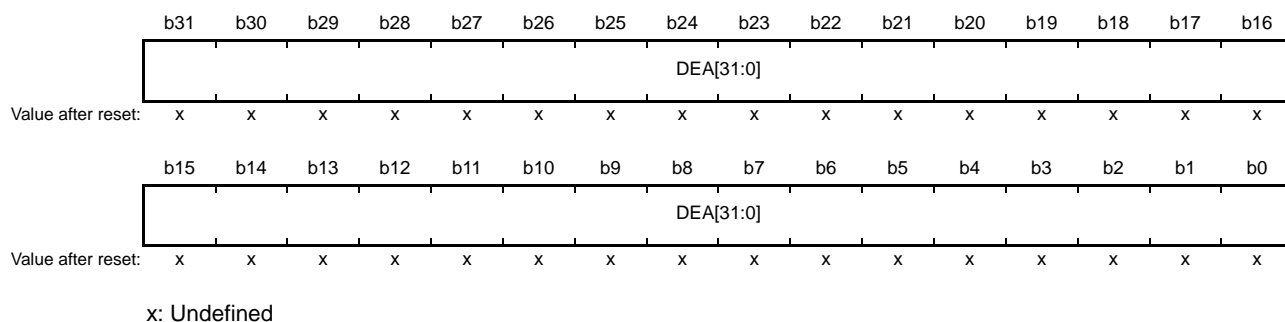
DRW Bit (Data Read/Write Bit)

For a memory-protection error produced by operand access, this bit indicates the read/write attribute of the access operation. This bit is only valid when the data memory-protection error generated bit (DA) is 1.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

13.2.7 Data Memory-Protection Error Address Register (MPDEA)

Address: 0008 6514h



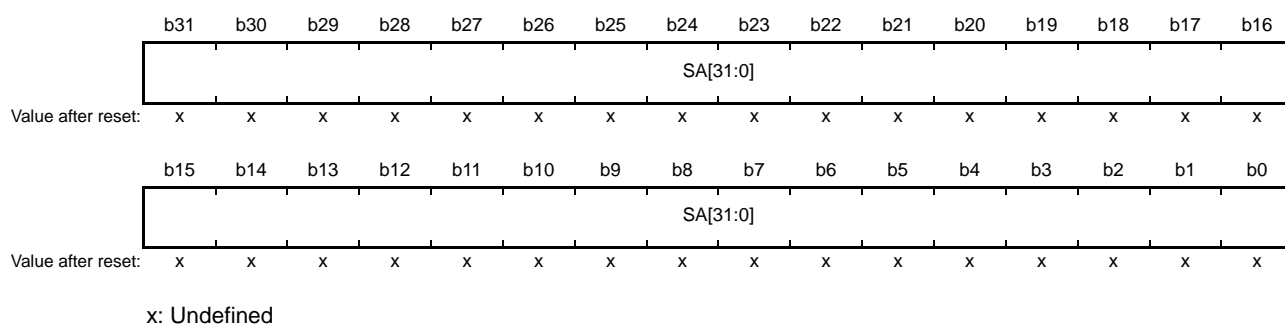
Bit	Symbol	Bit Name	Function	R/W
b31 to b0	DEA[31:0]	Data Memory-Protection Error Address	Data memory-protection error address	R

DEA[31:0] Bits (Data Memory-Protection Error Address)

These bits retain the address for which operand access generated a memory-protection error.

13.2.8 Region Search Address Register (MPSA)

Address: 0008 6520h



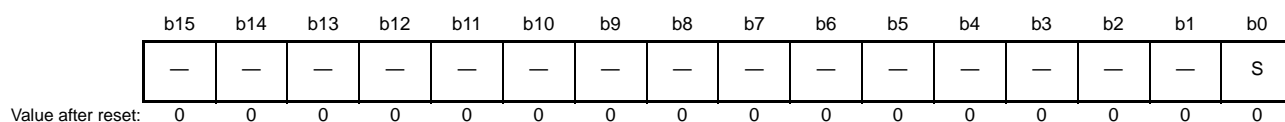
Bit	Symbol	Bit Name	Function	R/W
b31 to b0	SA[31:0]	Region Search Address	Address for region searching	R/W

SA[31:0] Bits (Region Search Address)

These bits specify the address for use in comparison with region-start addresses in the region-n start page number registers (RSPAGEn) and region-end addresses in the region-n end page number registers (REPAGEn).

13.2.9 Region Search Operation Register (MPOPS)

Address: 0008 6524h



Bit	Symbol	Bit Name	Function	R/W
b0	S	Region Search Operation	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: A region-search operation proceeds.	R/W
b15 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W

S Bit (Region Search Operation)

Setting this bit to 1 makes the memory-protection unit perform a region-search operation. The address specified in the region search address register (MPSA) is compared with the address information for individual regions to search for a hitting region.

The result of searching is stored in the data-hit region bits (HITD[7:0]) of the data-hit region register (MHITD). Moreover, the logical OR of the respective access control bits for hitting regions is stored in the data-hit region access control bits (UHACD[2:0]) in user mode.

13.2.10 Region Invalidation Operation Register (MPOPI)

Address: 0008 6526h



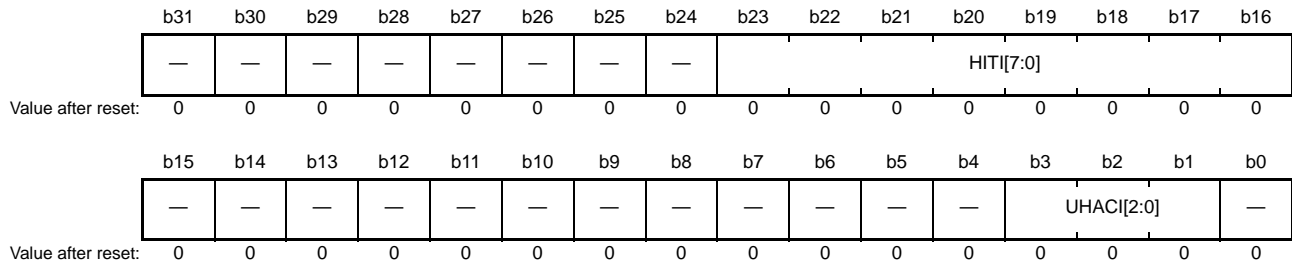
Bit	Symbol	Bit Name	Function	R/W
b0	INV	Region Invalidate Start	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: All access-controlled areas are invalidated.	R/W
b15 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W

INV Bit (Region Invalidate Start)

Setting this bit to 1 clears the valid (V) bits in all of the region-n end page number registers (REPAGEn) to 0. After a V bit is cleared to 0, all settings other than background access-control settings are invalid.

13.2.11 Instruction-Hit Region Register (MHITI)

Address: 0008 6528h



Bit	Symbol	Bit Name	Function	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b3 to b1	UHACI[2:0]	Instruction-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Read permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution is permitted.	R
b15 to b4	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b23 to b16	HITI[7:0]	Instruction-Hit Region	When the instruction memory-protection error generated (IA) bit in the MPESTS = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to an instruction memory-protection error. Other than above b23 0: Instruction memory-protection error was not generated in region 7. 1: Instruction memory-protection error was generated in region 7. b22 0: Instruction memory-protection error was not generated in region 6. 1: Instruction memory-protection error was generated in region 6. b21 0: Instruction memory-protection error was not generated in region 5. 1: Instruction memory-protection error was generated in region 5. b20 0: Instruction memory-protection error was not generated in region 4. 1: Instruction memory-protection error was generated in region 4. b19 0: Instruction memory-protection error was not generated in region 3. 1: Instruction memory-protection error was generated in region 3. b18 0: Instruction memory-protection error was not generated in region 2. 1: Instruction memory-protection error was generated in region 2. b17 0: Instruction memory-protection error was not generated in region 1. 1: Instruction memory-protection error was generated in region 1. b16 0: Instruction memory-protection error was not generated in region 0. 1: Instruction memory-protection error was generated in region 0.	R
b31 to b24	—	Reserved	The read value is 0. The write value should always be 0.	R/W

UHACI[2:0] Bits (Instruction-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) for the region where the instruction memory-protection error was generated.

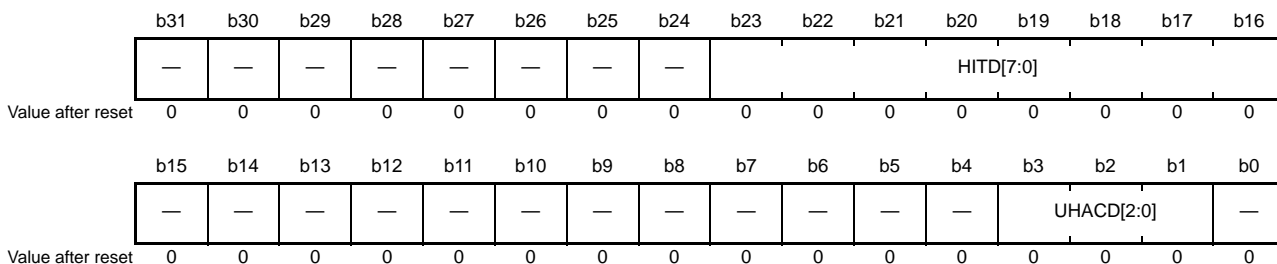
If the error was generated in an overlap between regions, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

HITI[7:0] Bits (Instruction-Hit Region)

These bits indicate the region where an instruction memory-protection error was generated. These bits are set to 0000 0000b in response to the generation of an instruction memory-protection error in the background region.

13.2.12 Data-Hit Region Register(MHITD)

Address: 0008 652Ch



Bit	Symbol	Bit Name	Function	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b3 to b1	UHACD[2:0]	Data-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R
b15 to b4	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b23 to b16	HITD[7:0]	Data-Hit Region	When the data memory-protection error generated (DA) bit = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to a data memory-protection error. Other than above b23 0: Neither a data memory-protection error nor a search hit was generated in region 7. 1: A data memory-protection error or search hit was generated in region 7. b22 0: Neither a data memory-protection error nor a search hit was generated in region 6. 1: A data memory-protection error or search hit was generated in region 6. b21 0: Neither a data memory-protection error nor a search hit was generated in region 5. 1: A data memory-protection error or search hit was generated in region 5. b20 0: Neither a data memory-protection error nor a search hit was generated in region 4. 1: A data memory-protection error or search hit was generated in region 4. b19 0: Neither a data memory-protection error nor a search hit was generated in region 3. 1: A data memory-protection error or search hit was generated in region 3. b18 0: Neither a data memory-protection error nor a search hit was generated in region 2. 1: A data memory-protection error or search hit was generated in region 2. b17 0: Neither a data memory-protection error nor a search hit was generated in region 1. 1: A data memory-protection error or search hit was generated in region 1. b16 0: Neither a data memory-protection error nor a search hit was generated in region 0. 1: A data memory-protection error or search hit was generated in region 0.	R
b31 to b24	—	Reserved	The read value is 0. The write value should always be 0.	R/W

UHACD[2:0] Bits (Data-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) that have been set for the region where a data memory-protection error was generated or the region that produced a hit in region searching.

When an error is generated in an overlap between regions or a hit was generated in region searching, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

HITD[7:0] Bits (Data-Hit Region)

These bits indicate the region where a data memory-protection error was generated or the region that produced a hit in a region search. These bits are set to 0000 0000b for a data memory-protection error generated in the background region.

Note: When access to a register of memory protection unit in user mode generates a data memory-protection error, the value in this register is cleared to 0000 0000h.

13.3 Functions

13.3.1 Memory Protection

Memory protection means monitoring, in accord with the access-control information that has been set for the individual access-control regions and the background region, whether or not access by programs running in user mode violates the access-control settings. The memory-protection unit notifies the CPU of access-control violations (or memory-protection errors) when they are detected, causing the CPU to start access-exception processing.

Memory protection is enabled by setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1.

An instruction memory-protection error is generated on detection of an instruction-execution violation and a data memory-protection error is generated on detection of an operand-access reading or writing violation. Operand access that leads to a data memory-protection error is not actually executed.

13.3.2 Region Search

Region search means enquiry as to which of the eight specified access regions was “hit” and how the access-control information (permission to execute, to read, and to write) is set.

When the region search operation (S) bit in the region-search operation (MPOP) register is set to 1, the address specified in the region search address (MPSA) register is compared with the addresses for the individual regions. After a region search is executed, the data-hit region register (MHITD) indicates the logical OR of the access-control information for the region which was “hit” and for the other regions.

13.3.3 Protection of Registers Related to the Memory-Protection Unit

Registers related to the memory-protection unit are not accessible through means of access other than operand access by the CPU (i.e. by instruction fetching or DMA). Attempted access to registers related to the memory-protection unit in user mode through operand access by the CPU leads to a data memory-protection error regardless of whether or not memory protection is in effect at the given location.

13.3.4 Flow for Determination of Access by the Memory-Protection Function

Figure 13.2 shows the flow of determination in the case of data access and Figure 13.3 shows the flow of determination in the case of instruction access.

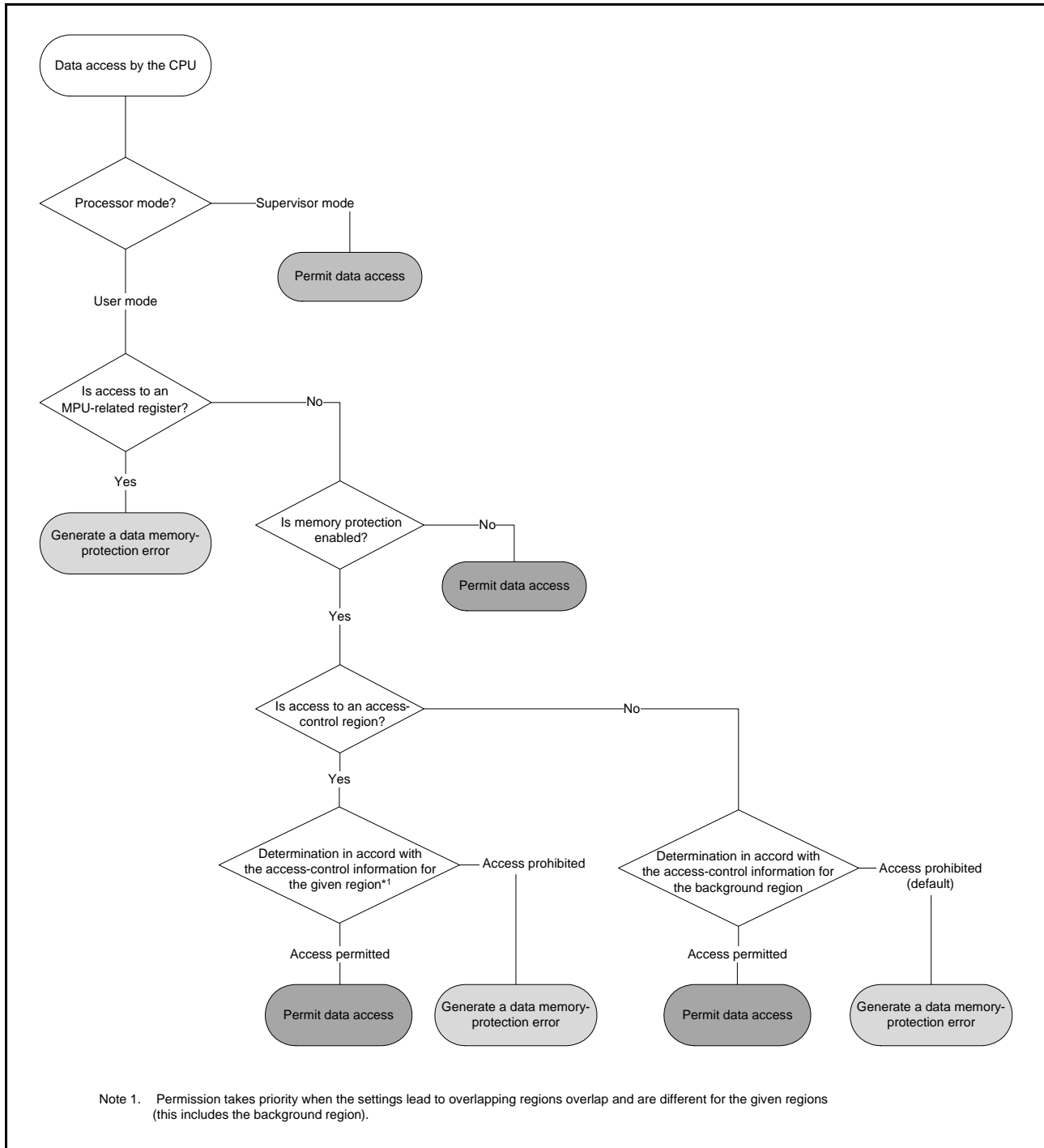


Figure 13.2 Flow of Determination for Data Access

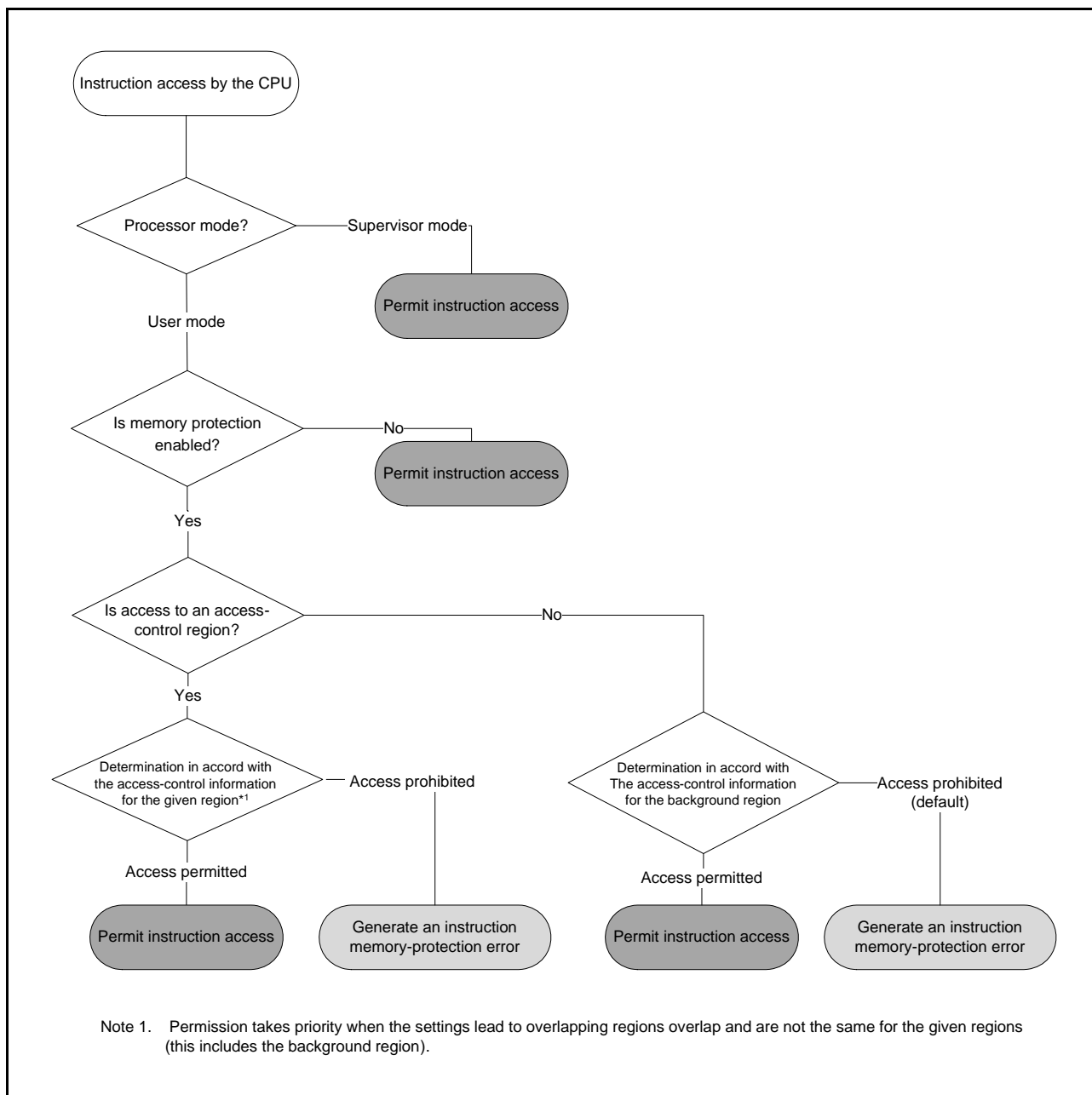


Figure 13.3 Flow of Determination for Instruction Access

13.4 Procedures for Using Memory Protection

13.4.1 Setting Access-Control Information

Access-control information for the various regions is set in supervisor mode.

Settings for up to eight access-control regions are made in the region-n start page number registers (RSPAGEn) and region-n end page number registers (REPAGEn), where n = 0 to 7.

Settings for the background access-control region are made in the background access-control register (MPBAC).

13.4.2 Enabling Memory Protection

Setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1 while operation is in supervisor mode enables memory protection.

13.4.3 Transition to User Mode

After updating the registers related to the memory-protection unit, be sure to read the registers for which writing was performed and check that the settings have been made as the final step before the transition to user mode.

Either of the methods below can be used for the transition from supervisor mode to user mode.

- Set the processor mode setting (PM) bit in the copy of the processor status word (PSW) saved in the stack area to 1 (the setting for user mode) and then execute an RTE instruction.
- Set the PM bit in the backup processor status word (BPSW) to 1 and then execute an RTFI instruction.

Note: Using an MVTC or POPC instruction to write to the PSW.PM bit is invalid. Use an RTE or RTFI instruction to update the value of the PSW.PM bit.

The memory-protection unit starts checking instruction-execution access and operand access by the CPU on the transition to user mode.

13.4.4 Processing in Response to Memory-Protection Errors

The CPU starts access-exception processing on detection of a violation of protection set up by the access-control information (i.e. a memory-protection error). For details on CPU operations in access-exception processing, refer to section 10., Exceptions.

To determine whether an instruction memory-protection error or data memory-protection error has been generated, check the values of the instruction memory-protection error generated (IA) and data memory-protection error generated (DA) bits in the memory-protection error status (MPESTS) register from within the exception-processing routine.

After confirming the type of error, clear the memory-protection error status (MPESTS) register by writing 1 to the status clearing (MPE) bit in the memory-protection error status clearing (MPECLR) register.

(1) When a data memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the address of the operand for which access led to a memory-protection error is stored in the data memory-protection error address register (MPDEA) and the region information for the region where the memory-protection error was generated is stored in the data-hit region register (MHITD).

- Violations of access control in access to valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical “or” of the region access-control information for the location where the error occurred is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

- Violations of access control for the background region, besides access to valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

Referring to this information can pinpoint the sources of errors.

(2) When an instruction memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the region information for the region where the memory-protection error was generated is stored in the instruction-hit region register (MHITI).

- Violations of access control in access to valid regions 0 to 7

The instruction-hit region bit (MHITI.HITI[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical “or” of the region access-control information for the location where the error occurred is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

- Violations of access control for the background region, besides access to valid regions 0 to 7

The instruction-hit region bits (MHITI.HITI[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

Referring to this information can pinpoint the sources of errors.

14. DMA Controller (DMACA)

The RX62N/RX621 Group incorporates a 4-channel direct memory access controller (DMACA).

The DMACA is a module to transfer data without the CPU. When a DMA transfer request is generated, the DMACA transfers data stored at the transfer source address to the transfer destination address.

14.1 Overview

Table 14.1 lists the specifications of the DMACA, and Figure 14.1 shows a block diagram of the DMACA.

Table 14.1 Specifications of DMACA

Item		Description
Number of channels		4 (DMACn (n = 0 to 3))
Transfer space		512 Mbytes (00000000h to 0FFFFFFFh and F0000000h to FFFFFFFFh excluding reserved areas)
Maximum transfer volume		1023K data (Maximum number of transfers in block transfer mode: 1023 data × 1024 blocks)
DMACA request source		<ul style="list-style-type: none"> ▪ Activation source selectable for each channel Software trigger Interrupt requests from peripheral modules or trigger input to external interrupt input pins*
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3 (Channel 0: Highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1023
Transfer mode	Normal transfer mode	<ul style="list-style-type: none"> ▪ One data transfer by one DMA transfer request ▪ Free running mode (setting in which total number of data transfers is not specified) settable
	Repeat transfer mode	<ul style="list-style-type: none"> ▪ One data transfer by one DMA transfer request ▪ Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. ▪ Maximum settable repeat size: 1024
	Block transfer mode	<ul style="list-style-type: none"> ▪ One block data transfer by one DMA request ▪ Maximum settable block size: 1023 data
Selective functions	Extended repeat area function	<ul style="list-style-type: none"> ▪ Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed ▪ Area of two bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Interrupt request	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	Generated when the repeat size of data transfer is completed or the extended repeat area overflows.

Note: For details on DMACA activation sources, see section 14.3.4, Activation Sources.

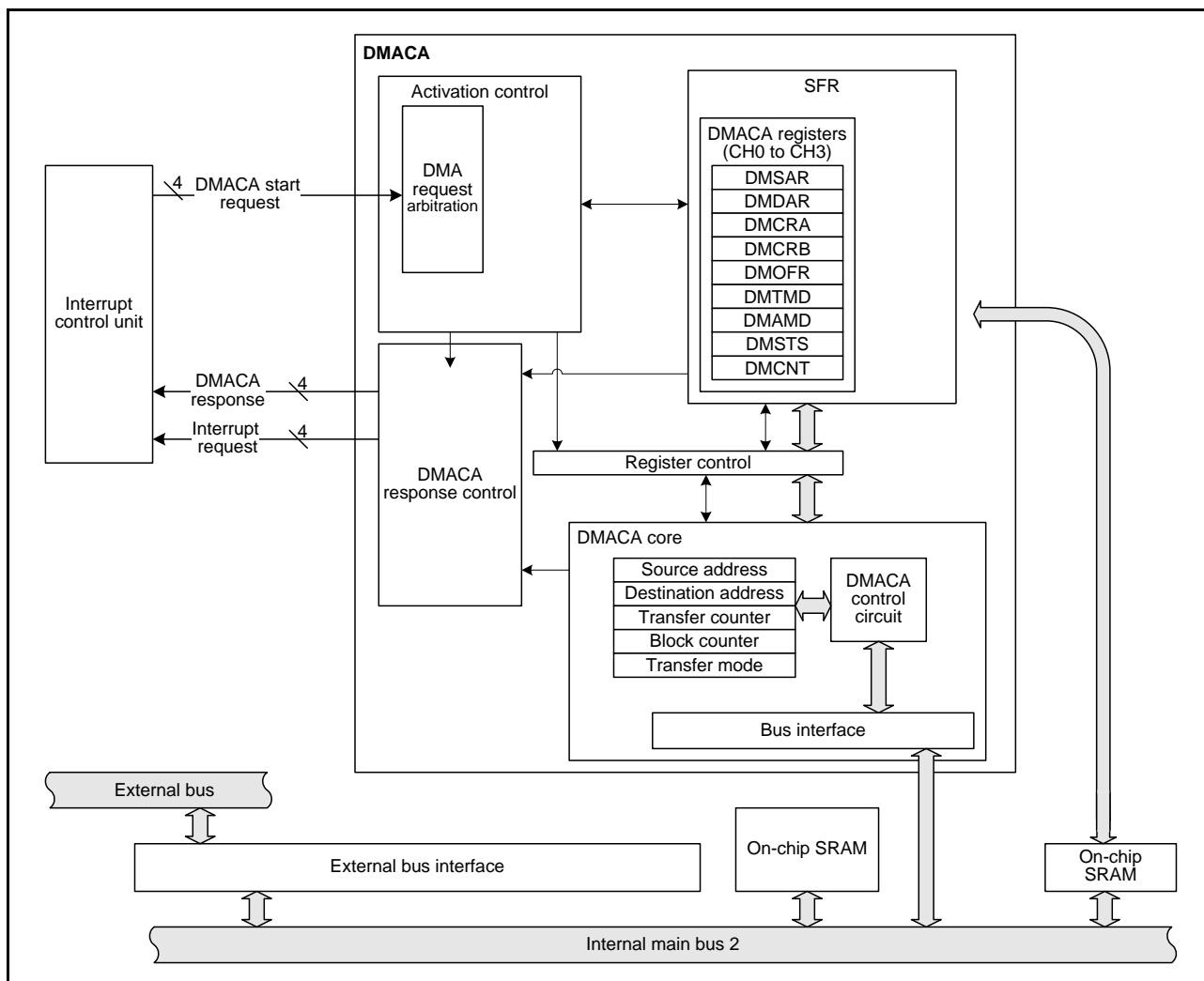


Figure 14.1 Block Diagram of DMACA

14.2 Register Descriptions

Table 14.2 lists the registers of the DMACA. Registers of DMAC0 to DMAC3 have same functions.

Table 14.2 Registers of DMACA (1 / 2)

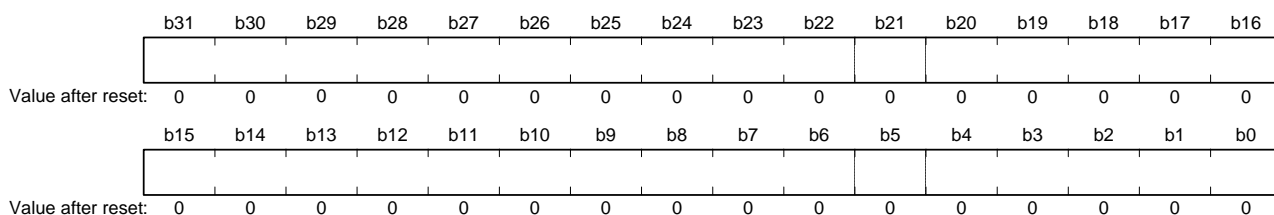
Channel	Register Name	Symbol	Value after Reset	Address	Access Size
DMAC0	DMA source address register	DMSAR	0000 0000h	0008 2000h	32
	DMA destination address register	DMDAR	0000 0000h	0008 2004h	32
	DMA transfer count register	DMCRA	0000 0000h	0008 2008h	32
	DMA block transfer count register	DMCRB	0000h	0008 200Ch	16
	DMA transfer mode register	DMTMD	0000h	0008 2010h	16
	DMA interrupt setting register	DMINT	00h	0008 2013h	8
	DMA address mode register	DMAMD	0000h	0008 2014h	16
	DMA offset register	DMOFR	0000 0000h	0008 2018h	32
	DMA transfer enable register	DMCNT	00h	0008 201Ch	8
	DMA software start register	DMREQ	00h	0008 201Dh	8
	DMA status register	DMSTS	00h	0008 201Eh	8
	DMA activation source flag control register	DMCSL	00h	0008 201Fh	8
DMAC1	DMA source address register	DMSAR	0000 0000h	0008 2040h	32
	DMA destination address register	DMDAR	0000 0000h	0008 2044h	32
	DMA transfer count register	DMCRA	0000 0000h	0008 2048h	32
	DMA block transfer count register	DMCRB	0000h	0008 204Ch	16
	DMA transfer mode register	DMTMD	0000h	0008 2050h	16
	DMA interrupt setting register	DMINT	00h	0008 2053h	8
	DMA address mode register	DMAMD	0000h	0008 2054h	16
	DMA transfer enable register	DMCNT	00h	0008 205Ch	8
	DMA software start register	DMREQ	00h	0008 205Dh	8
	DMA status register	DMSTS	00h	0008 205Eh	8
	DMA activation source flag control register	DMCSL	00h	0008 205Fh	8
	DMAC2	DMA source address register	DMSAR	0000 0000h	0008 2080h
DMA destination address register		DMDAR	0000 0000h	0008 2084h	32
DMA transfer count register		DMCRA	0000 0000h	0008 2088h	32
DMA block transfer count register		DMCRB	0000h	0008 208Ch	16
DMA transfer mode register		DMTMD	0000h	0008 2090h	16
DMA interrupt setting register		DMINT	00h	0008 2093h	8
DMA address mode register		DMAMD	0000h	0008 2094h	16
DMA transfer enable register		DMCNT	00h	0008 209Ch	8
DMA software start register		DMREQ	00h	0008 209Dh	8
DMA status register		DMSTS	00h	0008 209Eh	8
DMA activation source flag control register		DMCSL	00h	0008 209Fh	8

Table 14.2 Registers of DMACA (2 / 2)

Channel	Register Name	Symbol	Value after Reset	Address	Access Size
DMAC3	DMA source address register	DMSAR	0000 0000h	0008 20C0h	32
	DMA destination address register	DMDAR	0000 0000h	0008 20C4h	32
	DMA transfer count register	DMCRA	0000 0000h	0008 20C8h	32
	DMA block transfer count register	DMCRB	0000h	0008 20CCh	16
	DMA transfer mode register	DMTMD	0000h	0008 20D0h	16
	DMA interrupt setting register	DMINT	00h	0008 20D3h	8
	DMA address mode register	DMAMD	0000h	0008 20D4h	16
	DMA transfer enable register	DMCNT	00h	0008 20DCh	8
	DMA software start register	DMREQ	00h	0008 20DDh	8
	DMA status register	DMSTS	00h	0008 20DEh	8
	DMA activation source flag control register	DMCSL	00h	0008 20DFh	8
DMAC	DMA module start register	DMAST	00h	0008 2200h	8

14.2.1 DMA Source Address Register (DMSAR)

Addresses: DMAC0.DMSAR 0008 2000h, DMAC1.DMSAR 0008 2040h
 DMAC2.DMSAR 0008 2080h, DMAC3.DMSAR 0008 20C0h



Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source start address.	00000000h to 0FFFFFFFh (256 Mbytes) F0000000h to FFFFFFFFh (256 Mbytes)	R/W

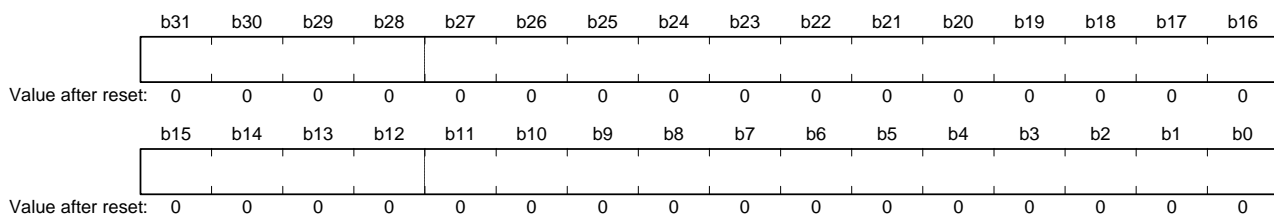
DMSAR specifies the start address of the transfer source.

Set DMSAR while DMACA activation is disabled (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).

Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading DMSAR returns the extended value.

14.2.2 DMA Destination Address Register (DMDAR)

Addresses: DMAC0.DMDAR 0008 2004h, DMAC1.DMDAR 0008 2044h
 DMAC2.DMDAR 0008 2084h, DMAC3.DMDAR 0008 20C4h



Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	00000000h to 0FFFFFFFh (256 Mbytes) F0000000h to FFFFFFFFh (256 Mbytes)	R/W

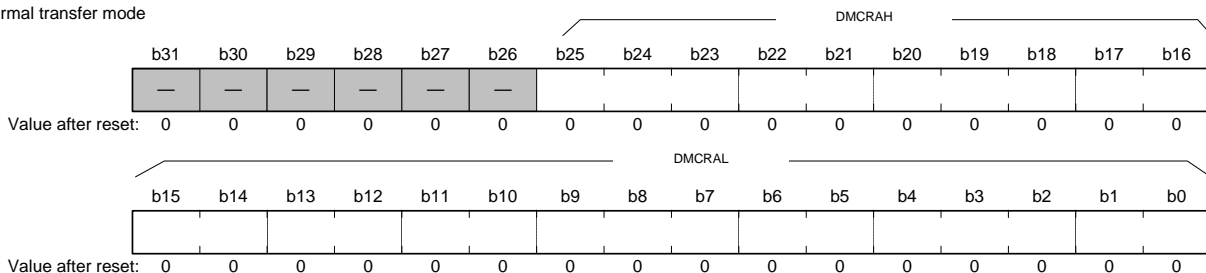
DMDAR specifies the start address of the transfer destination. Set DMDAR while DMACA activation is disabled (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).

Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading DMSAR returns the extended value.

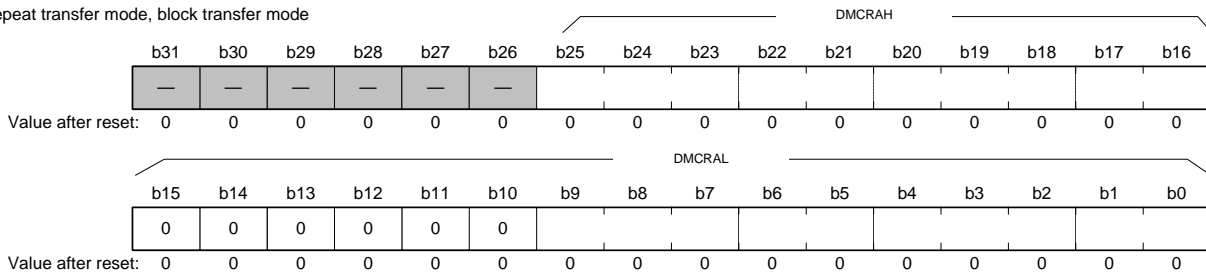
14.2.3 DMA Transfer Count Register (DMCRA)

Addresses: DMAC0.DMCRA 0008 2008h, DMAC1.DMCRA 0008 2048h
 DMAC2.DMCRA 0008 2088h, DMAC3.DMCRA 0008 20C8h

• Normal transfer mode



• Repeat transfer mode, block transfer mode



Symbol	Bit Name	Description	R/W
DMCRAL	Lower bits of transfer count	Specifies the number of transfer operations	R/W
DMCRAH	Upper bits of transfer count		R/W

Note : Set the same value for DMCRAH and DMCRAL in repeat transfer mode and block transfer mode.

DMCRA specifies the number of DMA transfers. The functions of DMCRAL and DMCRAH depend on the transfer mode as described below.

(1) Normal transfer mode (MD[1:0] bits in DMACn.DMTMD = 00b)

DMCRAL functions as a 16-bit transfer counter. The number of transfer operations is one when the setting is 0001h, and 65535 when it is FFFFh.

The value is decremented by one each time data is transferred.

When the setting is 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode).

DMCRAH is not used in normal transfer mode. Write 0000h to DMCRAH.

(2) Repeat transfer mode (MD[1:0] bits in DMACn.DMTMD = 01b)

DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter.

The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In repeat transfer mode, a value in the range of 000h to 3FFh (1 to 1024) can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in DMCRAH is loaded into DMCRAL.

(3) Block transfer mode (MD[1:0] bits in DMACn.DMTMD = 10b)

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.

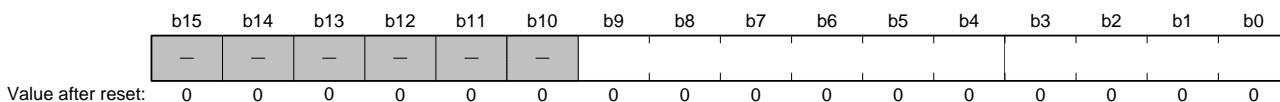
The number of transfers is one when the setting is 001h and 1023 when it is 3FFh. In block transfer mode, a value in the range of 001h to 3FFh can be set for DMCRAH and DMCRAL. Setting a value of 000h is prohibit

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in DMCRAH is loaded into DMCRAL.

14.2.4 DMA Block Transfer Count Register (DMCRB)

Addresses: DMAC0.DMCRB 0008 200Ch, DMAC1.DMCRB 0008 204Ch
 DMAC2.DMCRB 0008 208Ch, DMAC3.DMCRB 0008 20CCh



Bit	Description	Setting Range	R/W
b9 to b0	Specifies the number of block transfer operations or repeat transfer operations.	001h to 3FFh (1 to 1023) 000h (1024)	R/W
b15 to b10	Reserved	These bits are always read as 0. The write value should be 0.	R/W

DMCRB specifies the number of block transfer operations and repeat transfer operations in block and repeat transfer mode, respectively.

The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. The value is decremented by one each time data is transferred.

In normal transfer mode, a value of 3FFh should be set.

14.2.5 DMA Transfer Mode Register (DMTMD)

Addresses: DMAC0.DMTMD 0008 2010h, DMAC1.DMTMD 0008 2050h,
DMAC2.DMTMD 0008 2090h, DMAC3.DMTMD 0008 20D0h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCTG[1:0]	DMA Request Source Select	b1 b0 0 0: Software 0 1: Interrupts* from peripheral modules or external interrupt input pins 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Transfer Data Size Select	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
b11, b10	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Setting prohibited	R/W

Note : DMACA activation source is selected using the DMRSRn registers of the ICU. For details, refer to section 14.3.4, Activation Sources, and section 11, Interrupt Control Unit (ICUa).

DMTMD specifies the DMA transfer mode.

DCTG[1:0] Bits (DMA Request Source Select)

DCTG[1:0] select the DMACA activation source: activation by software or activation by an interrupt request.

SZ[1:0] Bits (Transfer Data Size Select)

SZ[1:0] select the data size of one transfer from among 8, 16, and 32 bits.

DTS[1:0] Bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat or block transfer mode. In normal transfer mode, setting these bits is invalid.

MD[1:0] Bits (Transfer Mode Select)

MD[1:0] select the DMA transfer mode from among normal transfer mode, repeat transfer mode, and block transfer mode.

14.2.6 DMA Interrupt Setting Register (DMINT)

Addresses: DMAC0.DMINT 0008 2013h, DMAC1.DMINT 0008 2053h,
DMAC2.DMINT 0008 2093h, DMAC3.DMINT 0008 20D3h



Bit	Symbol	Bit Name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

DMINT enables/disables the following DMACA interrupt requests.

DARIE Bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the DTE bit in DMCNT is cleared to 0. At the same time, the ESIF bit in DMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer.

When the extended repeat area is not specified for the destination address, this bit is ignored.

SARIE Bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while this bit is set to 1, the DTE bit in DMCNT is cleared to 0. At the same time, the ESIF bit in DMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer.

When the extended repeat area is not specified for the source address, this bit is ignored.

RPTIE Bit (Repeat Size End Interrupt Enable)

When this bit is set to 1 in repeat transfer mode, the DTE bit in DMCNT is cleared to 0 after completion of a 1-repeat size data transfer. At the same time, the ESIF bit in DMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10 (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DTE bit in DMCNT is cleared to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the ESIF bit in DMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10 (= repeat area or block area is not specified.)

ESIE Bit (Transfer Escape End Interrupt Enable)

This bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.

The transfer escape end interrupt is generated when the ESIF bit in DMSTS is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by clearing this bit or the ESIF bit in DMSTS to 0.

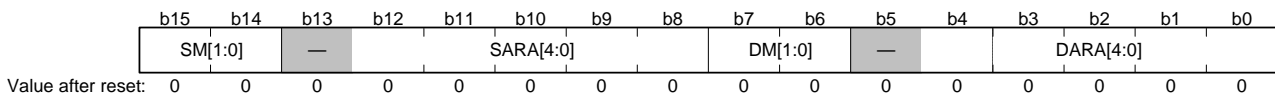
DTIE Bit (Transfer End Interrupt Enable)

This bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DTIF bit in DMSTS is set to 1 with this bit set to 1. The transfer end interrupt is cleared by clearing this bit or the DTIF bit in DMSTS to 0.

14.2.7 DMA Address Mode Register (DMAMD)

Addresses: DMAC0.DMAMD 0008 2014h, DMAC1.DMAMD 0008 2054h,
DMAC2.DMAMD 0008 2094h, DMAC3.DMAMD 0008 20D4h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area	Specifies the extended repeat area on the destination address. For details on the settings, see Table 14.3.	R/W
b5	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition * 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area	Specifies the extended repeat area on the source address. For details on the settings, see Table 14.3.	R/W
b13	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W
b15, b14	SM[1:0]	Source Address Update Mode	b15 b14 0 0: Source address is fixed. 0 1: Offset addition* 1 0: Source address is incremented. 1 1: Source address is decremented.	R/W

Note : Offset addition can be specified only for DMAC0.

DMAMD specifies the DMACA address mode.

DARA[4:0] Bits (Destination Address Extended Repeat Area)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2^1 (2 bytes) and 2^{17} (128 Mbytes).

When the lower address overflows the extended repeat area by address increment, the top address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the bottom address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, or when DMACn.DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write "00000b" in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DARIE bit in DMINT set to 1. Table 14.3 shows the settings and the corresponding extended repeat areas.

DM[1:0] Bits (Destination Address Update Mode)

These bits select the mode of updating the destination address.

When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address.

Offset addition can be specified only for DMAC0.

SARA[4:0] Bits (Source Address Extended Repeat Area)

These bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2^1 (2 bytes) and 2^{17} (128 Mbytes).

When the lower address overflows the extended repeat area by address increment, the top address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the bottom address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, or when DMACn.DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write "00000b" in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the SARIE bit in DMINT set to 1. Table 14.3 shows the settings and the corresponding extended repeat areas.

SM Bit (Source Address Update Mode)

These bits select the mode of updating the source address.

When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address.

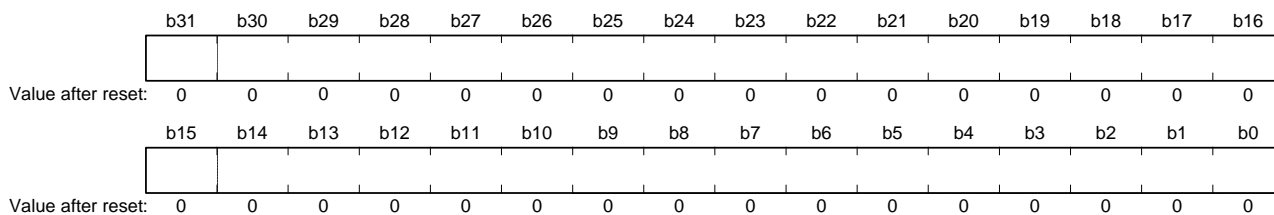
Offset addition can be specified only for DMAC0.

Table 14.3 SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas

SARA4 to SARA0 or DARA4 to DARA0	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	(Setting prohibited)

14.2.8 DMA Offset Register (DMOFR)

Address: DMAC0.DMOFR 0008 2018h



Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	00000000h to 00FFFFFFh (0 bytes to (16M – 1) bytes) FF000000h to FFFFFFFFh (–16M bytes to –1 byte)	R/W

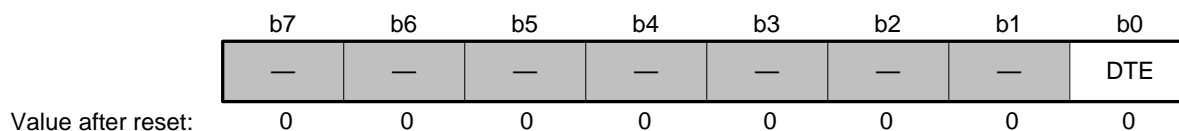
DMOFR specifies the address offset.

Write to this register while the DMACA operation is stopped or DMA transfer is disabled (not during data transfer).

Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

14.2.9 DMA Transfer Enable Register (DMCNT)

Addresses: DMAC0.DMCNT 0008 201Ch, DMAC1.DMCNT 0008 205Ch
DMAC2.DMCNT 0008 209Ch, DMAC3.DMCNT 0008 20DCh



Bit	Symbol	Bit Name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

DMCNT enables or disables DMA transfer for the corresponding channel.

DTE Bit (DMA Transfer Enable)

When the DMST bit in DMAST is set to 1 (DMACA activation is enabled) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

[Setting condition]

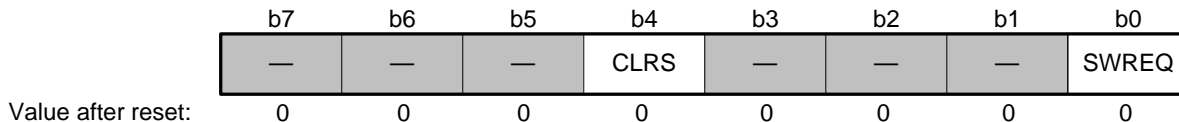
- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.

14.2.10 DMA Software Start Register (DMREQ)

Addresses: DMAC0.DMREQ 0008 201Dh, DMAC1.DMREQ 0008 205Dh, DMAC2.DMREQ 0008 209Dh, DMAC3.DMREQ 0008 20DDh



Bit	Symbol	Bit Name	Description	R/W
b0	SWREQ	DMACA Software Start	0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
b3 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b4	CLRS	DMACA Software Start Bit Auto Clear Select	0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

DMREQ starts DMACA transfer by software.

SWREQ Bit (DMACA Software Start)

When 1 is written to this bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is cleared to 0 if the CLRS bit is set to 0. This bit is not cleared to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DCTG[1:0] bits in DMTMD are set to 00b (DMA activation source is software).

Setting this bit is invalid when the DCTG[1:0] bits in DMTMD are set to a value other than 00b.

To start DMA transfer by software with the CLRS bit set to 0, check that the SWREQ bit is 0 and then write 1 to SWREQ.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

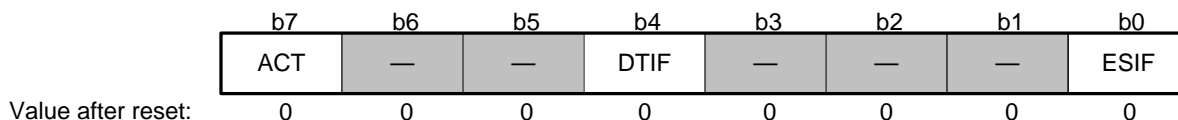
- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

CLRS Bit (DMACA Software Start Bit Auto Clear Select)

This bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

14.2.11 DMA Status Register (DMSTS)

Addresses: DMAC0.DMSTS 0008 201Eh, DMAC1.DMSTS 0008 205Eh,
DMAC2.DMSTS 0008 209Eh, DMAC3.DMSTS 0008 20DEh



Bit	Symbol	Bit Name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/(W)*
b3 to b1	—	Reserved	These bits are always read as 0. Writing to these bits has no effect.	R
b4	DTIF	Transfer End Interrupt Flag	0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/(W)*
b6, b5	—	Reserved	These bits are always read as 0. Writing to these bits has no effect.	R
b7	ACT	DMACA Active Flag	0: DMACA operation is suspended. 1: DMACA is operating.	R

Note : * Only 0 can be written to clear the flag.

DMSTS indicates DMACA transfer status.

ESIF Flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the RPTIE bit in DMINT set to 1.
- When 1-block data transfer is completed in block transfer mode with the RPTIE bit in DMINT set to 1.
- When an extended repeat area overflow on the source address occurs while the SARIE bit in DMINT is set to 1 and the SARA[4:0] bits in DMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs while the DARIE bit in DMINT is set to 1 and the DARA[4:0] bits in DMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer destination address)

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DTE bit in DMCNT.

DTIF Flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of DMCRAL becoming 0 on completion of transfer)
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of DMCRB becoming 0 on completion of transfer)
- When the specified number of blocks have been transferred in block transfer mode (the value of DMCRB becoming 0 on completion of transfer)

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DTE bit in DMCNT

ACT Bit (DMACA Active Flag)

This flag indicates whether the DMACA is in the idle or active state.

[Setting condition]

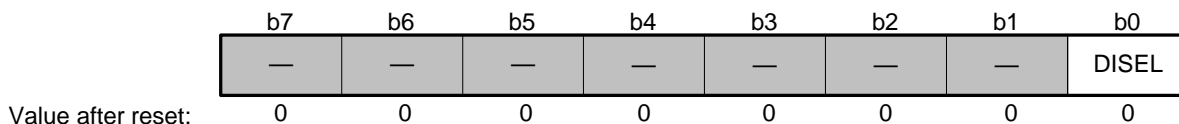
- When the DMACA starts data transfer operation

[Clearing condition]

- When data transfer in response to one transfer request is completed

14.2.12 DMA Activation Source Flag Control Register (DMCSL)

Addresses: DMAC0.DMCSL 0008 201Fh, DMAC1.DMCSL 0008 205Fh,
DMAC2.DMCSL 0008 209Fh, DMAC3.DMCSL 0008 20DFh



Bit	Symbol	Bit Name	Description	R/W
b0	DISEL	Interrupt Select	0: At the end of transfer, clear the interrupt flag of the activation source to 0. 1: At the end of transfer, the interrupt flag of the activation source issues an interrupt to the CPU.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

DMCSL controls the interrupt flag of the interrupt control unit (ICU), that is the activation source of the DMACA.

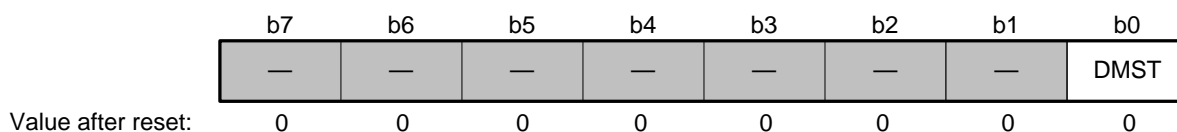
DISEL Bit (Interrupt Select)

This bit selects whether the interrupt flag of the activation source of the DMACA is cleared to 0 or issues an interrupt to the CPU, at the end of transfer.

When DMTMD.DCTG[1:0] = 00b (activation is by software), the setting of the DISE bit do not affect the operation.

14.2.13 DMA Module Start Register (DMAST)

Address: 0008 2200h



Bit	Symbol	Bit Name	Description	R/W
b0	DMST	DMACA Operation Enable	0: DMACA activation is disabled. 1: DMACA activation is enabled.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

DMAST enables or disables DMACA activation for all the channels.

DMST Bit (DMACA Operation Enable)

When the DMST bit is 1, the DMACA can be started in all the channels.

Before starting DMA transfer, set the DMST bit and DMCNT.DTE bit for the channel to 1.

[Setting condition]

- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

14.3 Operation

14.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMCRAL of DMACn. When these bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode). Setting DMCRB of DMACn is invalid in normal transfer mode. Except in free running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.

Table 14.4 summarizes the register update operation in normal transfer mode, and Figure 14.2 shows the operation in normal transfer mode.

Table 14.4 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
DMACn.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*
DMACn.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*
DMACn.DMCRAL	Transfer count	Decrement by one/not updated (in free running mode)
DMACn.DMCRAH	—	Not updated (Not used in normal transfer mode)
DMACn.DMCRB	—	Not updated (Not used in normal transfer mode)

Note: * Offset addition can be specified only for DMAC0.

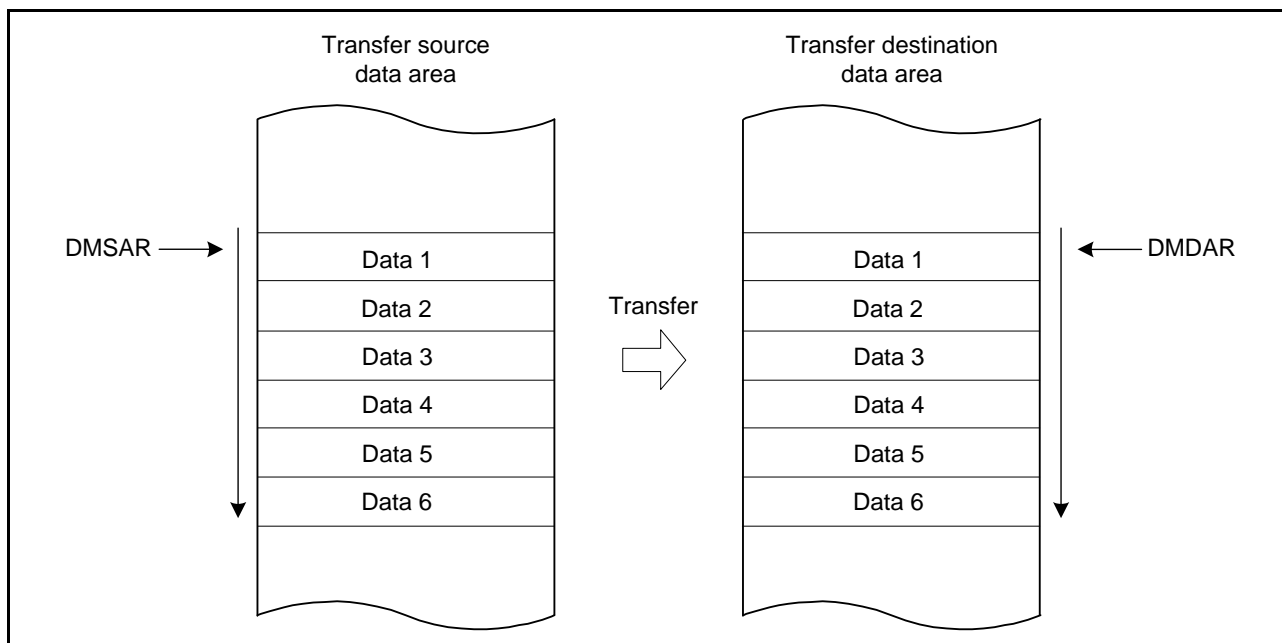


Figure 14.2 Operation in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request. A maximum of 1K data can be set as a total repeat transfer size using DMCRA of the DMACn.

A maximum of 1K can be set as the number of repeat transfer operations using DMCRB of the DMACn; therefore, a maximum of 1M data (1K data × 1K count of repeat transfer operations) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested.

A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.

Table 14.5 summarizes the register update operation in repeat transfer mode, and Figure 14.3 shows the operation in repeat transfer mode.

Table 14.5 Register Update Operation in Repeat Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request	
		When DMACn.DMCRAL is not 1	When DMACn.DMCRAL is 1 (Transfer of the Last Data in Repeat Size)
DMACn.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*	<ul style="list-style-type: none"> • DMACn.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition* • DMACn.DMTMD.DTS[1:0] = 01b Initial value of DMACn.DMSAR • DMACn.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*
DMACn.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*	<ul style="list-style-type: none"> • DMACn.DMTMD.DTS[1:0] = 00b Initial value of DMACn.DMDAR • DMACn.DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition* • DMACn.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*
DMACn.DMCRAH	Repeat size	Not updated	Not updated
DMACn.DMCRAL	Transfer count	Decrement by one	DMACn.DMCRAH
DMACn.DMCRB	Count of repeat transfer operations	Not updated	Decrement by one

Note: * Offset addition can be specified only for DMAC0.

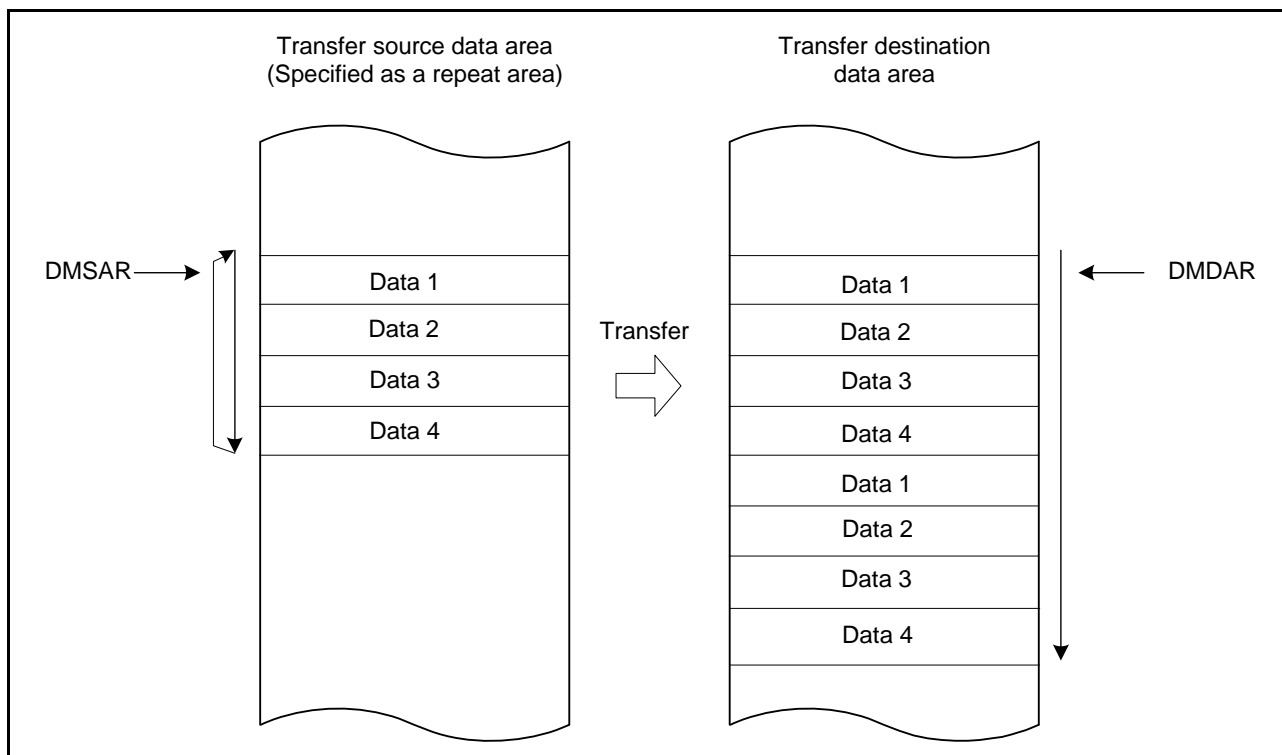


Figure 14.3 Operation in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request. A maximum of 1023 data can be set as a total block transfer size using DMCRA of the DMACn.

A maximum of 1K can be set as the number of block transfer operations using DMCRB of the DMACn; therefore, a maximum of 1023 data (1023 data × 1K block) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR of the DMACn) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested.

Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.

Table 14.6 summarizes the register update operation in block transfer mode, and Figure 14.4 shows the operation in block transfer mode.

Table 14.6 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
DMACn.DMSAR	Transfer source address	<ul style="list-style-type: none"> DMACn.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition* DMACn.DMTMD.DTS[1:0] = 01b Initial value of DMACn.DMSAR DMACn.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*
DMACn.DMDAR	Transfer destination address	<ul style="list-style-type: none"> DMACn.DMTMD.DTS[1:0] = 00 b Initial value of DMACn.DMDAR DMACn.DMTMD.DTS[1:0] = 01 b Increment/decrement/fixed/offset addition* DMACn.DMTMD.DTS[1:0] = 10 b Increment/decrement/fixed/offset addition*
DMACn.DMCRAH	Repeat size	Not updated
DMACn.DMCRAL	Transfer count	DMACn.DMCRAH
DMACn.DMCRB	Block count	Decrement by one

Note: * Offset addition can be specified only for DMAC0.

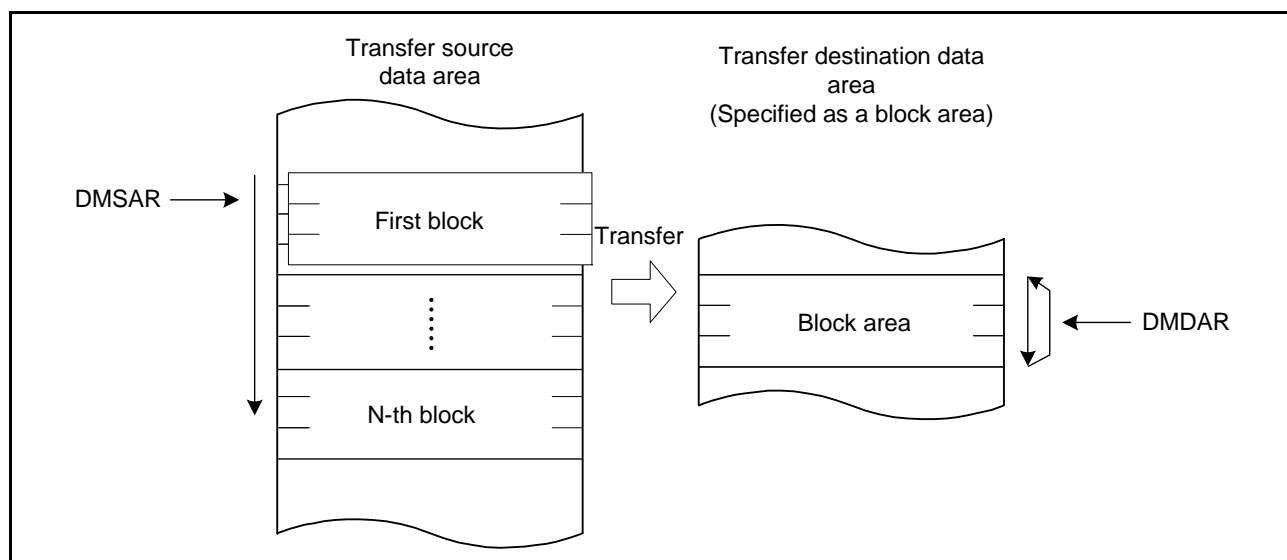


Figure 14.4 Operation in Block Transfer Mode

14.3.2 Extended Repeat Area Function

The DMACA supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR) of DMACn.

The extended repeat area on the source address is specified by the SARA[4:0] bits in DMAMD of DMACn. The extended repeat area on the destination address is specified by the DARA[4:0] bits in DMAMD of DMACn. The size can be specified separately for the source and destination sides. However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested to the CPU. When an overflow occurs in the extended repeat area on the transfer source while the SARIE bit in DMINT of DMACn is set to 1, the ESIF flag in DMSTS of DMACn is set to 1 and the DTE bit in DMINT of DMACn is cleared to 0 to stop DMA transfer. At this time, if the ESIE bit in DMINT of DMACn is set to 1, an interrupt by an extended repeat area overflow is requested to the CPU or DTC. When the DARIE bit in DMINT of DMACn is set to 1, an overflow on the extended repeat area set in DMDAR occurs, meaning that the destination side is a target.

Figure 14.5 shows an example of the extended repeat area operation.

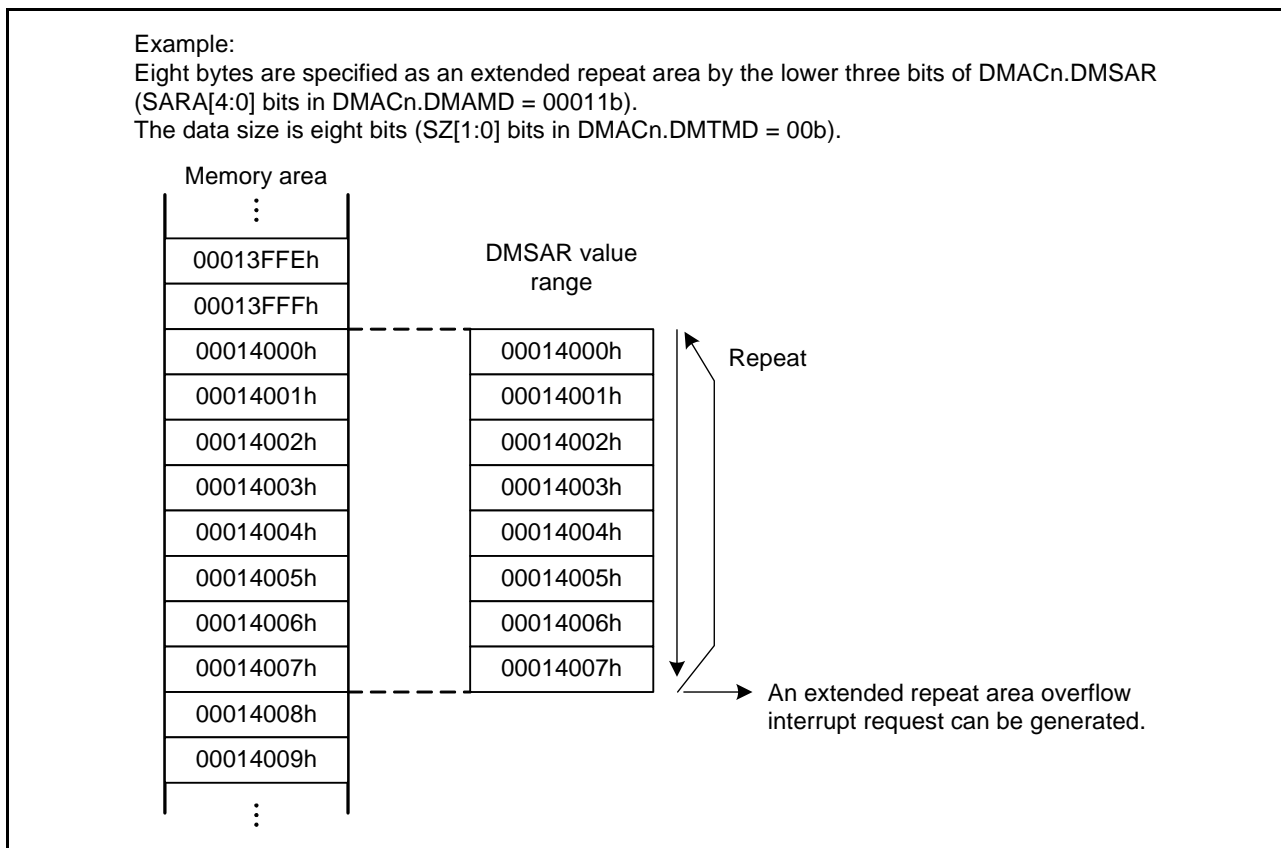


Figure 14.5 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 14.6 shows an example when the extended repeat area function is used in block transfer mode.

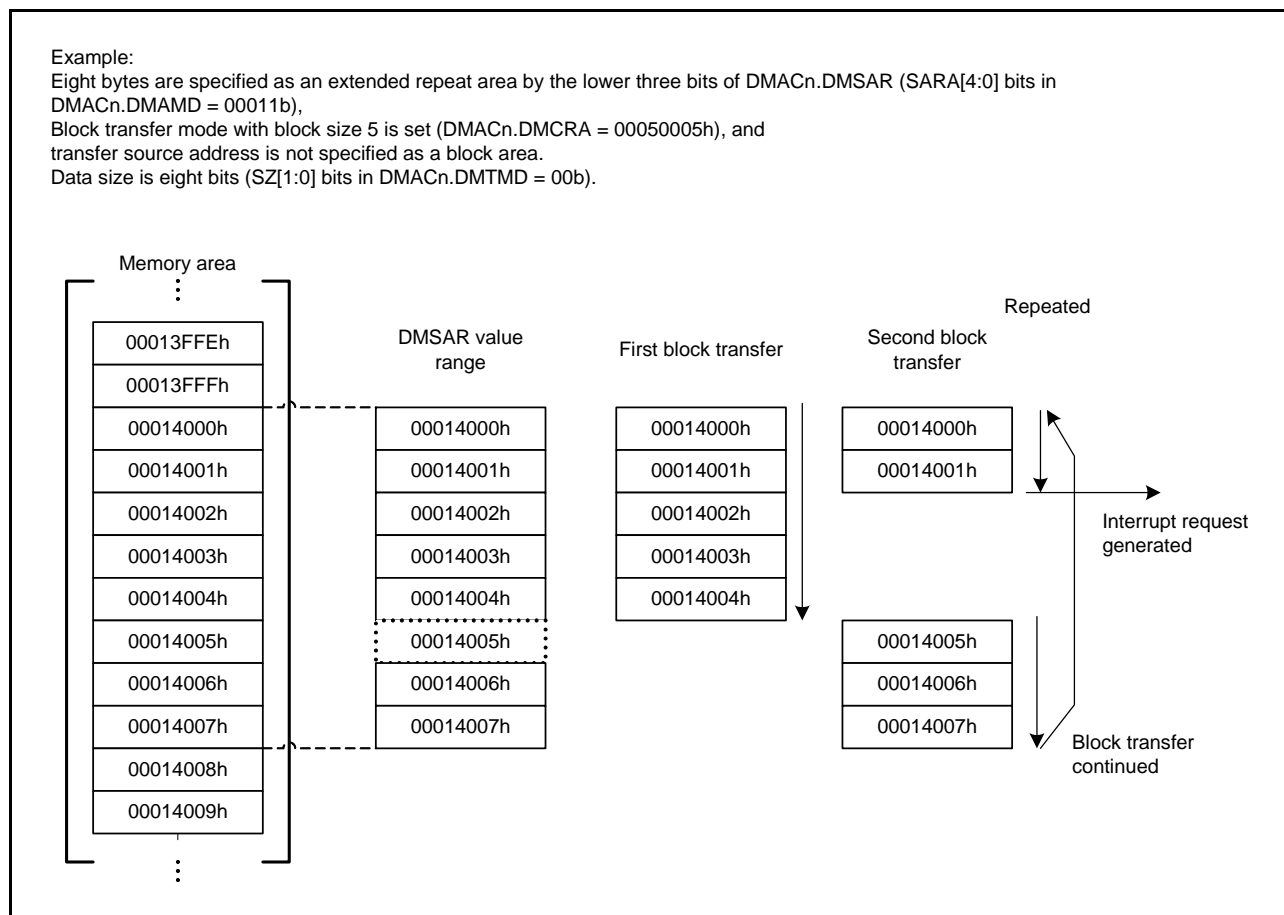


Figure 14.6 Example of Extended Repeat Area Function in Block Transfer Mode

14.3.3 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. When the offset addition is selected, the offset specified by the DMA offset register (DMOFR of DMAC0) is added to the address every time the DMACA performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in DMOFR of DMAC0. In this case, the negative value must be 2's complement.

Address update function using offset can be specified only for the DMAC0 channel.

Table 14.7 shows the address update method in each address update mode.

Table 14.7 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of DMACn.DMAMD.SM[1:0] and DMACn.DMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different SZ[1:0] Settings in DMTMD of DMACn)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMAC0.DMOFR*		
Increment	10b	+1	+2	+4
Decrement	11b	-1	-2	-4

Note: * When setting a negative value in the DMA offset register, the value must be 2's complement. The 2's complement is obtained by the following formula.

2's complement of a negative offset value = 1 + ~offset (~: bit inversion)

(1) Basic Transfer Using Offset

Figure 14.7 shows an example of address updating using offset addition.

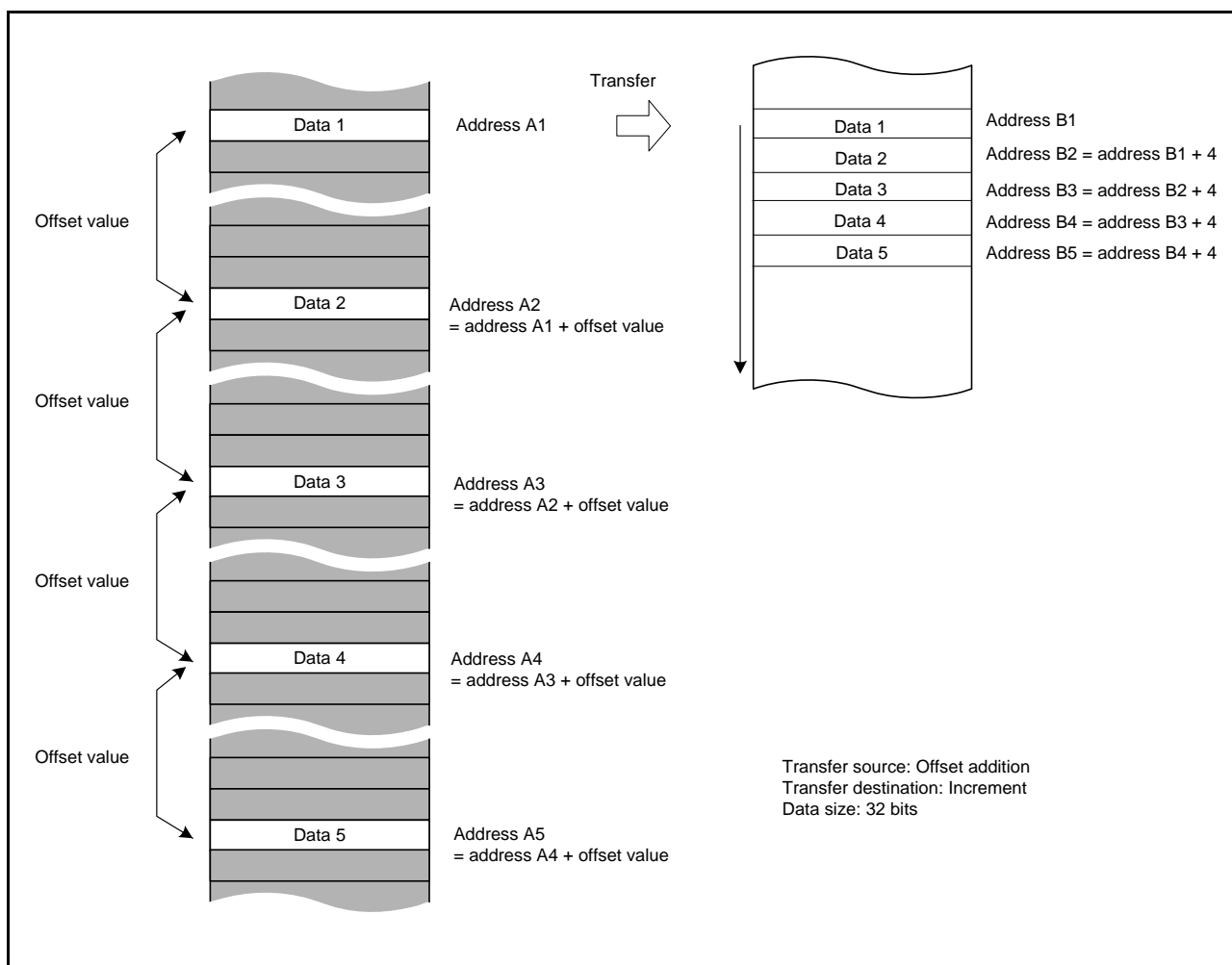


Figure 14.7 Example of Address Updating by Offset Addition

In Figure 14.7, the transfer data is 32 bits long, and offset addition and increment are set as the transfer source address update mode and transfer destination address update mode, respectively. The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

(2) Example of XY Conversion Using Offset Addition

Figure 14.8 shows the XY conversion using offset addition in repeat transfer mode.

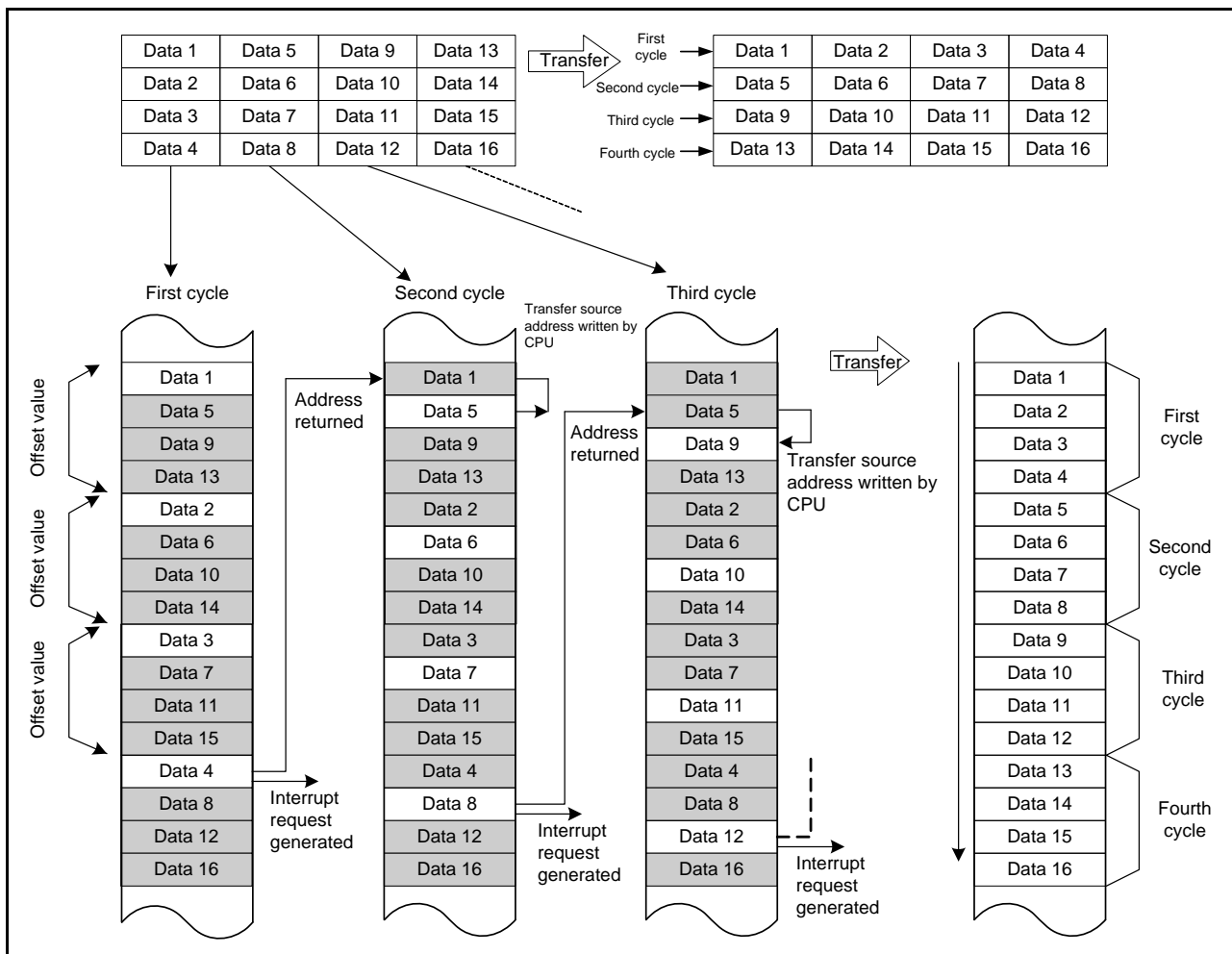


Figure 14.8 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

In Figure 14.8, the source address side is specified as the repeat area by the DMAC0.DMAMD register and the offset addition is selected. The offset value is set to $4 \times$ transfer data size (when the transfer data size is 32 bits, H'00000010 is set in DMOFR of DMAC0, as an example). The repeat size is set to $4 \times$ transfer data size (when transfer data size is 32 bits, the repeat size is set to $4 \times 4 = 16$ bytes, as an example). Increment is specified for the transfer destination address. A repeat size end interrupt is requested when the RPTIE bit in DMINT of DMAC0 is set to 1 and transfer of the repeat size of data is completed. When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, the transfer source address returns to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, write the address of data 5 to DMSAR of DMAC0 using the CPU (when the data access size is 32 bits, write the data 1 address + 4). When the DTE bit in DMCNT of DMAC0 is set to 1, the transfer is resumed by the following transfer request. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 14.9 shows a flowchart of the XY conversion.

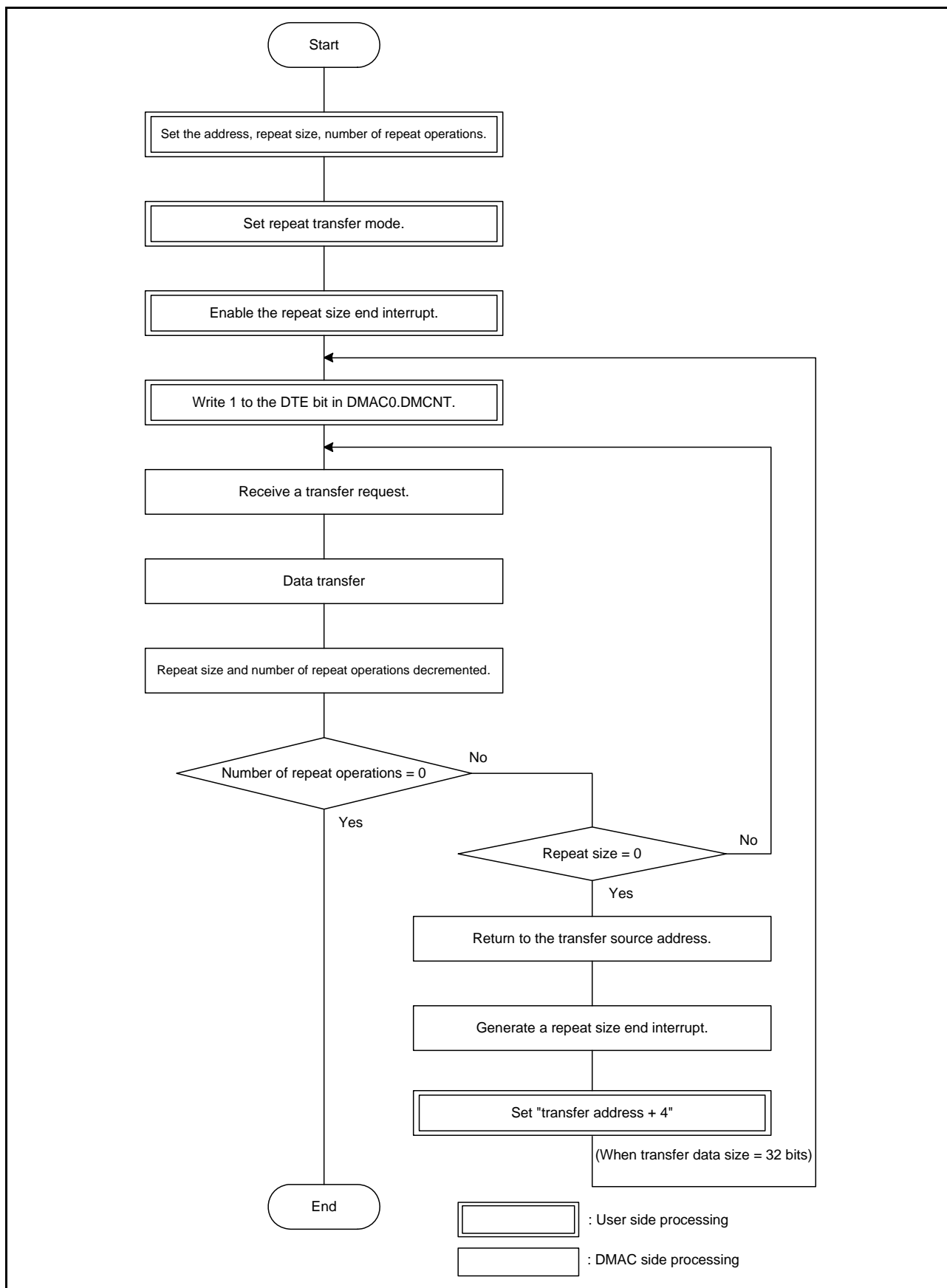


Figure 14.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

14.3.4 Activation Sources

Software, the interrupt requests from the peripheral modules, and the external interrupt requests can be specified as the DMACA activation sources. Setting the DCTG[1:0] bits in DMTMD of DMACn selects the activation source.

(1) DMACA Activation by Software

Setting the DCTG[1:0] bits in DMTMD of DMACn to 00b enables the DMACA activation by software.

To start DMA transfer by software, set the DCTG[1:0] bits in DMTMD to 00b, and then set the DTE bit in DMCNT to 1 (DMAC activation is enabled) and the SWREQ bit in DMREQ to 1 (DMA transfer is requested) with the DTE bit in DMCNT set to 1 (DMA transfer is enabled).

When the CLRS bit in DMREQ of DMACn is 0, the SWREQ bit in DMREQ of DMACn is cleared to 0 after data transfer is started in response to a DMA transfer request. When the CLRS bit is 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) DMACA Activation by Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

Interrupt requests from the on-chip peripheral modules and external interrupt requests can be specified as the DMACA activation sources. The activation source can be selected separately for each channel using the DMRSRn registers (n = 0 to 3) of the ICU. Table 14.8 lists the interrupt requests that can be specified as the DMACA activation sources.

The DMACA is activated when an interrupt request from the on-chip peripheral module or an external interrupt request is generated while the DCTG[1:0] bits in DMTMD of DMACn is set to 01b (interrupts from the peripheral modules and the external interrupt pins are selected), the DMST bit in DMAST is set to 1 (DMACA activation is enabled), and the DTE bit in DMCNT of DMACn is set to 1 (DMA transfer is enabled).

Table 14.8 Interrupt Requests Specified as DMACA Activation Sources (1 / 2)

DMA Request Sources (Interrupt Requests from On-Chip Peripheral Module or External Interrupt Requests)	ICU.DMRSRn
CMI0 (CMT0 compare match interrupt of compare match timer unit 0)	28
CMI1 (CMT1 compare match interrupt of compare match timer unit 1)	29
CMI2 (CMT2 compare match interrupt of compare match timer unit 2)	30
CMI3 (CMT3 compare match interrupt of compare match timer unit 3)	31
D0FIFO0 (USB0 D0FIFO transfer request)	36
D1FIFO0 (USB0 D1FIFO transfer request)	37
D0FIFO1 (USB1 D0FIFO transfer request)	40
D0FIFO1 (USB1 D0FIFO transfer request)	41
SPRI0 (RSPI0 receive buffer full interrupt)	45
SPTI0 (RSPI0 transmit buffer empty interrupt)	46
SPRI1 (RSPI1 receive buffer full interrupt)	49
SPTI1 (RSPI1 transmit buffer empty interrupt)	50
IRQ0 (external pin interrupt)	64
IRQ1 (external pin interrupt)	65
IRQ2 (external pin interrupt)	66
IRQ3 (external pin interrupt)	67
ADI0 (AD0 interrupt of A/D converter unit 0)	98
ADI1 (AD1 interrupt of A/D converter unit 1)	99
S12ADI0 (S12AD interrupt)	102
TGIA0 (MTU0 input capture/compare match interrupt of multi function timer pulse unit 0)	114
TGIA1 (MTU1 input capture/compare match interrupt of multi function timer pulse unit 0)	121
TGIA2 (MTU2 input capture/compare match interrupt of multi function timer pulse unit 0)	125

Table 14.8 Interrupt Requests Specified as DMACA Activation Sources (2 / 2)

DMA Request Sources (Interrupt Requests from On-Chip Peripheral Module or External Interrupt Requests	ICU.DMRSRn
TGIA3 (MTU3 input capture/compare match interrupt of multi function timer pulse unit 0)	129
TGIA4 (MTU4 input capture/compare match interrupt of multi function timer pulse unit 0)	134
TGIA6 (MTU6 input capture/compare match interrupt of multi function timer pulse unit 1)	142
TGIA7 (MTU7 input capture/compare match interrupt of multi function timer pulse unit 1)	149
TGIA8 (MTU8 input capture/compare match interrupt of multi function timer pulse unit 1)	153
TGIA9 (MTU9 input capture/compare match interrupt of multi function timer pulse unit 1)	157
TGIA10 (MTU10 input capture/compare match interrupt of multi function timer pulse unit 1)	162
RX10 (receive data full interrupt of serial communication interface SCI0)	215
RX10 (receive data full interrupt of serial communication interface SCI0)	215
TX10 (transmit data empty interrupt of serial communication interface SCI0)	216
RX11 (receive data full interrupt of serial communication interface SCI1)	219
TX11 (transmit data empty interrupt of serial communication interface SCI1)	220
RX12 (receive data full interrupt of serial communication interface SCI2)	223
TX12 (transmit data empty interrupt of serial communication interface SCI2)	224
RX13 (receive data full interrupt of serial communication interface SCI3)	227
TX13 (transmit data empty interrupt of serial communication interface SCI3)	228
RX15 (receive data full interrupt of serial communication interface SCI5)	235
TX15 (transmit data empty interrupt of serial communication interface SCI5)	236
RX16 (receive data full interrupt of serial communication interface SCI6)	239
TX16 (transmit data empty interrupt of serial communication interface SCI6)	240
ICRX10 (receive data full interrupt of I ² C bus interface RIIC0)	247
ICTX10 (transfer data empty interrupt of I ² C bus interface RIIC0)	248
ICRX11 (receive data full interrupt of I ² C bus interface RIIC1)	251
ICTX11 (transfer data empty interrupt of I ² C bus interface RIIC1)	252

14.3.5 Activating the DMACA

Figure 14.10 shows the register setting procedure.

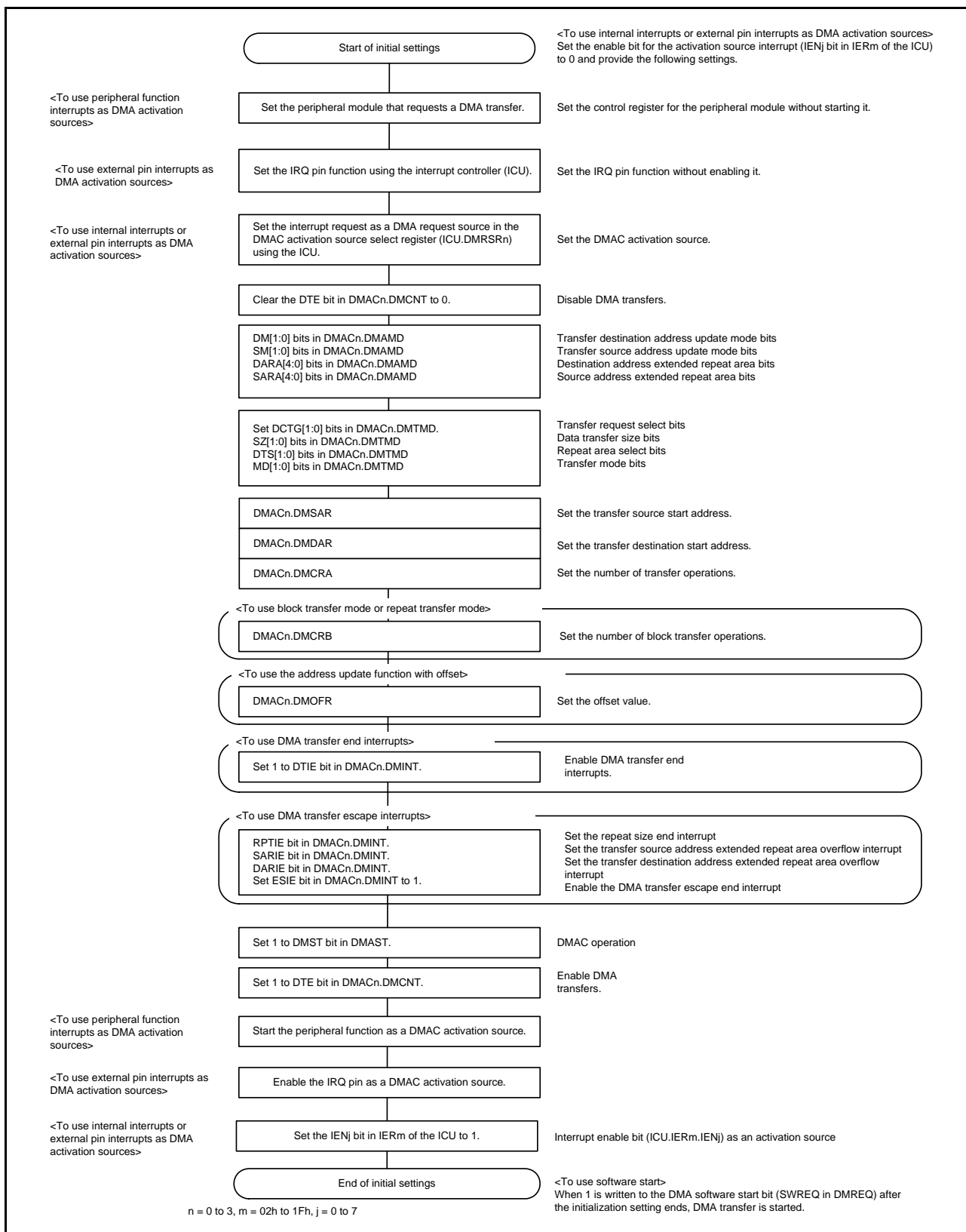


Figure 14.10 Register Setting Procedure

14.3.6 Starting DMA Transfer

Setting the DTE bit in DMCNT of DMAC to 1 (DMA transfer enabled) and setting the DMST bit in DMAST to 1 (DMACA start) enable DMA transfer of channel n (n = 0 to 3).

During DMACA transfer of other channels or DTC transfer, new activation requests cannot be accepted. When the current transfer ends, a DMA transfer request of highest-priority channel is accepted and DMA transfer of the channel starts. When a DMA transfer request is accepted and DMA transfer starts, the ACT bit in DMSTS is set to 1 (DMACA is in the active state).

14.3.7 Registers during DMA Transfer

The DMACA registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMCNT, and DMSTS of DMAC.

(1) DMA Source Address Register (DMACn.DMSAR)

When data has been transferred in response to one transfer request, the contents of DMSAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, refer to Table 14.4 to Table 14.6.

(2) DMA Destination Address Register (DMACn.DMDAR)

When data has been transferred in response to one transfer request, the contents of DMSAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, refer to Table 14.4 to Table 14.6.

(3) DMA Transfer Count Register (DMACn.DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, refer to Table 14.4 to Table 14.6.

(4) DMA Block Transfer Count Register (DMACn.DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, refer to Table 14.4 to Table 14.6.

(5) DMA Transfer Enable Bit (DMACn.DMCNT.DTE)

Although the DTE bit in DMCNT of DMACn enables or disables data transfer by the CPU write access, it is automatically cleared to 0 by the DMACA according to the DMA transfer state.

The conditions for clearing this bit by the DMACA are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt
- When 0 is written to this bit
- Reset state

Writing to the registers for the channels when the corresponding DTE bit in DMCNT is set to 1 is prohibited (except for DMCNT). In this case, writing must be performed after the bit is cleared to 0.

(6) DMA Active Flag (DMACn.DMSTS.ACT)

The ACT bit in DMSTS of DMAC indicates whether the DMAC is in the idle or active state.

This flag is set to 1 when the DMACA starts data transfer, and is cleared to 0 when data transfer in response to one transfer request is completed.

Even when DMA transfer is stopped by writing 0 to the DTE bit in DMCNT of DMAC, this flag remains 1 until DMA transfer is completed.

(7) Transfer End Interrupt Flag (DMACn.DMSTS.DTIF)

The DTIF bit in DMSTS of DMAC is set to 1 after transfer of the total transfer size of data is completed.

When both this flag and the DTIE bit in DMINT of DMAC are set to 1, a transfer end interrupt is requested.

This flag is set to 1 when the DMA transfer bus cycle is completed and the ACT flag in DMSTS of DMAC is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DTE bit in DMCNT of DMAC is set to 1 during the interrupt handling.

(8) Transfer Escape End Interrupt Flag (DMACn.DMSTS.ESIF)

The ESIF bit in DMSTS of DMAC is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the ESIE bit in DMINT of DMAC are set to 1, a transfer escape end interrupt is requested.

This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the ACT flag in DMSTS of DMAC is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DTE bit in DMCNT of DMAC is set to 1 during an interrupt handling.

Before sending an interrupt request from the DMACA to the CPU or the DTC, the interrupt control register must be set.

For details, see section 11, Interrupt Control Unit (ICUa).

14.3.8 Channel Priority

When multiple DMA requests are present, the DMACA determines the priority of channels that have DMA requests. The channel priority is fixed as channel 0 > channel 1 > channel 2 > channel 3 (channel 0: highest).

When a DMA request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

14.3.9 Operation Timing

Figure 14.11 and Figure 14.12 show timing examples of DMACA operation.

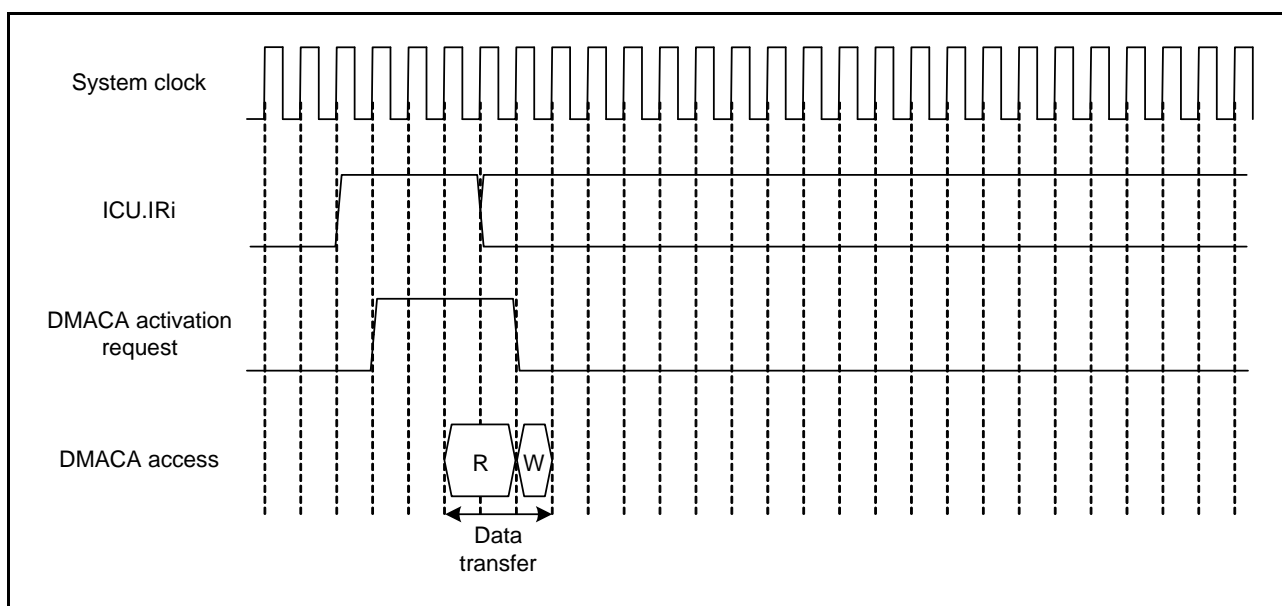


Figure 14.11 DMACA Operation Timing Example (1) (DMACA Started by Interrupt from Peripheral Module or External Interrupt Input Pin, in Normal Transfer Mode or Repeat Transfer Mode)

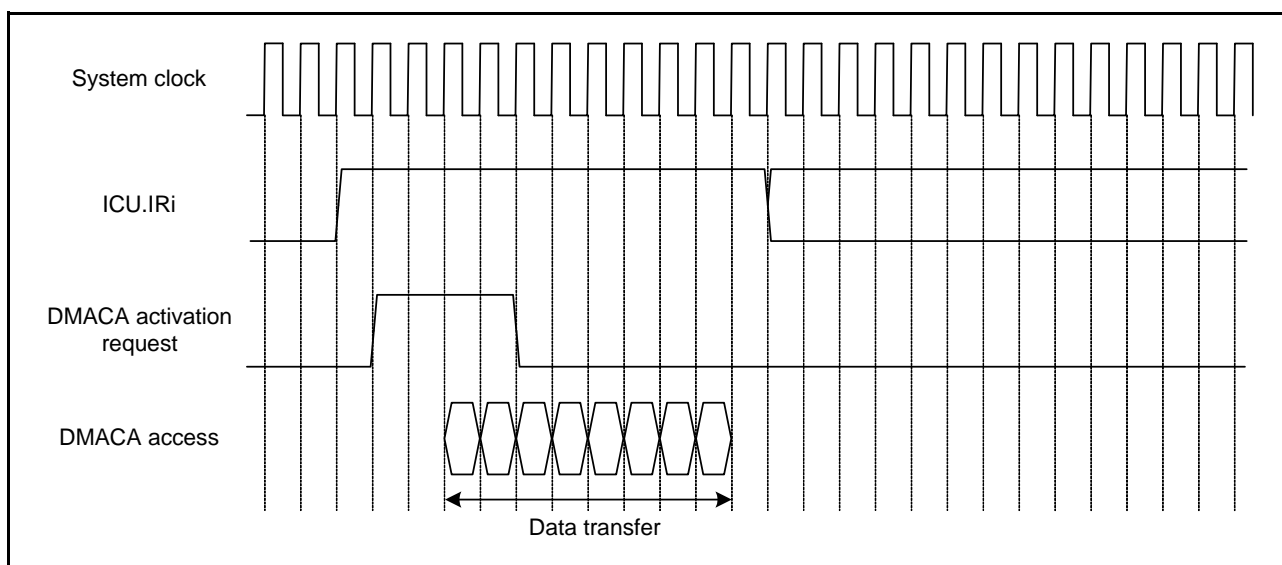


Figure 14.12 DMACA Operation Timing Example (2) (DMACA Started by Interrupt from Peripheral Module or External Interrupt Input Pin, in Block Transfer Mode with Block Size = 4)

14.3.10 DMACA Execution Cycles

Table 14.9 shows execution state in a single data transfer by the DMACA.

Table 14.9 DMACA Execution Cycles

Transfer Mode	Data Transfer (Read)	Data Transfer (Write)
Normal mode	Cr + 1	Cw
Repeat mode	Cr + 1	Cw
Block mode*1	Cr × P	Cw × P

Note 1. This indicates the case when the block size is two or more. When the block size is one, the number of cycles is the same as for normal transfer.

[Legend]

P: Block size (set by DMCRAH and DMCRAL)

Cr: Access cycles for data read destination

Cw: Access cycles for data write destination

(Cr and Cw values depend on the access destination. See section 37, RAM, section 38, ROM (Flash Memory for Code Storage), section 5, I/O Registers, and section 12.2.5, External Bus for the number of cycles for the access destination. The unit of "+1" in Data Read is system clock cycles (ICLK). For operation examples, see section 14.3.9, Operation Timing.)

14.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DTE bit in DMCNT and the ACT bit in DMSTS of DMACn are changed from 1 to 0, indicating that DMA transfer has ended.

14.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (DMACn.DMTMD.MD[1:0] = 00b)

When the value of the DMACn.DMCRAL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACn is cleared to 0 and the DTIF bit in DMSTS of DMACn is set to 1 at the same time. If the DTIE bit in DMINT of DMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (DMACn.DMTMD.MD[1:0] = 01b)

When the value of DMCRB of DMACn changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACn is cleared to 0 and the DTIF bit in DMSTS of DMACn is set to 1 at the same time. If the DTIE bit in DMINT of DMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(3) In Block Transfer Mode (DMACn.DMTMD.MD[1:0] = 10b)

When the value of DMCRB of DMACn changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACn is cleared to 0 and the DTIF bit in DMSTS of DMACn is set to 1 at the same time. If the DTIE bit in DMINT of DMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the DMACA to the CPU or the DTC, the interrupt control register must be set. For details, see section 11, Interrupt Control Unit (ICUa).

14.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the RPTIE bit in DMINT of DMACn is set to 1. When the interrupt is requested to complete DMA transfer, the DTE bit in DMCN of DMACn is cleared to 0 and the ESIF bit in DMSTS of DMACn is set to 1. If the ESIE bit in DMINT of DMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size data is completed.

Before sending an interrupt request from the DMACA to the CPU or the DTC, the interrupt control register must be set. For details, see section 11, Interrupt Control Unit (ICUa).

14.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the SARIE or DARIE bit in DMINT of DMACn is set to 1, an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DTE bit in DMCNT of DMACn is cleared to 0, and the ESIF bit in DMSTS of DMACn is set to 1. If the ESIE bit in DMINT of DMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.

In block transfer mode, even if an interrupt by an extended repeat area overflow is requested during a 1-block transfer, the remaining data in the block is transferred; transfer is terminated after a block transfer.

Before sending an interrupt request from the DMACA to the CPU or the DTC, the interrupt control register must be set. For details, see section 11, Interrupt Control Unit (ICUa).

14.5 Interrupts

The DMACA can output an interrupt request to the CPU or the DTC for each channel. Table 14.10 shows the relation among the interrupt sources, the interrupt status bits, and the interrupt enable bits. Figure 14.11 shows the schematic logic diagram of interrupt outputs.

Table 14.10 Relation among Interrupt Sources, Interrupt Status Bits, and Interrupt Enable Bits

Interrupt Sources	Interrupt Enable Bits	Interrupt Status Bits	Request Output Enable Bits
Transfer end	—	DMACn.DMSTS.DTIF	DMACn.DMINT.DTIE
Escape transfer end	Repeat size end	DMACn.DMINT.RPTIE	DMACn.DMINT.ESIE
	Source address extended repeat area overflow	DMACn.DMINT.SARIE	
	Destination address extended repeat area overflow	DMACn.DMINT.DARIE	

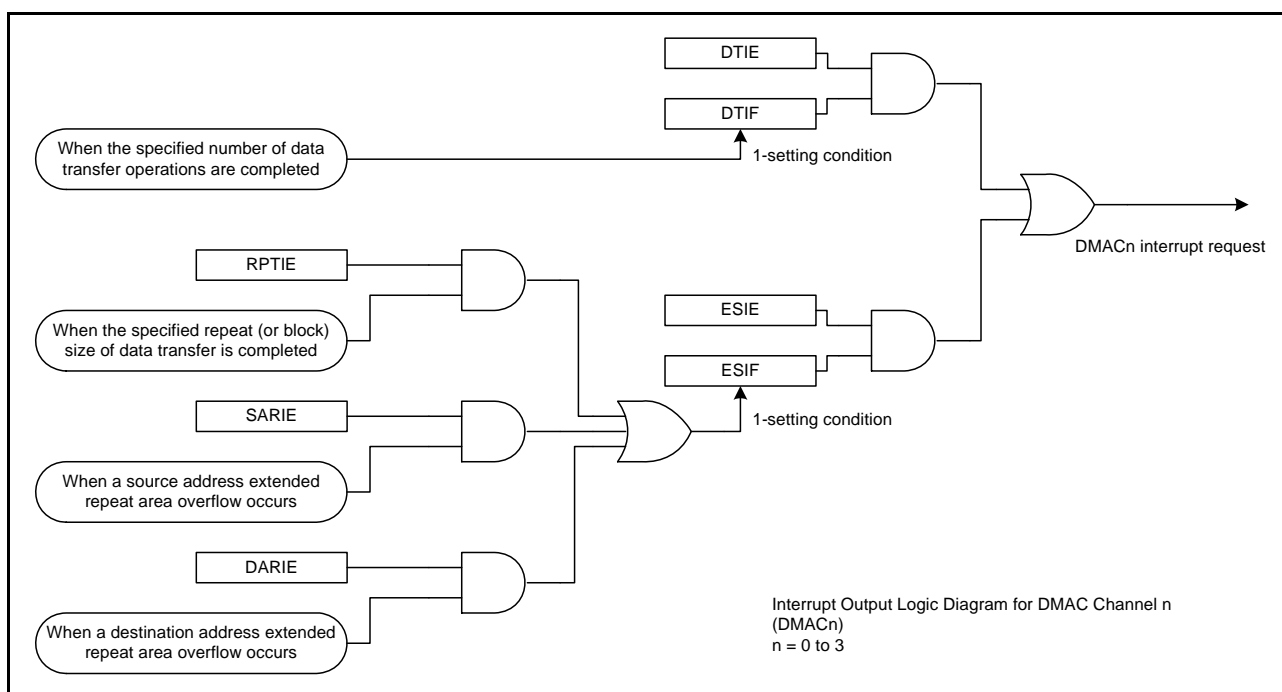


Figure 14.13 Schematic Logic Diagram of Interrupt Outputs

The interrupt cancellation method using the interrupt processing routine differs between when the DMA transfer ends or is stopped and when the transfer is continued.

(1) When transfer is performed with **DISEL** set to 0 and the transfer is continued after the next request generation

Set the interrupt priority level of the DMACA activation source to be lower than the IPL (processor interrupt priority level).

Execute the following processing at the DMACn interrupt.

Write 0 to the ICU.IERm.IENj bit for the activation source. Multiple-interrupt generation is prohibited until this processing ends.

- Write 1 to the DMCNT.DTE bit.
- Write 1 to the ICU.IERm.IENj bit for the activation source.

Note: m = 02h to 1Fh, j = 0 to 7

(2) When transfer is performed with **DISEL** set to 1 and the transfer is continued after the next request generation

Execute the following processing at the activation source interrupt.

- Check the DTIF and ESIF flags in DMSTS and if any of them is 1, write 0 to the ICU.IERm.IENj bit for the activation source.

Execute the following processing at the DMACn interrupt

- Write 1 to the DMCNT.DTE bit.
- Write 1 to the ICU.IERm.IENj bit for the activation source.

Note: m = 02h to 1Fh, j = 0 to 7

(3) When CPU interrupt processing or new transfer is provided after the next request generation (regardless of whether **DISEL** is 0 or 1)

Execute the following processing at the DMACn interrupt

- Write 0 to the ICU.IERm.IENj bit for the activation source.
- For a transfer end interrupt, write 0 to the DTIF flag in DMSTS. For a transfer escape interrupt, write 0 to the ESIF flag in DMSTS.
- To perform a new transfer, set registers according to the procedure in Figure 14.10. To end the transfer and generate a CPU interrupt, write 1 to the ICU.IERm.IENj bit for the activation source.

Note: m = 02h to 1Fh, j = 0 to 7

14.6 Low-Power Consumption Function

To place the DMACA in the module-stop state, all-module clock stop mode, software-standby mode, or deep software-standby mode, clear the ICU.IERm.IENj bit for the DMACA activation source to 0*, clear the DMAST.DMST and DMCNT.DTE bits to 0, and then perform the following processing.

Note: To generate a CPU interrupt as a restoration source from low-power consumption mode, write 1 to the ICU.IERm.IENj bit. For details of the setting, see section 11.6, Return from Power-Down States.

(1) Module Stop Function

Writing 1 to the MSTPA28 bit (transition to the module-stop state) in MSTPCRA enables the module-stop function of the DMACA. If DMA transfer is in progress at the time a 1 is written to the MSTPA28 bit, the transition to the module-stop state proceeds after DMAC transfer has ended. Writing 0 to the MSTPA28 bit releases the DMACA from the module-stop state.

(2) All-Module Clock Stop Mode

Writing 1 to the ACSE bit (all-module clock stop mode enabled) in MSTPCRA, writing 1 to all the bits in MSTPCRA and MSTPCRB, including the MSTPA28 bit (transition to the module-stop state), and executing a WAIT instruction causes a transition to the all-module clock stop mode. If DMA transfer is in progress at the time the WAIT instruction is executed, the DMAC can enter all-module clock stop mode after completion of the current DMA transfer.

After the DMAC returns from all-module clock stop mode, writing 0 to the MSTPA28 bit releases the DMACA from the module-stop state.

(3) Software Standby and Deep Software Standby Modes

Writing 1 to the SBYCR.SSBY bit (transition to software standby mode after WAIT instruction execution) and 0 to the DPSBYCR.DPSBY bit (transition to software standby mode after WAIT instruction execution), executing a WAIT instruction places the DMAC in software standby mode.

If DMA transfer is in progress at the time the WAIT instruction is executed, the DMAC enters software standby mode after completion of the current DMA transfer.

The DMAC enters deep software standby mode when the DPSBY bit in DPSBYCR is set to 1 (transition to deep software standby mode after WAIT instruction execution).

(4) Notes on Low-Power Consumption Function

For the timing of WAIT instruction execution and register settings, see section 9.7.7, Timing of Wait Instructions.

To continue DMACA activation after returning from low-power consumption mode, provide the settings according to the procedure shown in section 14.3.5, Activating the DMACA. To change the interrupt destination to the CPU or DTC, provide the settings according to the procedure shown section 11.4.3, Selecting Interrupt Request Destinations.

(m = 02h to 1Fh, j = 0 to 7)

14.7 Usage Notes

14.7.1 DMA Transfer to External Devices

In DMA transfer to an external device, the ACT bit in DMSTS of DMACAn may be cleared to 0 (DMACA transfer suspended) during the period from the beginning of the final data write to the end of the external bus access.

14.7.2 DMA Transfer to Peripheral Modules

In DMA transfer to a peripheral module, the ACT bit in DMSTS of DMACAn may be cleared to 0 (DMACA transfer suspended) during the period from the beginning of the final data write to the end of the peripheral bus access.

14.7.3 Access to the Registers during DMACA Transfer

The DMSAR, DMDAR, DMCRA, DMCRB, DMTMD, DMINT, DMAMD, DMOFR, and DMCSL registers of DMACn must not be accessed while the ACT bit in DMSTS of the same channel is set to 1 (DMACn operating state) or the DTE bit in DMCNT of the same channel is set to 1 (DMACA transfer enabled).

14.7.4 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

14.7.5 Interrupt Request by the DMACA Activation Source Flag Control Register (DMCSL) at the End of each Transfer

While the DMACAn.DMCSL.DISEL bit is 1, an interrupt is issued to the CPU at the end of each transfer that has been activated by the DMA. Unlike the transfer end interrupt that the DMACA outputs or the escape end interrupt, the interrupt of this type issues an interrupt to the CPU at the end of DMA transfer without clearing the interrupt flag of the DMACA activation source to 0 but changing the interrupt request destination to the CPU.

For the change of the settings on the interrupt flag or the interrupt request destination, see section 11, Interrupt Control Unit (ICUa). For the DMACAn.DMCSL.DISEL bit setting, see section 14.2.12, DMA Activation Source Flag Control Register (DMCSL).

14.7.6 Setting of DMACA Activation Source Select Register of the Interrupt Control Unit (ICU.DMRSRn)

The DMACA activation source select register (ICU.DMRSRn) should be set while the DMA transfer enable bit (DMACAn.DMCNT.DTE) is cleared to 0 (DMA transfer is disabled). Moreover, the DTC activation enable register (ICU.DTCERn) that corresponds to the same vector number that has been set by the ICU.DMRSRn register should not be set to 1. For details on the ICU.DTCERn and ICU.DMRSRn, see section 11, Interrupt Control Unit (ICUa).

14.7.7 Suspending or Restarting DMACA Activation

To suspend a DMACA activation request, write 0 to the interrupt enable bit for the activation source (ICU.IERm.IEN bit). To restart the DMA transfer, write 1 to the ICU.IERm.IEN bit with the setting shown in section 14.3.5, Activating the DMACA.

14.7.8 DMA Controller (DMACA)

In the RX62N and RX621 Groups, caution should be used when the communication function (SCI, RIIC, RSPI, USB) is used in combination with the DTC or DMACA function. For details, see section 11.7, Usage Notes.

15. EXDMA Controller (EXDMAC)

The RX62N/RX621 Group incorporates a 2-channel direct memory access controller (EXDMAC) designed exclusively for external bus transfer.

The EXDMAC is a module to transfer data without the CPU. When a DMA transfer request is generated, the EXDMAC transfers data stored at the transfer source address to the transfer destination address.

15.1 Overview

Table 15.1 lists the specifications of the EXDMAC, and Figure 15.1 shows a block diagram of the EXDMAC.

Table 15.1 Specifications of EXDMAC

Item		Description
Number of channels		2 (EXDMACn (n = 0, 1))
Transfer space		512 Mbytes (00000000h to 0FFFFFFFh and F0000000h to FFFFFFFFh excluding reserved areas)
Maximum transfer volume		1M data (Maximum number of transfer operations in block transfer mode: 1023 data × 1024 blocks)
DMA request source		<ul style="list-style-type: none"> Activation source selectable from the following three sources for each channel Software trigger Pins for external DMA transfer requests DMA transfer request from peripheral modules (compare match of MTU1)
Channel priority		Channel 0 > Channel 1 (Channel 0: Highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1023 data
	Cluster size	Number of data: 1 to 7 data
Transfer mode	Normal transfer mode	<ul style="list-style-type: none"> One data transfer by one DMA transfer request Free running mode (setting in which total number of data transfer operations is not specified) settable
	Repeat transfer mode	<ul style="list-style-type: none"> One data transfer by one DMA transfer request Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. Maximum settable repeat size: 1023 data
	Block transfer mode	<ul style="list-style-type: none"> One block data transfer by one DMA request Maximum settable block size: 1023 data
	Cluster transfer mode	<ul style="list-style-type: none"> One cluster data transfer by one DMA request Maximum settable cluster size: 7 data (28 bytes)
Address mode	Single address mode	<ul style="list-style-type: none"> Transfers data by accessing the transfer source or destination peripheral device with the EDACKn signal and specifying the address of the other peripheral device. Available in normal transfer mode, repeat transfer mode, and block transfer mode.
	Dual address mode	<ul style="list-style-type: none"> Transfers data by specifying the addresses of transfer source and destination. Available in normal transfer mode, repeat transfer mode, block transfer mode, and cluster transfer mode.
Selective functions	Extended repeat area function	<ul style="list-style-type: none"> Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed Area of two bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Interrupt request	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	Generated when the repeat size of data transfer is completed or the extended repeat area overflows.

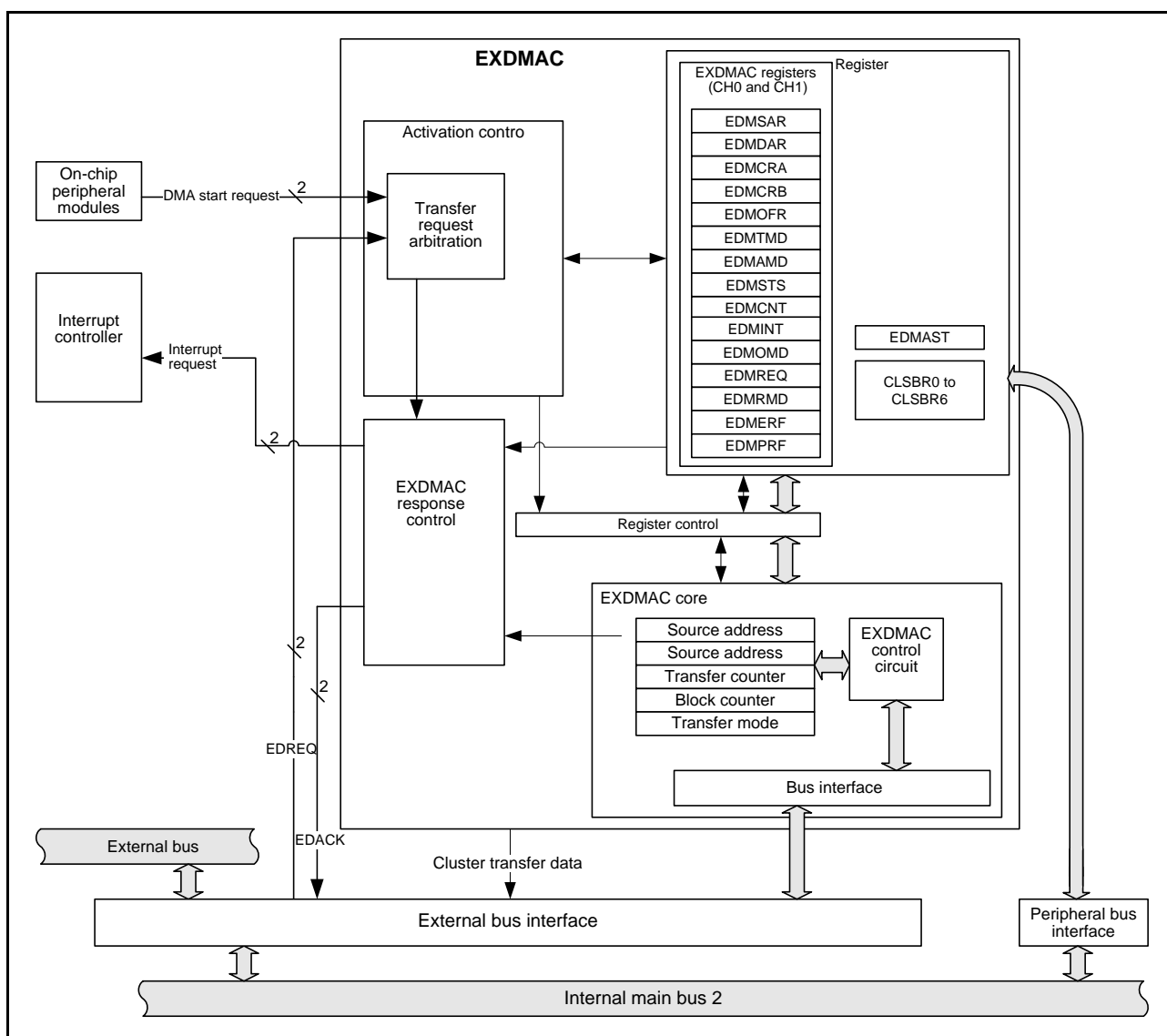


Figure 15.1 Block Diagram of EXDMAC

Table 15.2 lists the input/output pins of the EXDMAC.

Table 15.2 Pin Configuration of EXDMAC

Channel	Pin Name	I/O	Description
EXDMAC0	EDREQ0	Input	EXDMAC0 external DMA transfer requests
	EDACK0	Output	EXDMAC0 single address transfer acknowledge
EXDMAC1	EDREQ1	Input	EXDMAC1 external DMA transfer requests
	EDACK1	Output	EXDMAC1 single address transfer acknowledge

15.2 Register Descriptions

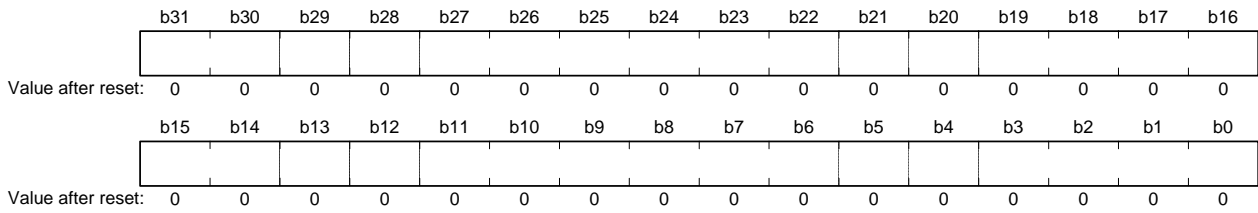
Table 15.3 lists the registers of the EXDMAC. Registers of EXDMAC0 and EXDMAC1 have same functions.

Table 15.3 Registers of EXDMAC

Channel	Register Name	Symbol	Value after Reset	Address	Access Size
EXDMAC0	EXDMA source address register	EDMSAR	0000 0000h	0008 2800h	32
	EXDMA destination address register	EDMDAR	0000 0000h	0008 2804h	32
	EXDMA transfer count register	EDMCRA	0000 0000h	0008 2808h	32
	EXDMA block transfer count register	EDMCRB	0000h	0008 280Ch	16
	EXDMA transfer mode register	EDMTMD	0000h	0008 2810h	16
	EXDMA output setting register	EDMOMD	00h	0008 2812h	8
	EXDMA interrupt setting register	EDMINT	00h	0008 2813h	8
	EXDMA address mode register	EDMAMD	0000 0000h	0008 2814h	32
	EXDMA offset register	EDMOFR	0000 0000h	0008 2818h	32
	EXDMA transfer enable register	EDMCNT	00h	0008 281Ch	8
	EXDMA software start register	EDMREQ	00h	0008 281Dh	8
	EXDMA status register	EDMSTS	00h	0008 281Eh	8
	EXDMA external request sense mode register	EDMRMD	00h	0008 2820h	8
	EXDMA external request flag register	EDMERF	00h	0008 2821h	8
	EXDMA peripheral request flag register	EDMPRF	00h	0008 2822h	8
EXDMAC1	EXDMA source address register	EDMSAR	0000 0000h	0008 2840h	32
	EXDMA destination address register	EDMDAR	0000 0000h	0008 2844h	32
	EXDMA transfer count register	EDMCRA	0000 0000h	0008 2848h	32
	EXDMA block transfer count register	EDMCRB	0000h	0008 284Ch	16
	EXDMA transfer mode register	EDMTMD	0000h	0008 2850h	16
	EXDMA output setting register	EDMOMD	00h	0008 2852h	8
	EXDMA interrupt setting register	EDMINT	00h	0008 2853h	8
	EXDMA address mode register	EDMAMD	0000 0000h	0008 2854h	32
	EXDMA transfer enable register	EDMCNT	00h	0008 285Ch	8
	EXDMA software start register	EDMREQ	00h	0008 285Dh	8
	EXDMA status register	EDMSTS	00h	0008 285Eh	8
	EXDMA external request sense mode register	EDMRMD	00h	0008 2860h	8
	EXDMA external request flag register	EDMERF	00h	0008 2861h	8
	EXDMA peripheral request flag register	EDMPRF	00h	0008 2862h	8
	EXDMAC	EXDMA module start register	EDMAST	00h	0008 2A00h
Cluster buffer register 0		CLSBR0	0000 0000h	0008 2BE0h	32
Cluster buffer register 1		CLSBR1	0000 0000h	0008 2BE4h	32
Cluster buffer register 2		CLSBR2	0000 0000h	0008 2BE8h	32
Cluster buffer register 3		CLSBR3	0000 0000h	0008 2BEC	32
Cluster buffer register 4		CLSBR4	0000 0000h	0008 2BF0h	32
Cluster buffer register 5		CLSBR5	0000 0000h	0008 2BF4h	32
Cluster buffer register 6		CLSBR6	0000 0000h	0008 2BF8h	32
Cluster buffer register 7		CLSBR7	0000 0000h	0008 2BFCh	32

15.2.1 EXDMA Source Address Register (EDMSAR)

Addresses: EXDMAC0.EDMSAR 0008 2800h, EXDMAC1.EDMSAR 0008 2840h



Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source start address.	00000000h to 0FFFFFFFh (256 Mbytes) F0000000h to FFFFFFFFh (256 Mbytes)	R/W

EDMSAR specifies the start address of the transfer source.

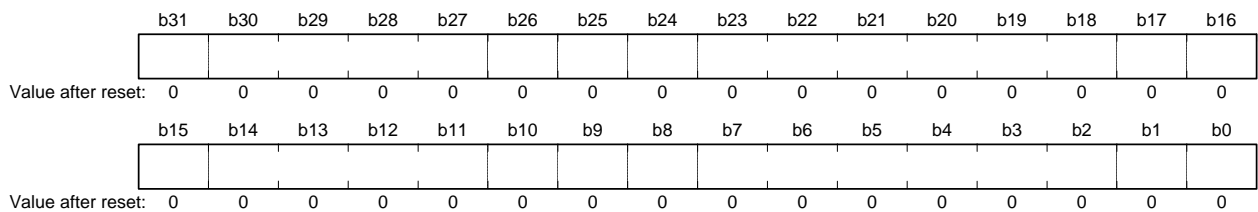
Set EDMSAR while EXDMAC activation is disabled (the DMST bit in EDMAST = 0) or DMA transfer is disabled (the DTE bit in EDMCNT = 0).

Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading EDMSAR returns the extended value.

EDMSAR must be accessed in 32 bits.

15.2.2 EXDMA Destination Address Register (EDMDAR)

Addresses: EXDMAC0.EDMDAR 0008 2804h, EXDMAC1.EDMDAR 0008 2844h



Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	00000000h to 0FFFFFFFh (256 Mbytes) F0000000h to FFFFFFFFh (256 Mbytes)	R/W

EDMDAR specifies the start address of the transfer destination.

Set EDMDAR while EXDMAC activation is disabled (the DMST bit in EDMAST = 0) or DMA transfer is disabled (the DTE bit in EDMCNT = 0).

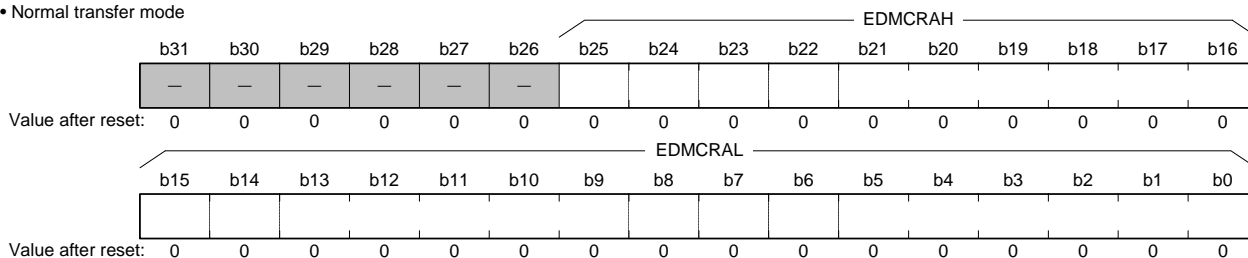
Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading EDMDAR returns the extended value.

EDMDAR must be accessed in 32 bits.

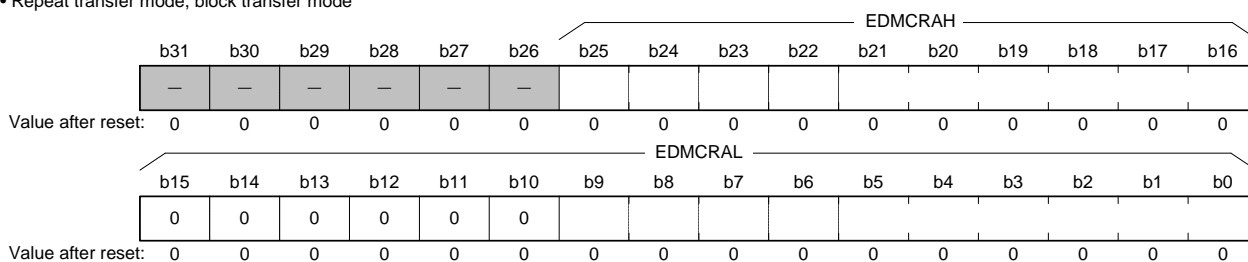
15.2.3 EXDMA Transfer Count Register (EDMCRA)

Addresses EXDMAC0.EDMCRA 0008 2808h, EXDMAC1.EDMCRA 0008 2848h

• Normal transfer mode

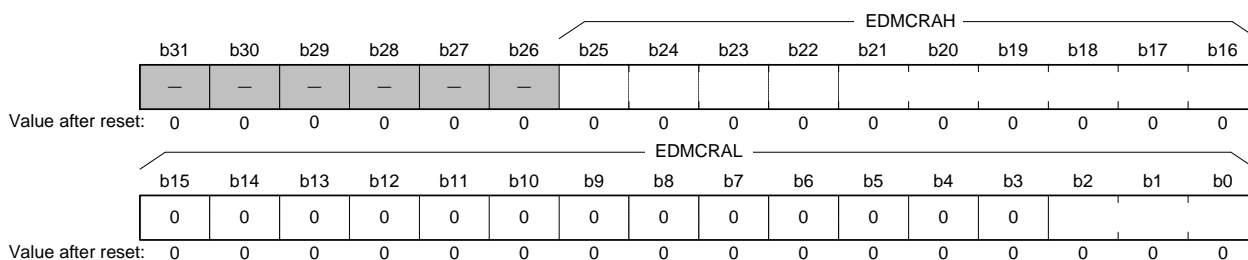


• Repeat transfer mode, block transfer mode



Note: The function differs depending on the transfer mode.

• Cluster transfer mode



Symbol	Bit Name	Description	R/W
EDMCRAL	Lower bits of transfer count	Specifies the number of transfer operations	R/W
EDMCRAH	Upper bits of transfer count		R/W

Note: * Set the same value for EDMCRAH and EDMCRAL in repeat transfer mode, block transfer mode, and cluster transfer mode.

EDMCRA specifies the number of DMA transfers. The functions of EDMCRAL and EDMCRAH depend on the transfer mode as described below.

(1) Normal transfer mode (MD[1:0] bits in EXDMACn.EDMTMD = 00b)

EDMCRAL functions as a 16-bit transfer counter.

The number of transfer operations is one when the setting is 0001h, and 65535 when it is FFFFh. The value is decremented by one each time data is transferred.

When the setting is 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode).

EDMCRAH is not used in normal transfer mode. Write 0000h to EDMCRAH.

(2) Repeat transfer mode (MD[1:0] bits in EXDMACn.EDMTMD = 01b)

EDMCRAH specifies the repeat size and EDMCRAL functions as a 10-bit transfer counter.

The number of transfer operations is one when the setting is 001h and 1023 when it is 3FFh. In repeat transfer mode, a value in the range of 001h to 3FFh can be set for EDMCRAH and EDMCRAL. Setting a value of 000h is prohibited.

Setting bits 15 to 10 in EDMCRAL is invalid. Write 0 to these bits.

The value in EDMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in EDMCRAH is loaded into EDMCRAL.

(3) Block transfer mode (MD[1:0] bits in EXDMACn.EDMTMD = 10b)

EDMCRAH specifies the block size and EDMCRAL functions as a 10-bit block size counter.

The number of transfers is one when the setting is 001h and 1023 when it is 3FFh. In block transfer mode, a value in the range of 001h to 3FFh can be set for EDMCRAH and EDMCRAL. Setting a value of 000h is prohibited.

Setting bits 15 to 10 in EDMCRAL is invalid. Write 0 to these bits.

The value in EDMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in EDMCRAH is loaded into EDMCRAL.

(4) Cluster transfer mode (MD[1:0] bits in EXDMACn.EDMTMD = 11b)

EDMCRAH specifies the cluster size and EDMCRAL functions as a 3-bit cluster size counter.

The number of transfers is one when the setting is 001h and seven when it is 007h. In cluster transfer mode, a value in the range of 001h to 007h can be set for EDMCRAH and EDMCRAL. Setting a value of 000h is prohibited.

Setting bits 15 to 3 in EDMCRAL is invalid. Write 0 to these bits.

The value in EDMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in EDMCRAH is loaded into EDMCRAL.

15.2.4 EXDMA Block Transfer Count Register (EDMCRB)

Addresses: EXDMAC0.EDMCRB 0008 280Ch, EXDMAC1.EDMCRB 0008 284Ch



Bit	Description	Setting Range	R/W
b9 to b0	Specifies the number of block transfer operations, repeat transfer operations, or cluster transfer operations.	001h to 3FFh (1 to 1023) 000h (1024)	R/W
b15 to b10	Reserved	These bits are always read as 0. The write value should be 0.	R/W

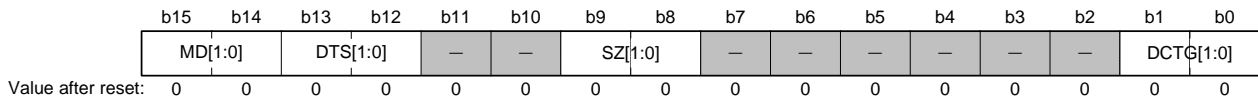
EDMCRB specifies the number of block transfer operations in block transfer mode, the number of repeat transfer operations in repeat transfer mode, or the number of cluster transfer operations in cluster transfer mode.

The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. The value is decremented by one each time data is transferred.

In normal transfer mode, a value of 3FFh should be set.

15.2.5 EXDMA Transfer Mode Register (EDMTMD)

Addresses: EXDMAC0.EDMTMD 0008 2810h, EXDMAC1.EDMTMD 0008 2850h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCTG[1:0]	DMA Request Source Select	b1 b0 0 0: Software 0 1: Setting prohibited 1 0: External DMA transfer request pin (EDREQn) 1 1: DMA transfer requests from the on-chip peripheral modules (compare match of MTU1)	R/W
b7 to b2	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Transfer Data Size Select	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
b11, b10	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Cluster transfer	R/W

Note 1. * One of the DMA request sources can be selected for each channel from among peripheral modules.

EDMTMD specifies the DMA transfer mode.

DCTG[1:0] Bits (DMA Request Source Select)

DCTG[1:0] select the EXDMAC activation source: activation by software, activation by an external DMA transfer request pin, or activation by DMA transfer requests from the on-chip peripheral modules (compare match of MTU1).

SZ[1:0] Bits (Transfer Data Size Select)

SZ[1:0] select the data size of one transfer from among 8, 16, and 32 bits.

DTS[1:0] Bits (Repeat Area Select)

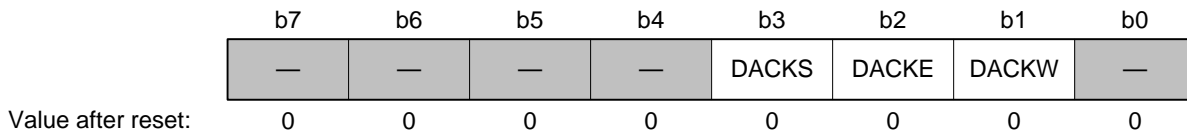
DTS[1:0] select either the transfer source or transfer destination as the repeat area in repeat, block, or cluster transfer mode.

MD[1:0] Bits (Transfer Mode Select)

MD[1:0] select the DMA transfer mode from among normal transfer mode, repeat transfer mode, block transfer mode, and cluster transfer mode.

15.2.6 EXDMA Output Setting Register (EDMOMD)

Addresses: EXDMAC0.EDMOMD 0008 2812h, EXDMAC1.EDMOMD 0008 2852h



Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W
b1	DACKW	EDACKn Pin Negate Wait	0: The EDACKn pin is negated at the same time as the RD# or WRn# pin is negated. 1: The EDACKn is negated one BCLK cycle before the RD# pin is negated or one BCLK cycle after the WRn# pin is negated.	R/W
b2	DACKE	EDACKn Pin Output Enable	0: EDACKn output is disabled. 1: EDACKn output is enabled.	R/W
b3	DACKS	EDACKn Pin Output Polarity Select	0: EDACKn pin polarity is active low. 1: EDACKn pin polarity is active high.	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

EDMOMD specifies the EXDMAC output signals.

DACKW Bit (EDACKn Pin Negate Wait)

DACKW selects the EDACKn pin negation timing in single address mode during normal, repeat, or block transfer to/from the CS area.

During the above transfer in single address mode (AMS bit in EDMAMD = 1), the EDACKn pin is negated at the same time as the RD# or WRn# pin is negated if this bit is 0; and the EDACKn pin is negated one BCLK cycle before the RD# pin is negated, or one BCLK cycle after the WRn# pin is negated if this bit is 1. In single address mode during normal, repeat, or block transfer to/from the SDRAM, setting this bit is invalid. EDACKn pin negation timing cannot be changed. Setting this bit is also invalid in dual address mode and during cluster transfer. In these cases, the EDACKn pin does not provide output.

DACKE Bit (EDACKn Pin Output Enable)

DACKE enables or disables EDACKn pin output. Setting this bit is invalid in dual address mode and during cluster transfer (EDACKn pin output not provided).

DACKS Bit (EDACKn Pin Output Polarity Select)

DACKS selects the EDACKn pin output polarity.

15.2.7 EXDMA Interrupt Setting Register (EDMINT)

Addresses: EXDMAC0.EDMINT 0008 2813h, EXDMAC1.EDMINT 0008 2853h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

EDMINT enables/disables the following EXDMAC interrupt requests.

DARIE Bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the DTE bit in EDMCNT is cleared to 0. At the same time, the ESIF bit in EDMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DTE bit in EDMCNT of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

SARIE Bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while this bit is set to 1, the DTE bit in EDMCNT is cleared to 0. At the same time, the ESIF bit in EDMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DTE bit in EDMCNT of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

RPTIE Bit (Repeat Size End Interrupt Enable)

When this bit is set to 1 in repeat transfer mode, the DTE bit in EDMCNT is cleared to 0 after completion of a 1-repeat size data transfer. At the same time, the ESIF bit in EDMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in EDMTMD are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DTE bit in EDMCNT is cleared to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the ESIF bit in EDMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in EDMTMD are 10b (= repeat area or block area is not specified.)

When this bit is set to 1 in cluster transfer mode, the DTE bit in EDMCNT is cleared to 0 after completion of a 1-cluster data transfer in the same way as repeat transfer mode. At the same time, the ESIF bit in EDMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in EDMTMD are 10b (= repeat area or block area is not specified.)

ESIE Bit (Transfer Escape End Interrupt Enable)

This bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.

The transfer escape end interrupt is generated when the ESIF bit in EXDMACn.EDMSTS is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by clearing this bit or the ESIF bit in EDMSTS to 0.

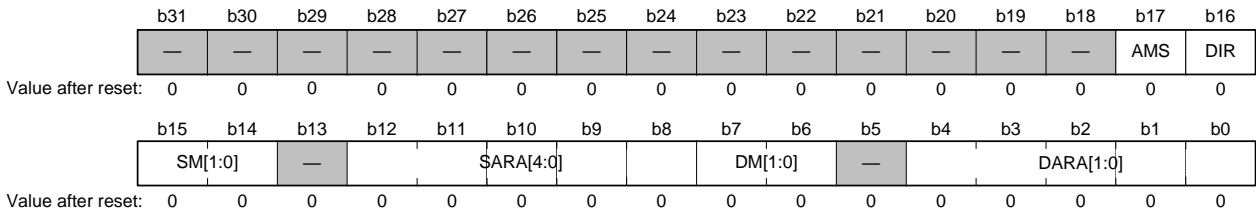
DTIE Bit (Transfer End Interrupt Enable)

This bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DTIF bit in EDMSTS is set to 1 with this bit set to 1. The transfer end interrupt is cleared by clearing this bit or the DTIF bit in EDMSTS to 0.

15.2.8 EXDMA Address Mode Register (EDMAMD)

Addresses: EXDMAC0.EDMAMD 0008 2814h, EXDMAC1.EDMAMD 0008 2854h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area	Specifies the extended repeat area on the destination address. For details on the settings, see Table 15.4.	R/W
b5	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition* 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area	Specifies the extended repeat area on the source address. For details on the settings, see Table 15.4.	R/W
b13	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W
b15, b14	SM[1:0]	Source Address Update Mode	b15 b14 0 0: Destination address is fixed. 0 1: Offset addition* 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b16	DIR	Single Address Direction Select	0: Data is transferred in single address mode using the EDMSAR register value as the transfer source address. EDACKn is output to the transfer destination. 1: Data is transferred in single address mode using the EDMDAR register value as the transfer destination address. EDACKn is output to the transfer destination.	R/W
b17	AMS	Address Mode Select	0: Dual address mode 1: Single address mode	R/W
b31 to b18	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note: * Offset addition can be specified only for EXDMAC0.

EDMAMD specifies the EDMAC address mode.

DARA[4:0] Bits (Destination Address Extended Repeat Area)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 21 (2 bytes) and 217 (128 Mbytes).

When the lower address overflows the extended repeat area by address increment, the top address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the bottom address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer, block transfer, or cluster transfer is selected, or when EXDMACn.EDMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write "00000b" in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DARIE bit in EDMINT set to 1. Table 15.4 shows the settings and the corresponding extended repeat areas.

DM[1:0] Bits (Destination Address Update Mode)

These bits select the mode of updating the destination address.

When increment is selected and the SZ[1:0] bits in EDMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in EDMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the EXDMAC 0.EDMOFR register is added to the address. Offset addition can be specified only for EXDMAC0.

SARA[4:0] Bits (Source Address Extended Repeat Area)

These bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 21 (2 bytes) and 217 (128 Mbytes).

When the lower address overflows the extended repeat area by address increment, the top address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the bottom address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer, block transfer, or cluster transfer is selected, or when EXDMACn.EDMTMD.DTS[1:0] = 01b (the transfer destination is specified as the repeat area or block area), write "00000b" in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the SARIE bit in EDMINT set to 1. Table 15.4 shows the settings and the corresponding extended repeat areas.

SM Bit (Source Address Update Mode)

These bits select the mode of updating the source address.

When increment is selected and the SZ[1:0] bits in EDMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in EDMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the EXDMAC0.EDMOFR register is added to the address. Offset addition can be specified only for EXDMAC0.

DIR Bit (Single Address Direction Select)

This bit selects the transfer destination or source, to which the addresses should be output in single address mode.

- Normal/Repeat/Block Transfer

When this bit is set to 0, data is transferred in single address mode using the EDMSAR register value as the transfer source address. Here, EDACKn can be output to the transfer destination device by setting the DACK bit in EDMOMD to 1 (EDACKn output is enabled).

When this bit is set to 1, data is transferred in single address mode using the EDMDAR register value as the transfer destination address. Here, EDACKn can be output to the transfer source device by setting the DACK bit in EDMOMD to 1 (EDACKn output is enabled). Setting the DIR bit is valid when the AMS bit in EDMAMD is 1 (single address mode); setting the DIR bit is invalid when the AMS bit is 0 (dual address mode).

- Cluster Transfer

When this bit is set to 0, data is transferred in cluster transfer read address mode using the EDMSAR register value as the transfer source address. Here, data can be transferred to the cluster buffers from the external device.

When this bit is set to 1, data is transferred in cluster transfer write address mode using the EDMDAR register value as the transfer destination address. Here, data can be transferred to the external device from the cluster buffers.

Setting the DIR bit is valid when the AMS bit in EDMAMD is 1 (single address mode); setting the DIR bit is invalid when the AMS bit is 0 (dual address mode).

AMS Bit (Address Mode Select)

This bit selects the address mode.

- Normal/Repeat/Block Transfer

When this bit is set to 0, dual address mode is selected and when set to 1, single address mode is selected.

When using single address mode, select the transfer source or destination device to which the addresses should be output using the DIR bit in EDMAMD.

- Cluster Transfer

When this bit is set to 0, dual address mode is selected and when set to 1, read or write address mode is selected.

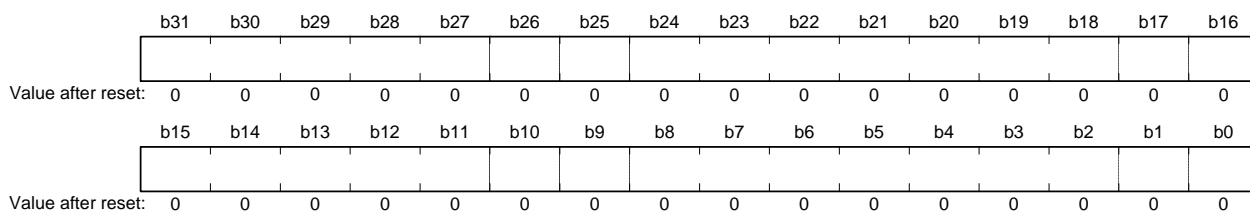
Select read or write address mode using the DIR bit in EDMAMD.

Table 15.4 SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas

SARA4 to SARA0 or DARA4 to DARA0	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	(Setting prohibited)

15.2.9 EXDMA Offset Register (EDMOFR)

Address: EXDMAC0.EDMOFR 0008 2818h



Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	00000000h to 00FFFFFFh (0 bytes to (16M – 1) bytes) FF000000h to FFFFFFFFh (–16M bytes to –1 byte)	R/W

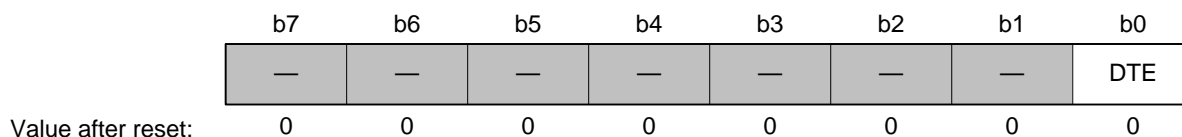
EDMOFR specifies the address offset.

Write to this register while the EXDMAC operation is stopped or DMA transfer is disabled (not during data transfer).

Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading EDMOFR returns the extended value.

15.2.10 EXDMA Transfer Enable Register (EDMCNT)

Addresses: EXDMAC0.EDMCNT 0008 281Ch, EXDMAC1.EDMCNT 0008 285Ch



Bit	Symbol	Bit Name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

EDMCNT enables or disables DMA transfer for the corresponding channel.

DTE Bit (DMA Transfer Enable)

When the DMST bit in EDMAST is set to 1 (EXDMAC activation is enabled) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

When the DTE bit is cleared to 0 during DMA transfer, DMA transfer is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer is resumed by setting the DTE bit to 1 again.

When a DTE bit is 1, writing to registers of the corresponding EXDMAC channel other than that containing the DTE bit is prohibited.

[Setting condition]

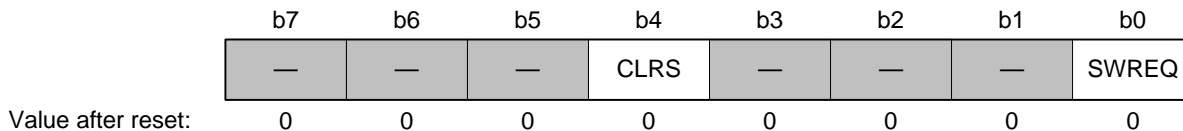
- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.

15.2.11 EXDMA Software Start Register (EDMREQ)

Addresses: EXDMAC0.EDMREQ 0008 281Dh, EXDMAC1.EDMREQ 0008 285Dh



Bit	Symbol	Bit Name	Description	R/W
b0	SWREQ	DMA Software Start	0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
b3 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b4	CLRS	DMA Software Start Bit Auto Clear Select	0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

EDMREQ starts DMA transfer by software.

SWREQ Bit (DMA Software Start)

When 1 is written to this bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is cleared to 0 if the CLRS bit is set to 0. This bit is not cleared to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DCTG[1:0] bits in EDMTMD are set to 00b (DMA activation source is software).

Setting this bit is invalid when the DCTG[1:0] bits in EDMTMD are set to a value other than 00b.

To start DMA transfer by software with CLRS set to 0, check that the SWREQ bit is 0 and then write 1 to SWREQ.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

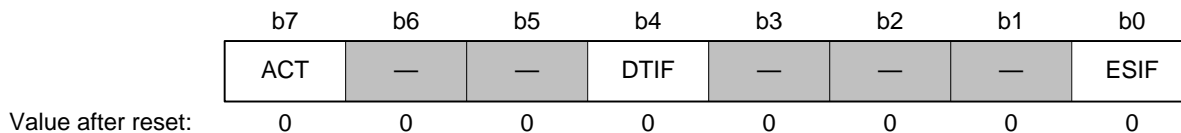
- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

CLRS Bit (DMA Software Start Bit Auto Clear Select)

This bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

15.2.12 EXDMA Status Register (EDMSTS)

Addresses: EXDMAC0.EDMSTS 0008 281Eh, EXDMAC1.EDMSTS 0008 285Eh



Bit	Symbol	Bit Name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W
b3 to b1	—	Reserved	These bits are always read as 0. Writing to these bits has no effect.	R
b4	DTIF	Transfer End Interrupt Flag	0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W
b6, b5	—	Reserved	These bits are always read as 0. Writing to these bits has no effect.	R
b7	ACT	EXDMA Active Flag	0: EXDMAC operation is suspended. 1: EXDMAC is operating.	R

EDMSTS indicates DMA transfer status.

ESIF Flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the RPTIE bit in EDMINT set to 1.
- When 1-block data transfer is completed in block transfer mode with the RPTIE bit in EDMINT set to 1.
- When 1-cluster data transfer is completed in cluster transfer mode with the RPTIE bit in EDMINT set to 1.
- When an extended repeat area overflow on the source address occurs while the SARIE bit in EDMINT is set to 1 and the SARA[4:0] bits in EDMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs while the DARIE bit in EDMINT is set to 1 and the DARA[4:0] bits in EDMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer destination address)

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DTE bit in EDMCNT.

DTIF Flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of EDMCRAL becoming 0 on completion of transfer)
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of EDMCRB becoming 0 on completion of transfer)
- When the specified number of blocks have been transferred in block transfer mode (the value of EDMCRB becoming 0 on completion of transfer)
- When the specified number of clusters have been transferred in cluster transfer mode (the value of EDMCRB becoming 0 on completion of transfer)

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DTE bit in EDMCNT

ACT Bit (EXDMA Active Flag)

This flag indicates whether the EXDMAC is in the idle or active state.

[Setting condition]

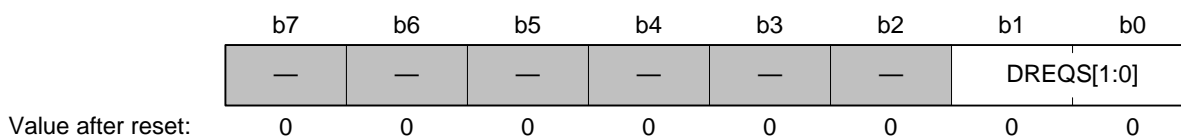
- When the EXDMAC starts data transfer operation

[Clearing condition]

- When data transfer in response to one transfer request is completed

15.2.13 EXDMA External Request Sense Mode Register (EDMRMD)

Addresses: EXDMAC0.EDMRMD 0008 2820h, EXDMAC1.EDMRMD 0008 2860h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	DREQS[1:0]	Request Input Sense Mode Set	b1 b0 0 0: Rising edge 0 1: Falling edge 1 0: Low level 1 1: (Setting prohibited)	R/W
b7 to b2	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

The EDMRMD register specifies the sense mode for the EDREQn pin.

DREQS[1:0] Bits (EDREQn Pin Sense Mode Set)

These bits specify the sense mode for the external DMA transfer request signal (EDREQn pin).

15.2.14 EXDMA External Request Flag Register (EDMERF)

Addresses: EXDMAC0.EDMERF 0008 2821h, EXDMAC1.EDMERF 0008 2861h

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

Bit	Symbol	Bit Name	Description	R/W
b0	EREQ	External Request Flag	This flag indicates the DMA transfer request from the EDREQn pin. 0: No request 1: Requested	R/(W)*
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note: * Writing 0 has no effect.

The EDMERF register detects the request from the EDREQn pin.

EREQ Flag (External Request Flag)

This flag detects the request from the external DMA transfer request signal (EDREQn pin).

[Setting conditions]

- When the level on the EDREQn pin changes from 0 to 1 while EXDMACn.EDMRMD.DREQS[1:0] = 00b
- When the level on the EDREQn pin changes from 1 to 0 while EXDMACn.EDMRMD.DREQS[1:0] = 01b
- When the level on the EDREQn pin is 0 while EXDMACn.EDMRMD.DREQS[1:0] = 10b (low level)

[Clearing conditions]

- When the EREQ flag is set to 1 (requested state) while EXDMACn.EDMRMD.DREQS[1:0] = 00b (rising edge) or 01b (falling edge) and then the DMA transfer is started by the external request
- When 1 is written to this flag while EXDMACn.EDMRMD.DREQS[1:0] = 00b (rising edge) or 01b (falling edge)
- When the EDREQn pin is set to 1 while EXDMACn.EDMRMD.DREQS[1:0] = 10b (low level)

15.2.15 EXDMA Peripheral Request Flag Register (EDMPRF)

Addresses: EXDMAC0.EDMPRF 0008 2822h, EXDMAC1.EDMPRF 0008 2862h

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

Bit	Symbol	Bit Name	Description	R/W
b0	PREQ	Peripheral Module Request Flag	This flag indicates the DMA transfer request from the peripheral module. 0: No request 1: Requested	R/(W)*
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note: * Writing 0 has no effect.

The EDMPRF register indicates the DMA transfer request from the peripheral module.

PREQ Flag (Peripheral Module Request Flag)

This flag indicates the DMA transfer request from the peripheral module.

[Setting condition]

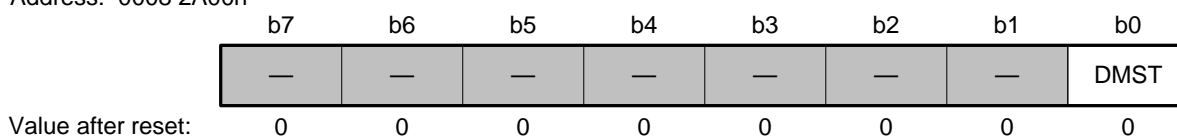
- When the DMA transfer request is generated from the peripheral module

[Clearing conditions]

- When the DMA transfer request is generated from the peripheral module, the PREQ flag is set to 1 (requested state) and the DMA transfer is started by the DMA transfer request from the peripheral module
- When 1 is written to this flag

15.2.16 EXDMA Module Start Register (EDMAST)

Address: 0008 2A00h



Bit	Symbol	Bit Name	Description	R/W
b0	DMST	EXDMAC Operation Enable	0: EXDMAC activation is disabled. 1: EXDMAC activation is enabled.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

EDMAST enables or disables EXDMAC activation for all the channels.

DMST Bit (EXDMAC Operation Enable)

When this bit is set to 1, EXDMAC activation is enabled for all the channels.

When 1 is written to the DTE bit in EDMCNT (DMA transfer is enabled) of multiple EXDMAC channels and then this bit is set to 1 (EXDMAC activation is enabled), the corresponding multiple channels can be placed in the transfer request acceptable state at the same time.

When the DMST bit is cleared to 0 during DMA transfer, DMA transfer for all channels is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer is resumed by setting the DMST bit to 1 again.

[Setting condition]

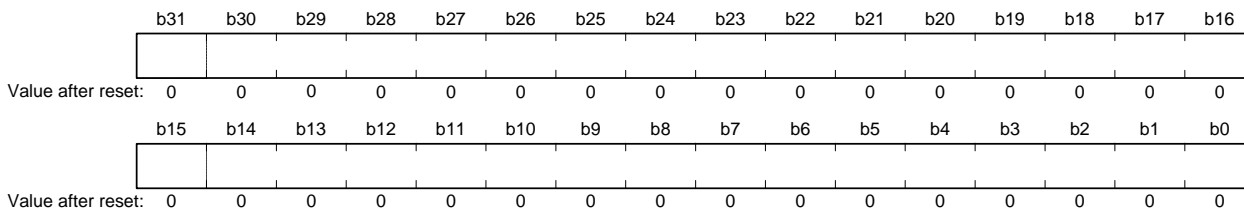
- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

15.2.17 Cluster Buffer Register i (CLSBRi) (i = 0 to 6)

Addresses: CLSBR0 0008 2BE0h, CLSBR1 0008 2BE4h, CLSBR2 0008 2BE8h, CLSBR3 0008 2BEC h,
 CLSBR4 0008 2BF0h, CLSBR5 0008 2BF4h, CLSBR6 0008 2BF8h, CLSBR7 0008 2BFC h



Bit	Description	R/W
b31 to b1	Buffer area for cluster transfer.	R/W

CLSBRi are buffer registers for cluster transfer. During cluster transfer, transferred data is sequentially stored in CLSBRi starting from CLSBR0. The cluster-transferred data or data written by the CPU is retained until another cluster transfer or data write by the CPU. When reading the cluster-transferred data with the CPU, confirm that cluster transfer has been completed and only refer to the data of the specified size for cluster; the other data is invalid.

During cluster transfer, the same CLSBRi is used for all the channels. If a conflict occurs between the write to CLSBRi by the CPU and cluster transfer, transferred data is not guaranteed. If a channel is set to cluster transfer in read or write address mode and another channel is set to cluster transfer, data to be transferred may be erroneously modified.

Data is stored in cluster buffers in the different manner depending on the transfer size setting (SZ[1:0] bits in EDMTMD).

- (1) Transfer Size is 8 Bits (EXDMACn.EDMTMD.SZ[1:0] = 00b)
 Data is stored in the lower 8 bits in the cluster buffers (CLSBRn[7:0]). Here, the upper 24 bits (CLSBRn[31:8]) are invalid. When the maximum cluster size is set to 7, 7-byte data is one cluster.
 Data is stored in CLSBR in the order of CLSBR0 to CLSBRj (j = cluster size value – 1).
- (2) Transfer Size is 16 Bits (EXDMACn.EDMTMD.SZ[1:0] = 01b)
 Data is stored in the lower 16 bits in the cluster buffers (CLSBRn[15:0]). Here, the upper 16 bits (CLSBRn[31:16]) are invalid. When the maximum cluster size is set to 7, 14-byte data is one cluster.
 Data is stored in CLSBR in the order of CLSBR0 to CLSBRj (j = cluster size value – 1).
- (3) Transfer Size is 32 Bits (EXDMACn.EDMTMD.SZ[1:0] = 10b)
 Data is stored in all the 32 bits in the cluster buffers (CLSBRn[31:0]). When the maximum cluster size is set to 7, 28-byte data is one cluster.
 Data is stored in CLSBR in the order of CLSBR0 to CLSBRj (j = cluster size value – 1).

15.3 Operation

15.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using EDMCRA of EXDMACn. When the EXDMACn.EDMCRAL bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode). Setting EDMCRB of EXDMACn is invalid in normal transfer mode. Except in free running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.

Table 15.5 summarizes the register update operation in normal transfer mode, and Figure 15.2 shows the operation in normal transfer mode.

Table 15.5 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
EXDMACn.EDMSAR	Transfer source address	Increment/decrement/offset addition*1
EXDMACn.EDMDAR	Transfer destination address	Increment/decrement/offset addition*1
EXDMACn.EDMCRAL	Transfer count	Decremented by one/not updated (in free running mode)
EXDMACn.EDMCRAH	Block size	Not updated (Not used in normal transfer mode)
EXDMACn.EDMCRB	Block count	Not updated (Not used in normal transfer mode)

Note 1. Offset addition can be specified only for EXDMAC0.

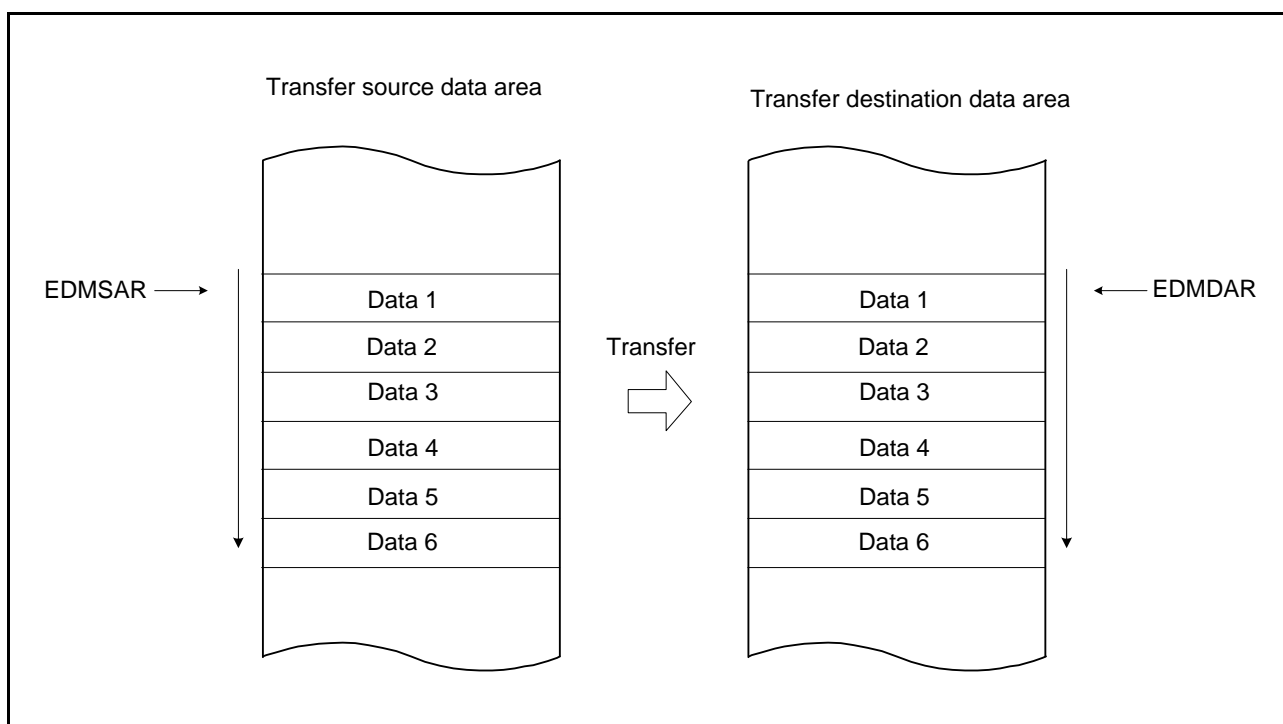


Figure 15.2 Operation in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request. A maximum of 1023 data can be set as a total repeat transfer size using EDMCRA of the EXDMACn.

A maximum of 1K can be set as the number of repeat transfer operations using EDMCRB of the EXDMACn; therefore, a maximum of 1023K data (1023 data × 1K) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (EXDMACn.EDMSAR or EXDMACn.EDMDAR) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in EDMCNT of EXDMACn in the repeat size end interrupt handling.

A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.

Table 15.6 summarizes the register update operation in repeat transfer mode, and Figure 15.3 shows the operation in repeat transfer mode.

Table 15.6 Register Update Operation in Repeat Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request	
		When EXDMACn.EDMCRAL is not 1	When EXDMACn.EDMCRAL is 1 (Transfer of the Last Data in Repeat Size)
EXDMACn.EDMSAR	Transfer source address	Increment/decrement/fixe d/offset addition*	<ul style="list-style-type: none"> • EXDMACn.EDMTMD.DTS[1:0] = 00 Increment/decrement/fixe d/offset addition* • EXDMACn.EDMTMD.DTS[1:0] = 01 Initial value of EXDMACn.EDMSAR • EXDMACn.EDMTMD.DTS[1:0] = 10 Increment/decrement/fixe d/offset addition*
EXDMACn.EDMDAR	Transfer destination address	Increment/decrement/fixe d/offset addition*	<ul style="list-style-type: none"> • EXDMACn.EDMTMD.DTS[1:0] = 00 Initial value of EXDMACn.EDMDAR • EXDMACn.EDMTMD.DTS[1:0] = 01 Increment/decrement/fixe d/offset addition* • EXDMACn.EDMTMD.DTS[1:0] = 10 Increment/decrement/fixe d/offset addition*
EXDMACn.EDMCRAH	Repeat size	Not updated	Not updated
EXDMACn.EDMCRAL	Transfer count	Decremente d by one	EXDMACn.EDMCRAH
EXDMACn.EDMCRB	Block count	Not updated	Decremente d by one

Note: * Offset addition can be specified only for EXDMAC0.

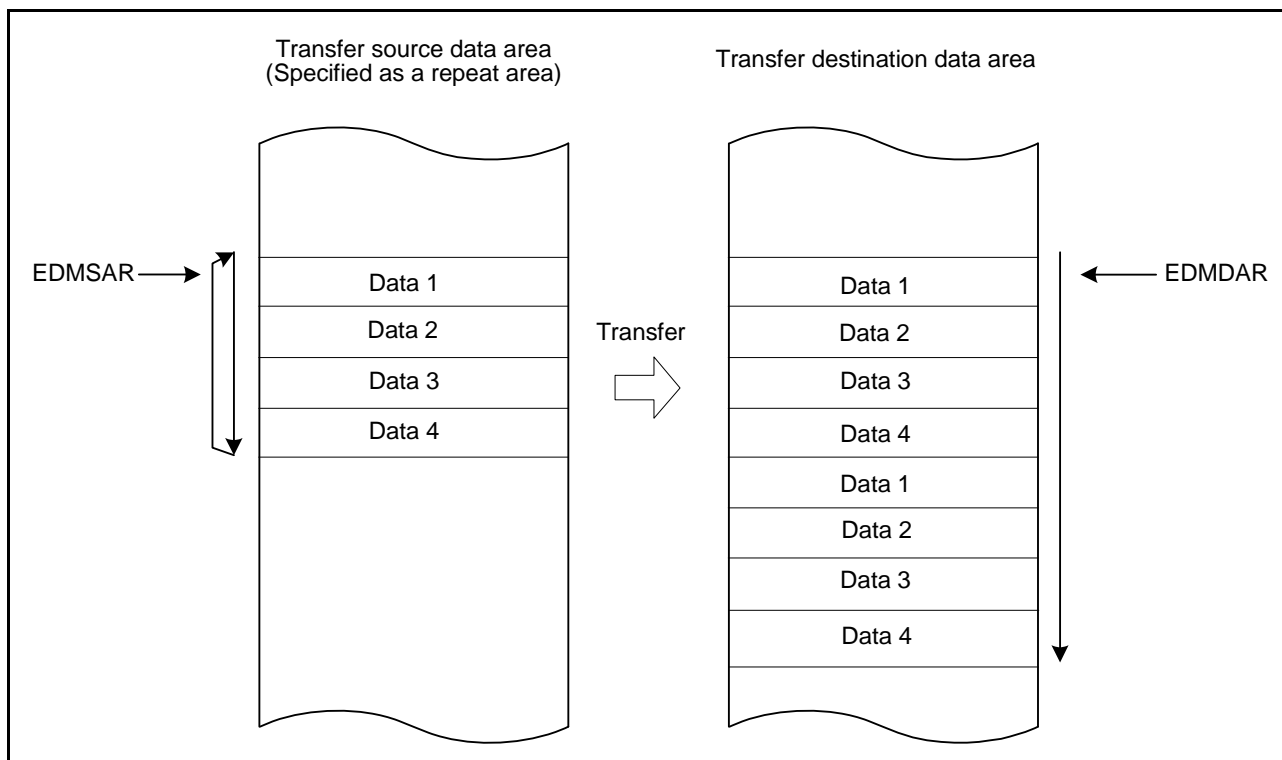


Figure 15.3 Operation in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request. A maximum of 1023 data can be set as a total block transfer size using EDMCRA of the EXDMACn.

A maximum of 1K can be set as the number of block transfer operations using EDMCRB of the EXDMACn; therefore, a maximum of 1023K data (1023 data × 1K blocks) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (EXDMACn.EDMSAR or EXDMACn.EDMDAR) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in EDMCNT of EXDMACn in the repeat size end interrupt handling.

Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.

Table 15.7 summarizes the register update operation in block transfer mode, and Figure 15.4 shows the operation in block transfer mode.

Table 15.7 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
EXDMACn.EDMSAR	Transfer source address	<ul style="list-style-type: none"> EXDMACn.EDMTMD.DTS[1:0] = 00 Increment/decrement/fixed/offset addition* EXDMACn.EDMTMD.DTS[1:0] = 01 Initial value of EXDMACn.EDMSAR EXDMACn.EDMTMD.DTS[1:0] = 10 Increment/decrement/fixed/offset addition*
EXDMACn.EDMDAR	Transfer destination address	<ul style="list-style-type: none"> EXDMACn.EDMTMD.DTS[1:0] = 00 Initial value of EXDMACn.EDMDAR EXDMACn.EDMTMD.DTS[1:0] = 01 Increment/decrement/fixed/offset addition* EXDMACn.EDMTMD.DTS[1:0] = 10 Increment/decrement/fixed/offset addition*
EXDMACn.EDMCRAH	Block size	Not updated
EXDMACn.EDMCRAL	Transfer count	EXDMACn.EDMCRAH
EXDMACn.EDMCRB	Block count	Decrement by one

Note: * Offset addition can be specified only for EXDMAC0.

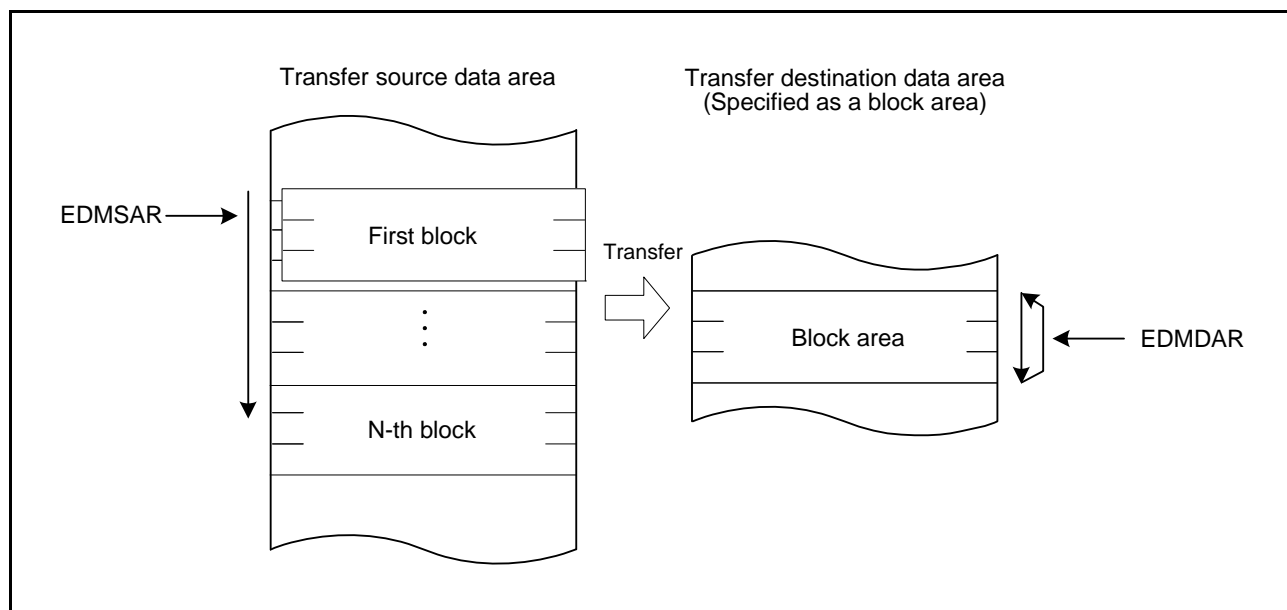


Figure 15.4 Operation in Block Transfer Mode

(4) Cluster Transfer Mode

In cluster transfer mode, a single cluster data is transferred by one transfer request. A maximum of 7 data can be set as a total cluster transfer size using EDMCRA of the EXDMACn.

A maximum of 1K can be set as the number of cluster transfer operations using EDMCRB of the EXDMACn; therefore, a maximum of 7K data (7 data × 1K) can be set as a total data transfer size.

The cluster transfer mode can be selected from among cluster transfer dual address mode, cluster transfer read address mode, and cluster transfer write address mode.

- Cluster transfer dual address mode

(EXDMACn.EDMTMD.MD[1:0] = 11, EXDMACn.EDMAMD.AMS = 0)

A single cluster data is transferred by one transfer request from the transfer source address to the cluster buffers. A single cluster data is then transferred from the cluster buffers to the transfer destination address.

- Cluster transfer read address mode

(EXDMACn.EDMTMD.MD[1:0] = 11, EXDMACn.EDMAMD.AMS = 1, EXDMACn.EDMAMD.DIR = 0)

A single cluster data is transferred by one transfer request from the transfer source address to the cluster buffers.

- Cluster transfer write address mode

(EXDMACn.EDMTMD.MD[1:0] = 11, EXDMACn.EDMAMD.AMS = 1, EXDMACn.EDMAMD.DIR = 1)

A single cluster data is transferred by one transfer request from the cluster buffers to the transfer destination address.

Table 15.8 summarizes the register update operation in cluster transfer mode, and Figure 15.5 shows the operation in cluster transfer mode.

Table 15.8 Register Update Operation in Cluster Transfer Mode (Dual Address Mode)

Register	Function	Update Operation after Completion of Single-Cluster Transfer by One Transfer Request
EXDMACn.EDMSAR	Transfer source address	<ul style="list-style-type: none"> EXDMACn.EDMTMD.DTS[1:0] = 00b Increment/decrement/fixd/offset addition* EXDMACn.EDMTMD.DTS[1:0] = 01b Initial value of EXDMACn.EDMSAR EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixd/offset addition*
EXDMACn.EDMDAR	Transfer destination address	<ul style="list-style-type: none"> EXDMACn.EDMTMD.DTS[1:0] = 00b Initial value of EXDMACn.EDMDAR EXDMACn.EDMTMD.DTS[1:0] = 01b Increment/decrement/fixd/offset addition* EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixd/offset addition*
EXDMACn.EDMCRAH	Cluster size	Not updated
EXDMACn.EDMCRAL	Transfer count	EXDMACn.EDMCRAH
EXDMACn.EDMCRB	Cluster count	Decrementd by one

Note: * Offset addition can be specified only for EXDMAC0.

In read address mode, the transfer destination address EXDMACn.EDMADAR is fixed (invalid).

In write address mode, the transfer destination address EXDMACn.EDMASAR is fixed (invalid).

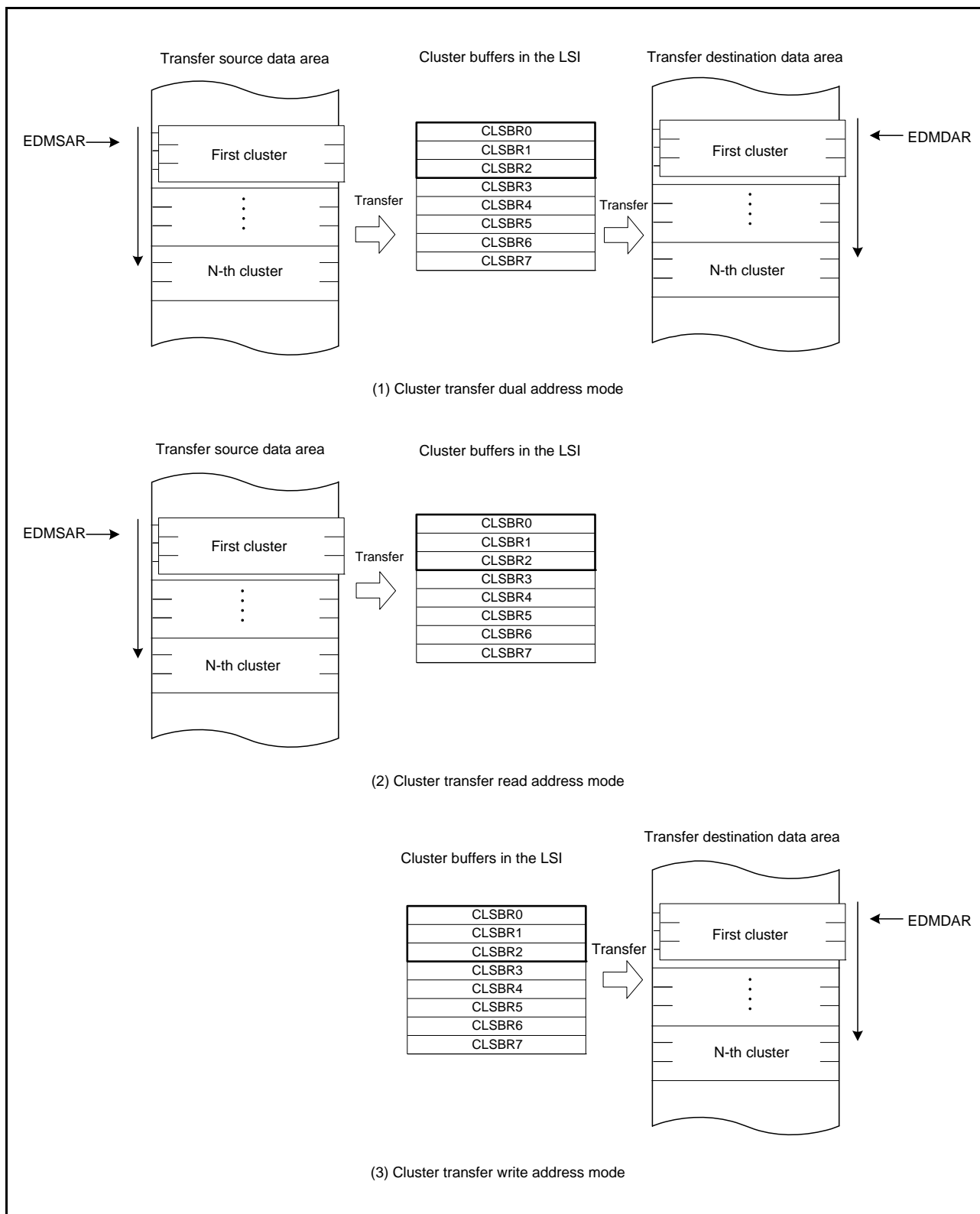


Figure 15.5 Operation in Cluster Transfer Mode

15.3.2 Extended Repeat Area Function

The EXDMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (EDMSAR) and transfer destination address register (EDMDAR) of EXDMACn.

The extended repeat area on the source address is specified by the SARA[4:0] bits in EDMAMD of EXDMACn. The extended repeat area on the destination address is specified by the DARA[4:0] bits in EDMAMD of EXDMACn. The size can be specified separately for the source and destination sides.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested to the CPU. When an overflow occurs in the extended repeat area on the transfer source while the SARIE bit in EDMINT of EXDMACn is set to 1, the ESIF flag in EDMSTS of EXDMACn is set to 1 and the DTE bit in EDMINT of EXDMACn is cleared to 0 to stop DMA transfer. At this time, if the ESIE bit in EDMINT of EXDMACn is set to 1, an interrupt by an extended repeat area overflow is requested to the CPU or DTC. When the DARIE bit in EDMINT of EXDMACn is set to 1, an overflow on the extended repeat area set in EDMDAR occurs, meaning that the destination side is a target. DMA transfer can be resumed by writing 1 to the DTE bit in EDMCNT of EXDMACn in the interrupt handling.

Figure 15.6 shows an example of the extended repeat area operation.

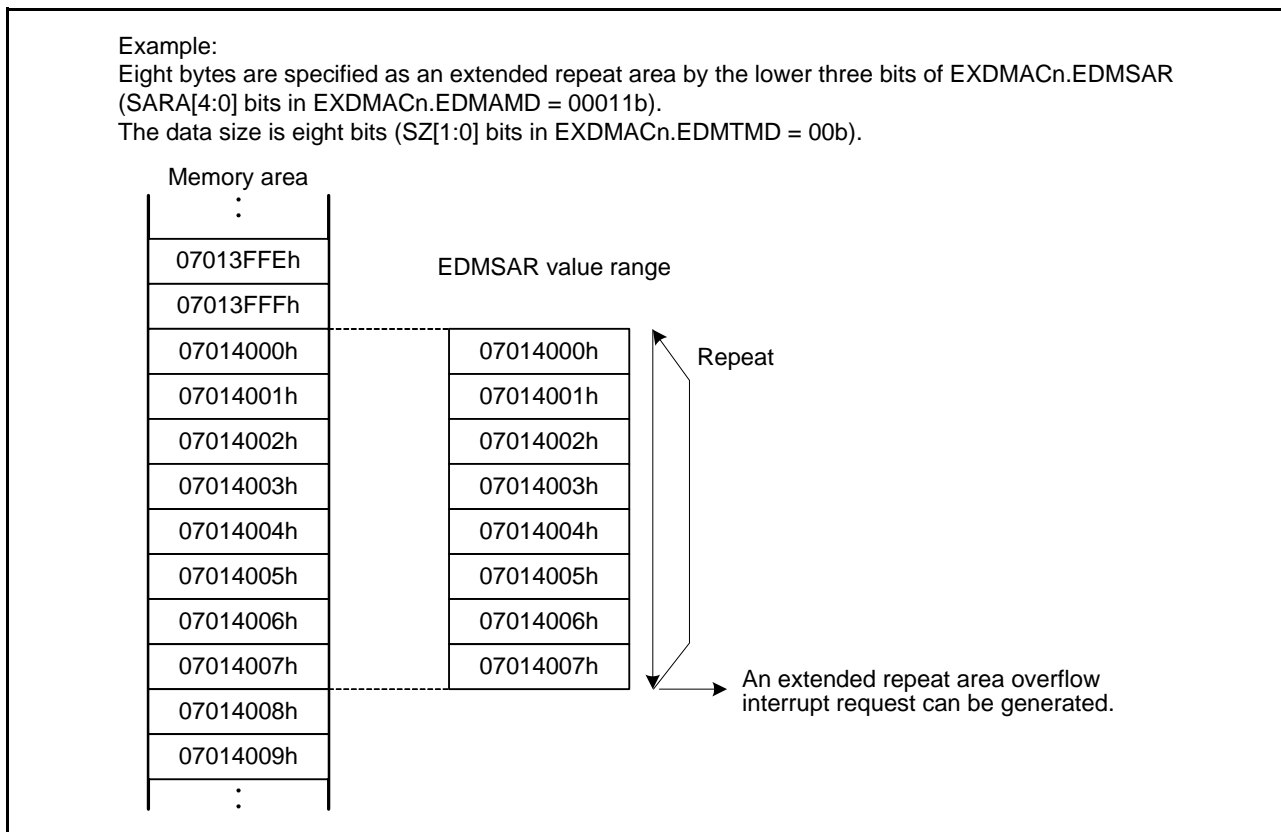


Figure 15.6 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode or cluster transfer mode, the following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size (or cluster size) is a power of 2 or the block size (or cluster size) boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block (or one cluster), the interrupt by the overflow is suspended until transfer of the block (or the cluster) is completed, and the transfer overruns.

Figure 15.7 shows an example when the extended repeat area function is used in block transfer mode.

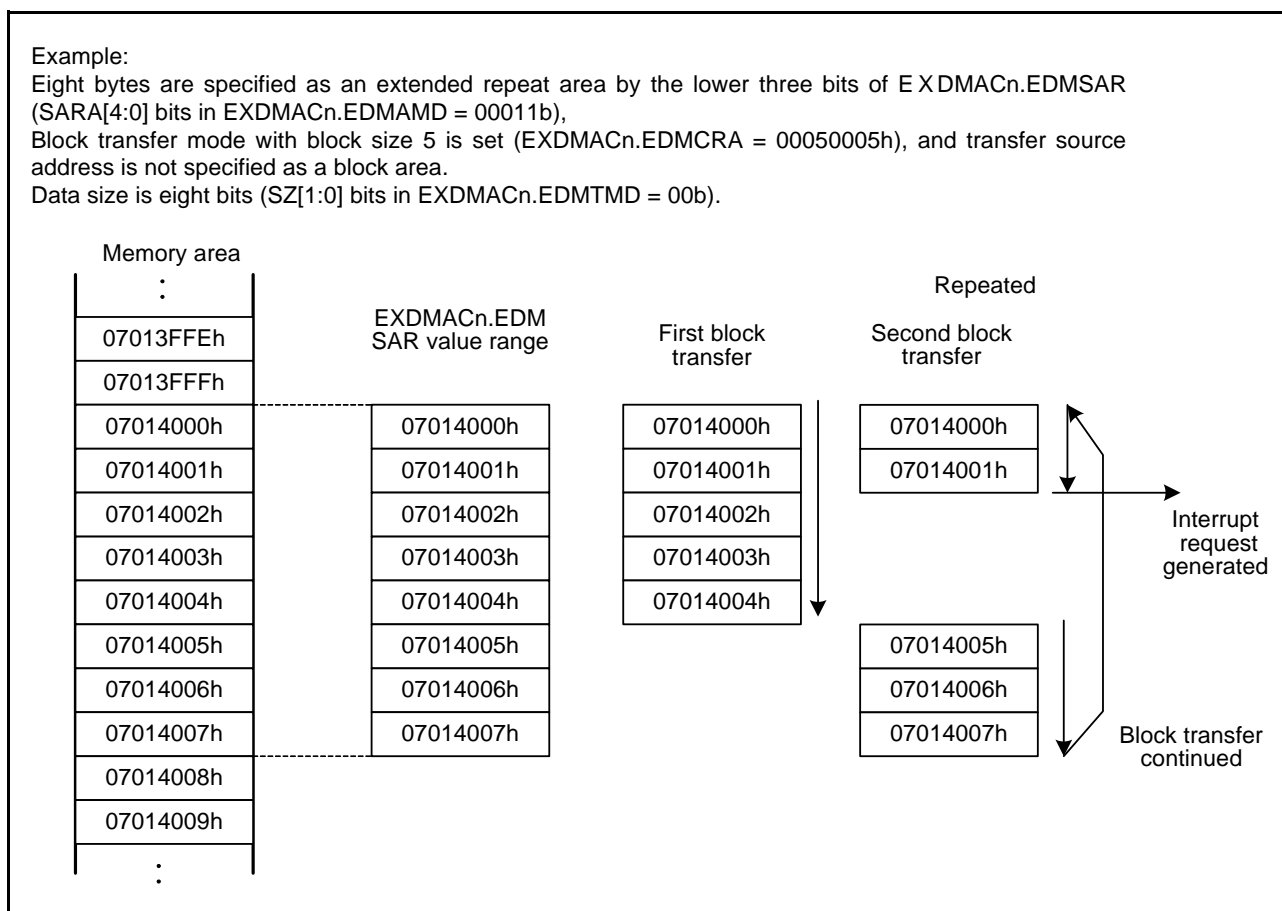


Figure 15.7 Example of Extended Repeat Area Function in Block Transfer Mode

15.3.3 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. When the offset addition is selected, the offset specified by the DMA offset register (EDMOFR of EXDMAC0) is added to the address every time the EXDMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in EDMOFR of EXDMAC0. In this case, the negative value must be 2's complement.

Address update function using offset can be specified only for the EXDMAC0 channel.

Table 15.9 shows the address update method in each address update mode.

Table 15.9 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of EXDMACn.EDMAMD.SM and EXDMACn.EDMAMD.DM for Address Update Modes	Address Update Method (for Different SZ[1:0] Settings in EDTMD of EXDMACn)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+EXDMAC0.EDMOFR*		
Increment	10b	+1	+2	+4
Decrement	11b	-1	-2	-4

Note: * When setting a negative value in the DMA offset register, the value must be 2's complement. The 2's complement is obtained by the following formula.

2's complement of a negative offset value = 1 + ~offset (~: bit inversion)

(1) Basic Transfer Using Offset Addition

Figure 15.8 shows an example of address updating using offset addition.

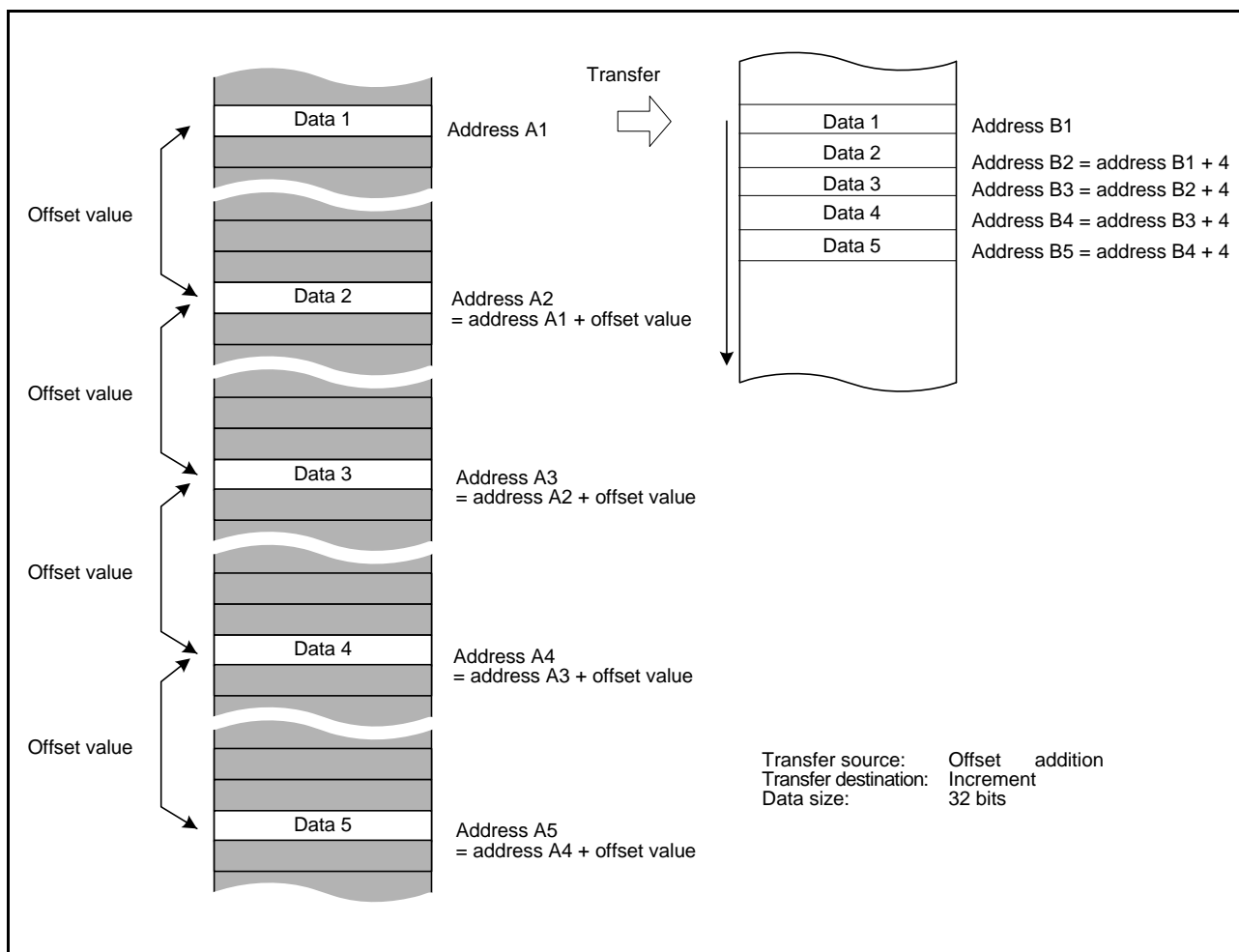


Figure 15.8 Example of Address Updating by Offset Addition

In Figure 15.8, the transfer data is 32 bits long, and offset addition and increment are set as the transfer source address update mode and transfer destination address update mode, respectively. The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

(2) Example of XY Conversion Using Offset Addition

Figure 15.9 shows the XY conversion using offset addition in repeat transfer mode.

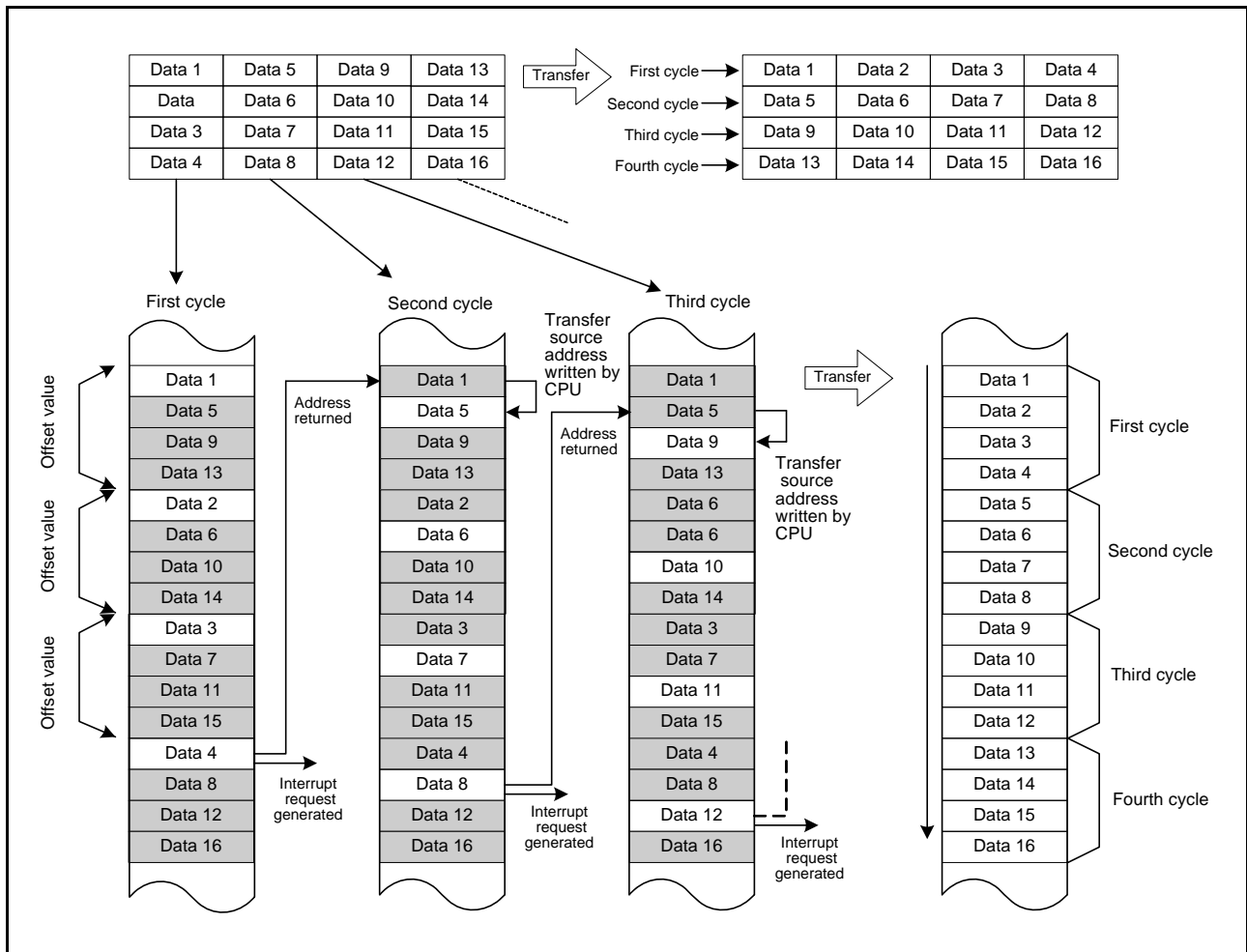


Figure 15.9 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

In Figure 15.9, the source address side is specified as the repeat area by EXDMAC0.EDMAMD and the offset addition is selected. The offset value is set to $4 \times$ transfer data size (when the transfer data size is 32 bits, H'00000010 is set in EXDMAC0.EDMOFR, as an example). The repeat size is set to $4 \times$ transfer data size (when transfer data size is 32 bits, the repeat size is set to $4 \times 4 = 16$ bytes, as an example). Increment is specified for the transfer destination address.

A repeat size end interrupt is requested when the RPTIE bit in EDMINT of EXDMAC0 is set to 1 and transfer of the repeat size of data is completed.

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, the transfer source address returns to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, write the address of data 5 to EDMSAR of EXDMAC0 using the CPU (when the data access size is 32 bits, write the data 1 address + 4). When the DTE bit in EDMCNT of EXDMAC0 is set to 1, the transfer is resumed from the state when the transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 15.10 shows a flowchart of the XY conversion.

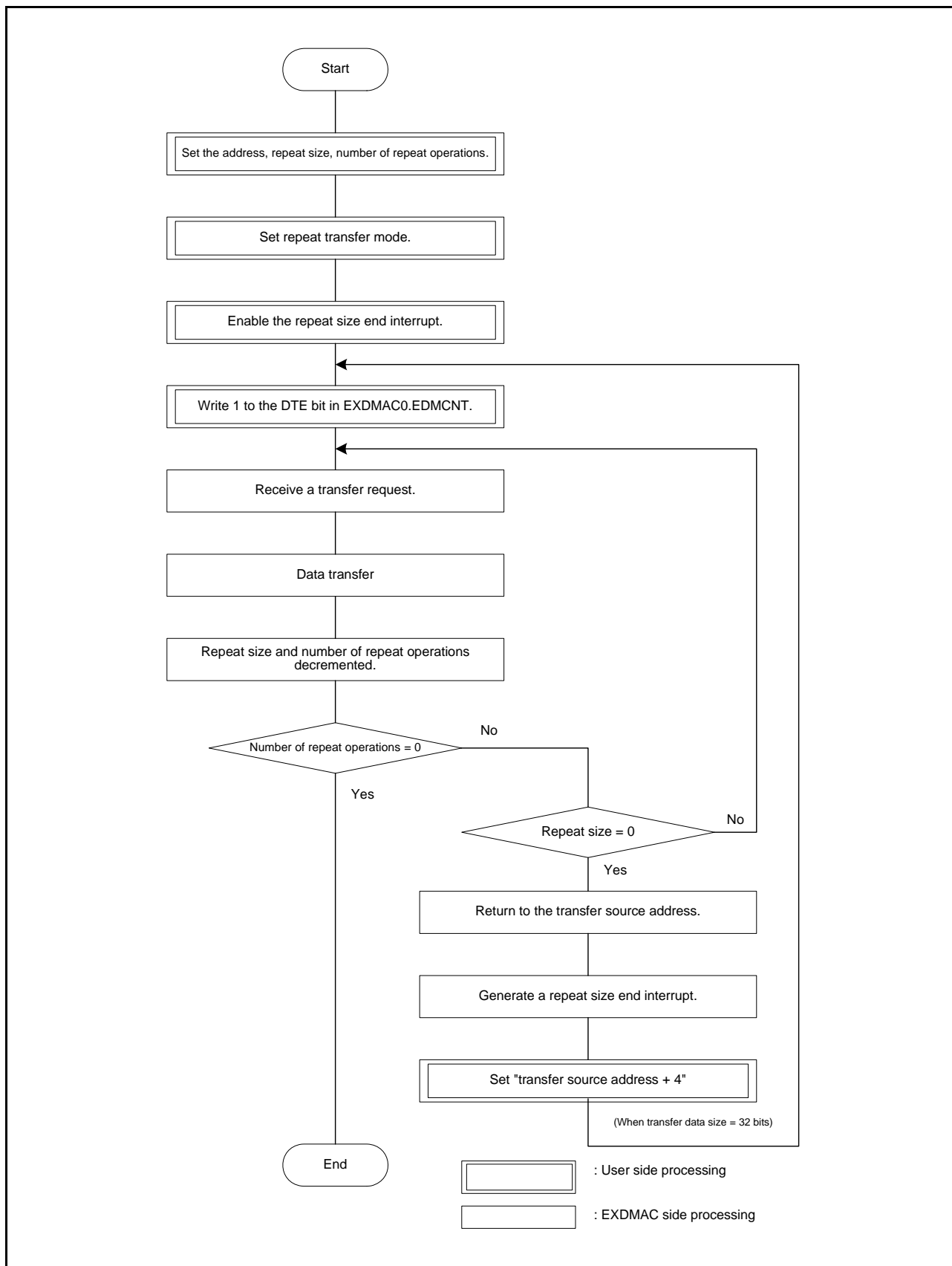


Figure 15.10 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

15.3.4 Address Modes

The EXDMAC provides dual and single address modes (dual, read, and write address modes in cluster transfer), either of which can be selectable. Table 15.10 shows the relationship between transfer modes and address modes.

Table 15.10 Relationship between Transfer Modes and Address Modes

Transfer Mode	Address Mode	Single Address Direction	EXDMAC Operation
Normal transfer mode (EDMTMD.MD[1:0] = 00)	Dual address mode (EDMAMD.AMS = 0)	—	Reads and then writes data.
	Single address mode (EDMAMD.AMS = 1)	Transfer source (EDMAMD.DIR = 0)	Only reads data; outputs EDACK to the device to be written to. The RX62N/RX621 receives no read data.
		Transfer destination (EDMAMD.DIR = 1)	Only writes data; outputs EDACK to the device to be read from. The RX62N/RX621 outputs no write data.
Repeat transfer mode (EDMTMD.MD[1:0] = 01)	Dual address mode (EDMAMD.AMS = 0)	—	Reads and then writes data.
	Single address mode (EDMAMD.AMS = 1)	Transfer source (EDMAMD.DIR = 0)	Only reads data; outputs EDACK to the device to be written to. The RX62N/RX621 receives no read data.
		Transfer destination (EDMAMD.DIR = 1)	Only writes data; outputs EDACK to the device to be read from. The RX62N/RX621 outputs no write data.
Block transfer mode (EDMTMD.MD[1:0] = 10)	Dual address mode (EDMAMD.AMS = 0)	—	Reads and writes data alternately for every data transfer specified by EDMTMD.SZ (transfer data size)
	Single address mode (EDMAMD.AMS = 1)	Transfer source (EDMAMD.DIR = 0)	Only reads data; outputs EDACK to the device to be written to. The RX62N/RX621 receives no read data.
		Transfer destination (EDMAMD.DIR = 1)	Only writes data; outputs EDACK to the device to be read from. The RX62N/RX621 outputs no write data.
Cluster transfer mode (EDMTMD.MD[1:0] = 11)	Dual address mode (EDMAMD.AMS = 0)	—	Reads data of cluster size and then writes data of cluster size.
	Read address mode (EDMAMD.AMS = 1)	Transfer source (EDMAMD.DIR = 0)	Only reads data of cluster size. Transfers data to the cluster buffers.
	Write address mode (EDMAMD.AMS = 1)	Transfer destination (EDMAMD.DIR = 1)	Only writes data of cluster size. Transfers data from the cluster buffers.

15.4 Transfer Operation

The operation of EXDMAC transfer is explained below. The EXDMAC operates in synchronization with the external bus clock (BCLK). Unless other specifications are made, the following is an example of operation when the external bus clock (BCLK) has the same frequency as the output on the BCLK pin.

15.4.1 Normal/Repeat Transfer Operation

(1) Dual Address Mode

Figure 15.11 shows the bus cycle example in normal-transfer dual address mode. In the example, a 16-bit data (SZ[1:0] = 01 in EDMTMD of EXDMACn) is transferred from a device with 16-bit 2-cycle access to another device with 16-bit 2-cycle access, which is started at the falling edge of EDREQ.

The bus cycles in repeat-transfer dual address mode are the same as those in normal-transfer dual address mode.

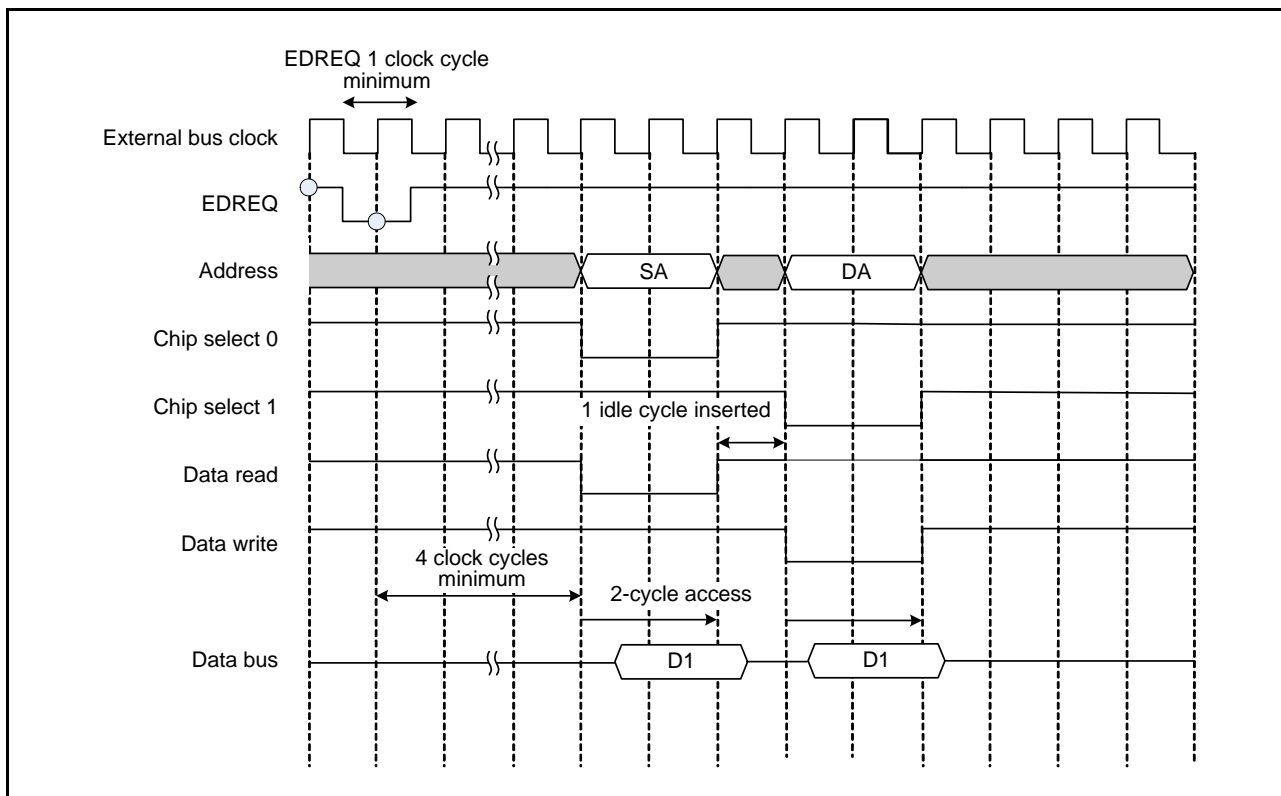


Figure 15.11 Bus Cycle Example in Normal-Transfer Dual Address Mode

(2) Single Address Mode

In single address mode, data is read from the transfer source address and directly transferred to the transfer destination device without being taken in the LSI. Here, EDACK is output to one of the external transfer-destination and transfer-source devices, and the address is simultaneously output to the other transfer device for access.

With the DIR bit in EDMAMD of EXDMACn set to 0, the transfer source address is output to the external bus and EDACK is output to the transfer destination. With the DIR bit in EDMAMD of EXDMACn set to 1, the transfer destination address is output to the external bus and the EDACK is output to the transfer source. Figure 15.12 shows the data flow in single address mode.

Figure 15.13 shows the bus cycle example in normal-transfer single address mode. In the example, one data is transferred in 2-cycle access when the DIR bit in EDMAMD of EXDMACn is set to 1 (transfer destination address is output) and when set to 0 (transfer source address is output).

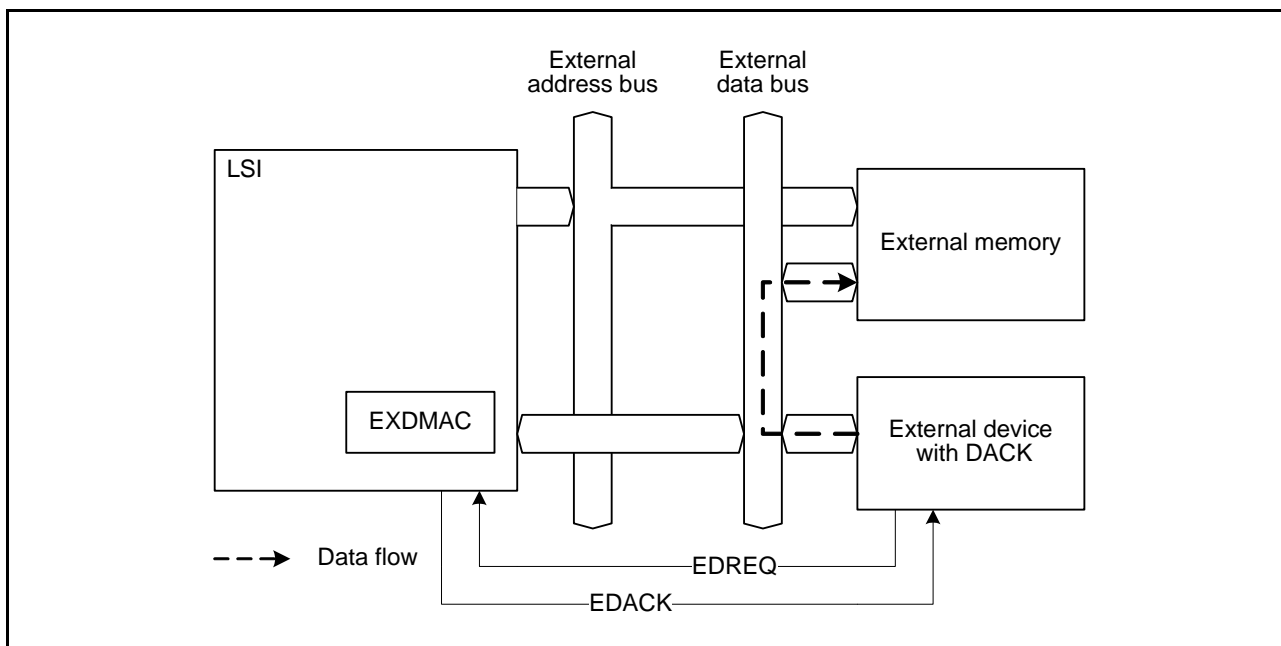


Figure 15.12 Data Flow in Single Address Mode (when DIR = 1)

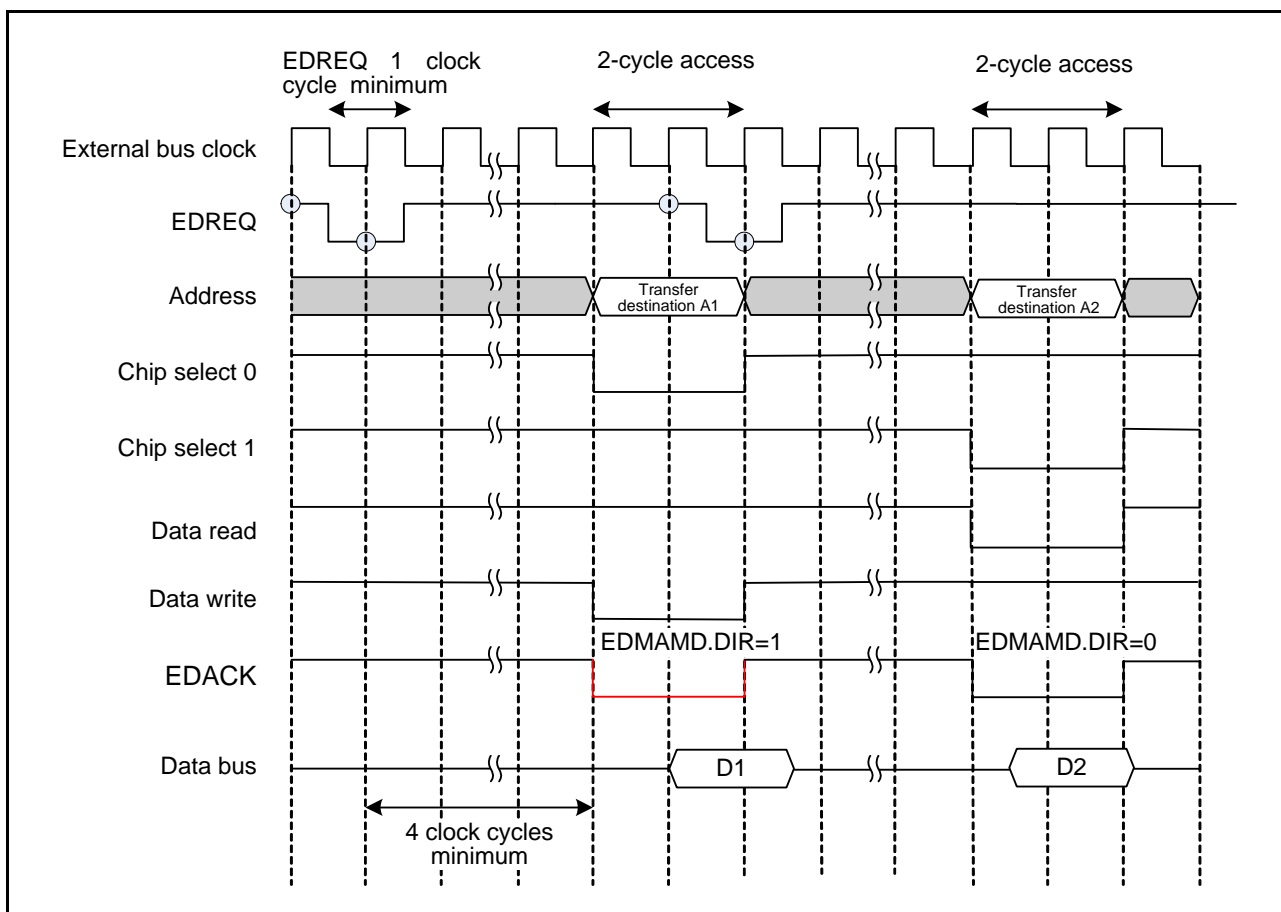


Figure 15.13 Bus Cycle Example in Normal-Transfer Single Address Mode

15.4.2 Block Transfer Operation

(1) Dual Address Mode

Figure 15.14 shows the bus cycle example in block-transfer dual address mode. In the example, a 16-bit data (SZ[1:0] = 01 in EDMTMD of EXDMACn) is transferred from a device with 16-bit 2-cycle access to another device with 16-bit 2-cycle access when the block size is 3.

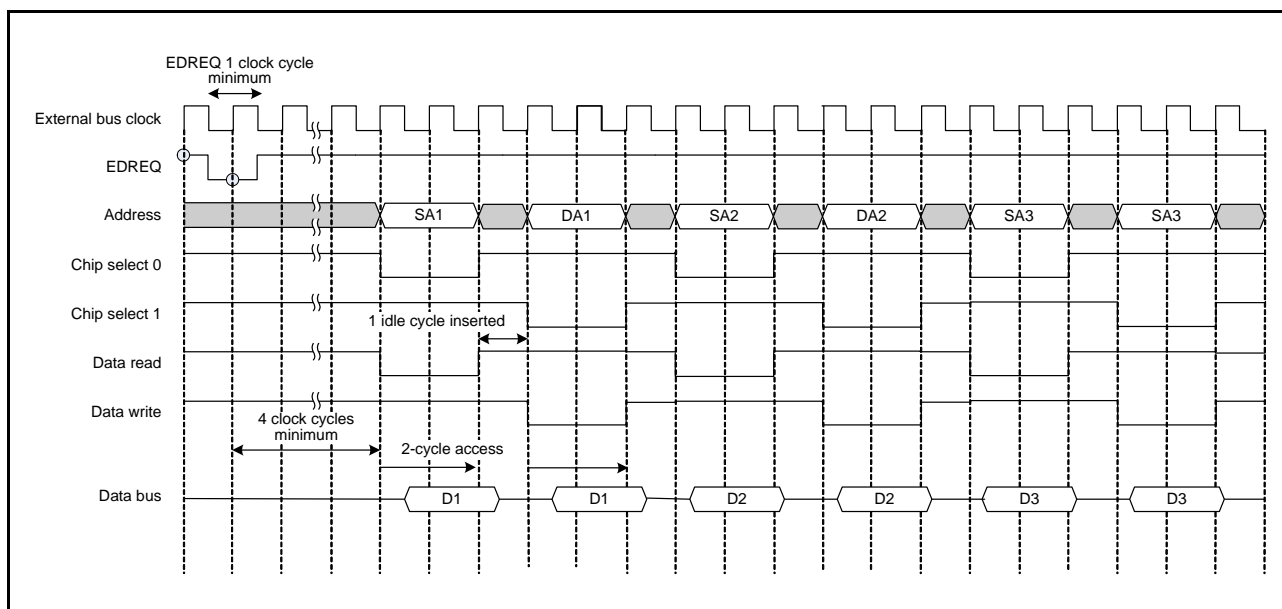


Figure 15.14 Bus Cycle Example in Block-Transfer Dual Address Mode

(2) Single Address Mode

Figure 15.15 shows the bus cycle example in block-transfer single address mode. In the example, a 16-bit data (SZ[1:0] = 01 in EDMTMD of EXDMACn) is transferred in three bus clock cycles from a device with EDACK with 16-bit access to another device with 16-bit access when the block size is 3.

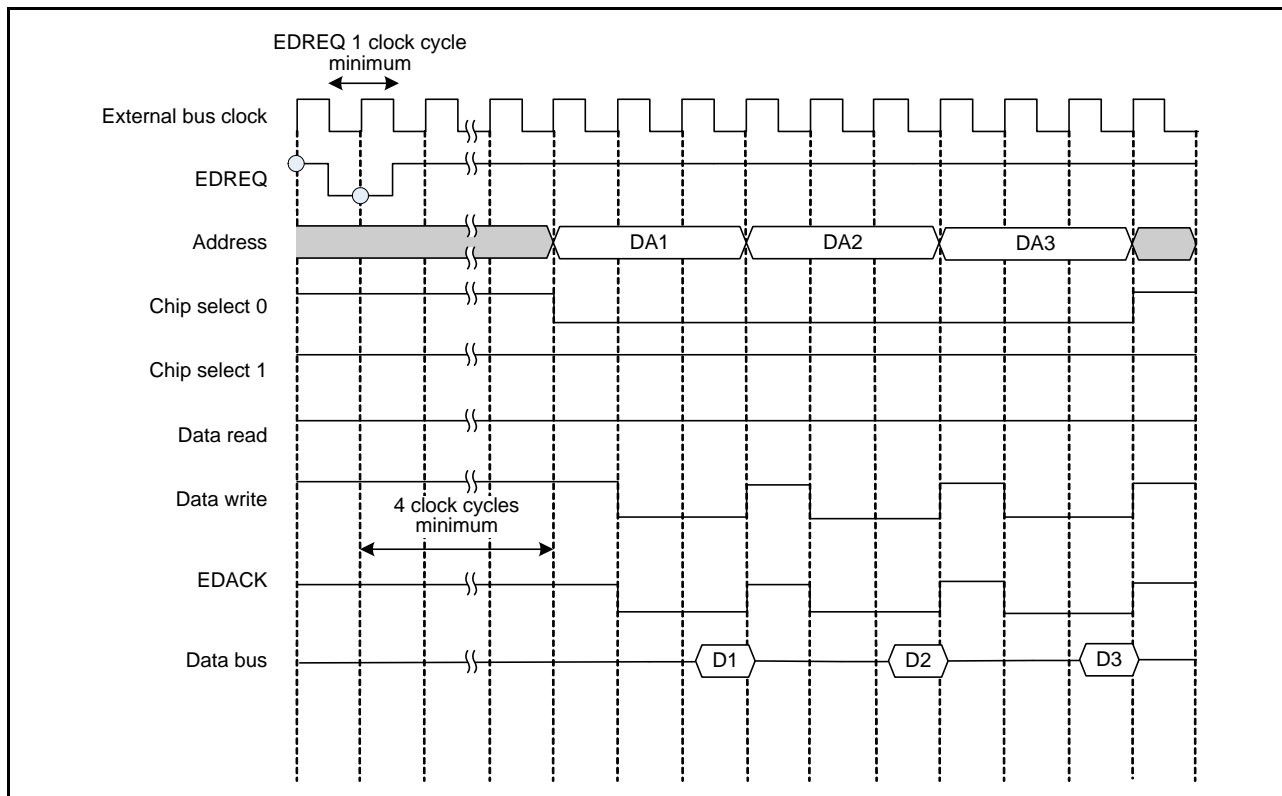


Figure 15.15 Data Flow in Block Transfer Single Address Mode

15.4.3 Cluster Transfer Operation

(1) Dual Address Mode

In cluster-transfer dual address mode, cluster-size data is transferred from the external transfer source device to the external transfer destination device via the cluster buffers. Figure 15.16 shows the data flow in cluster-transfer dual address mode and Figure 15.17 shows the bus cycle example, in which one cluster is transferred in two clock cycles when the cluster size is three.

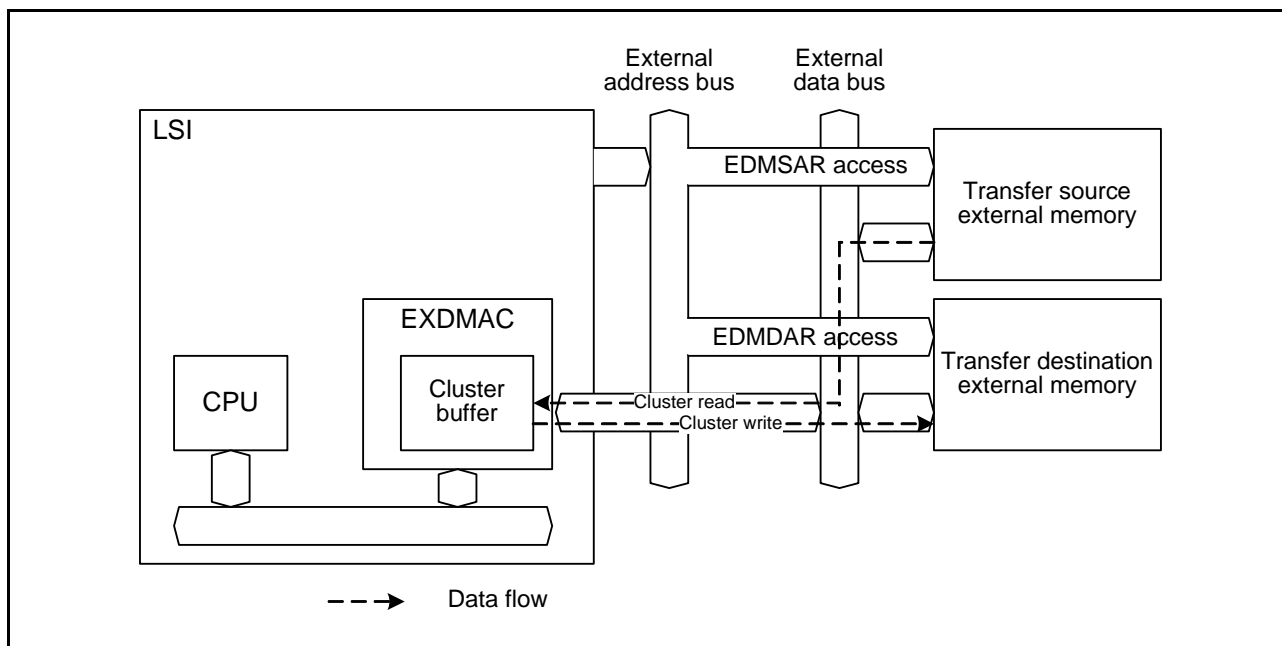


Figure 15.16 Data Flow in Cluster-Transfer Dual Address Mode

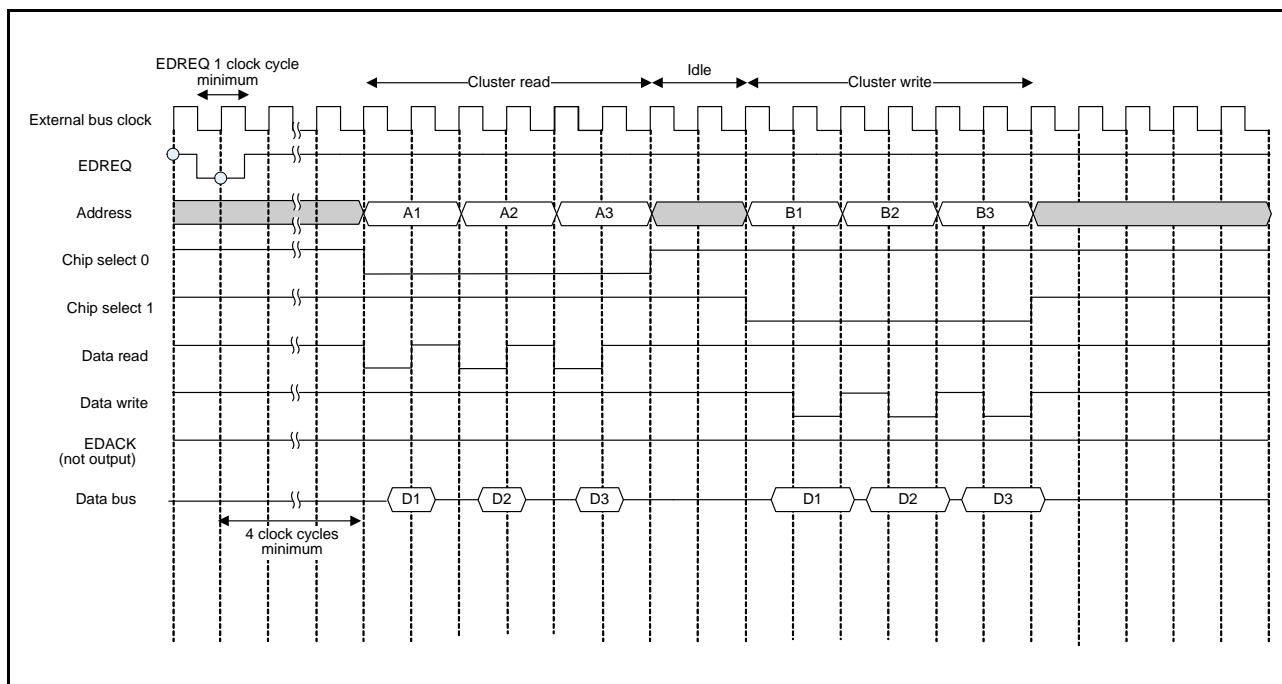


Figure 15.17 Bus Cycle Example in Cluster-Transfer Dual Address Mode

(2) Read Address Mode

In cluster-transfer read address mode, cluster-size data is transferred from the external transfer source device to the cluster buffers. The data transferred in the cluster buffers can be read by the CPU. Figure 15.18 shows the data flow in cluster-transfer read address mode and Figure 15.19 shows the bus cycle example, in which one cluster is transferred in two clock cycles when the cluster size is six.

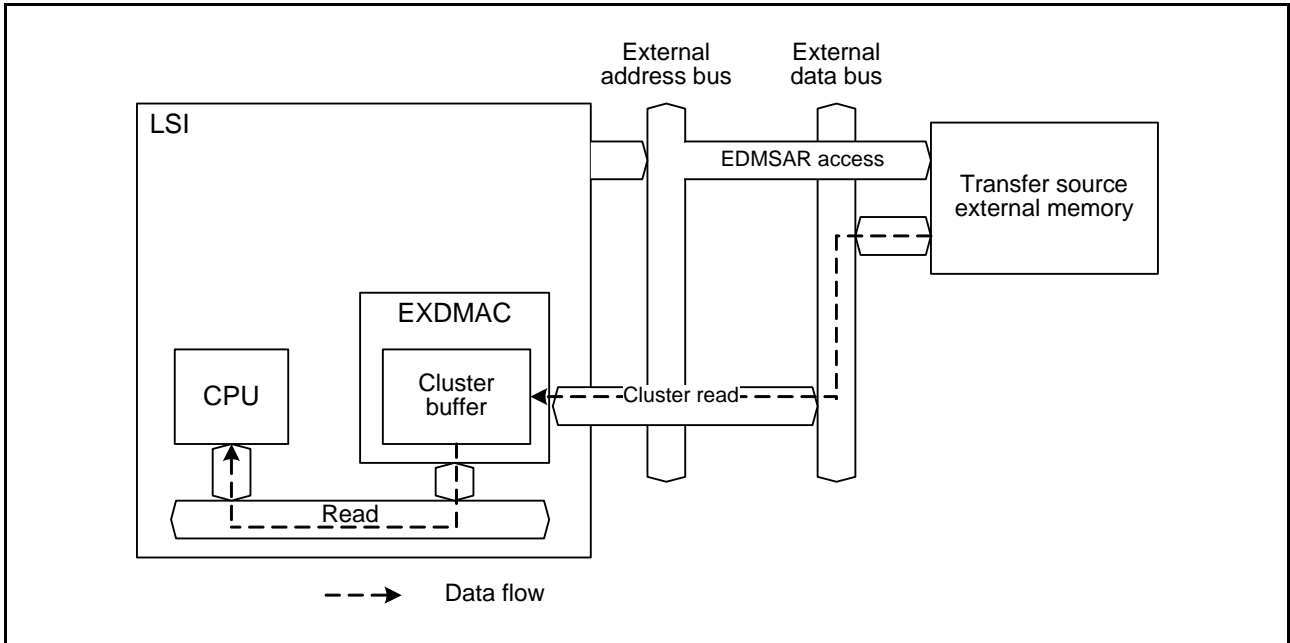


Figure 15.18 Data Flow in Cluster-Transfer Read Address Mode

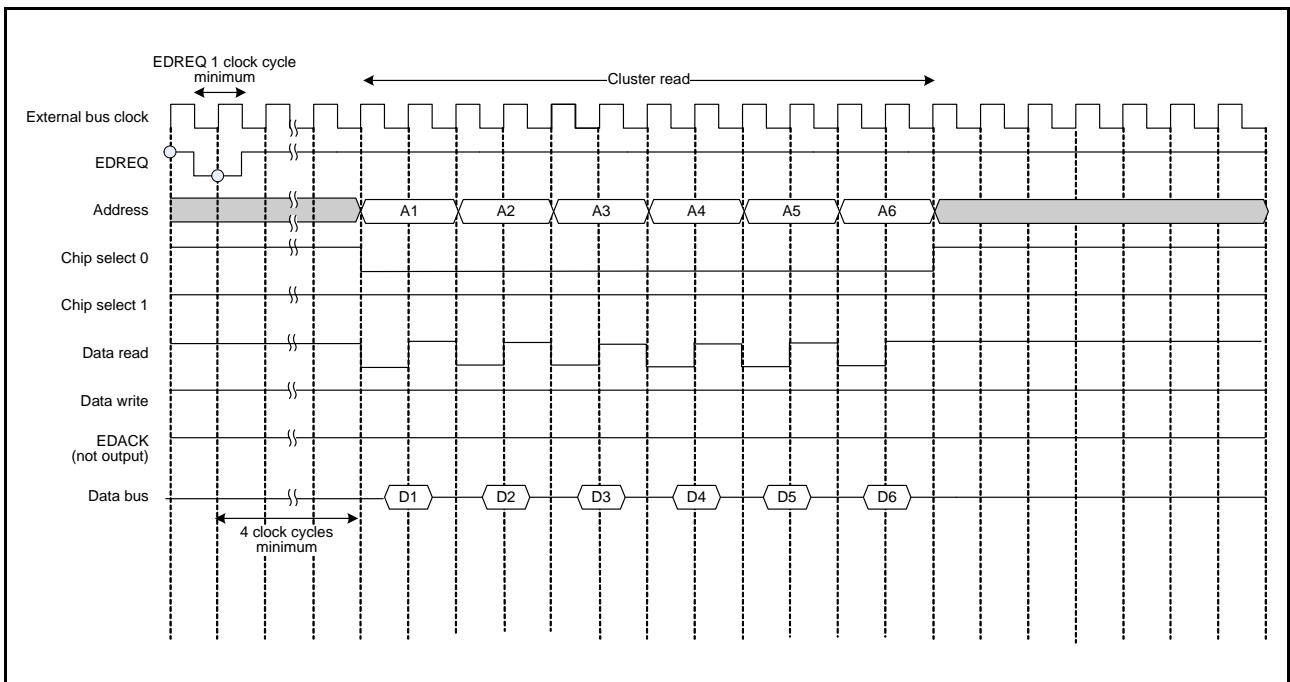


Figure 15.19 Bus Cycle Example in Cluster-Transfer Read Address Mode

(3) Write Address Mode

In cluster-transfer write address mode, the data is written to the cluster buffers by the internal bus master such as the CPU, DMACA, and DTC and then transferred to the external transfer destination device. Figure 15.20 shows the data flow in cluster-transfer write address mode and Figure 15.21 shows the bus cycle example, in which one cluster is transferred in two clock cycles when the cluster size is six.

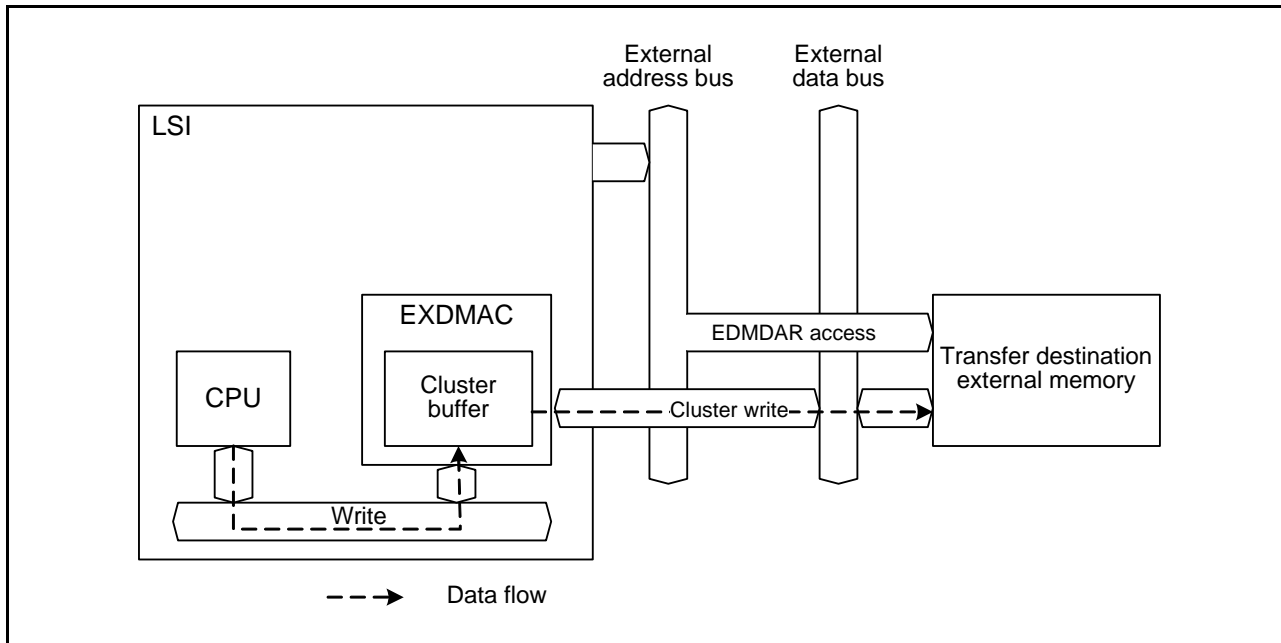


Figure 15.20 Data Flow in Cluster-Transfer Write Address Mode

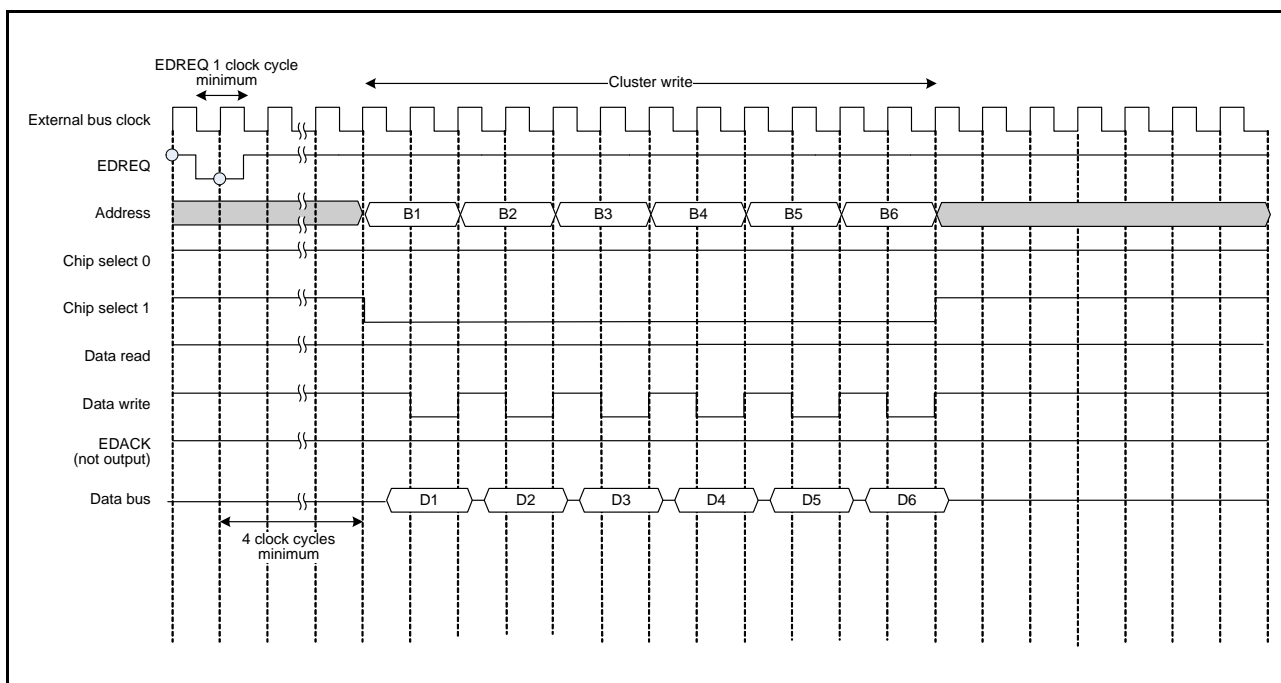


Figure 15.21 Bus Cycle Example in Cluster-Transfer Write Address Mode

15.5 Activation Sources and Procedures for Activation

15.5.1 Activation Sources

The EXDMAC can be activated by software, an external DMA transfer request pin (EDREQ pin), or DMA transfer requests from the on-chip peripheral modules (compare match of MTU1). Setting the DCTG[1:0] bits in EDMTMD of EXDMACn selects the activation source.

(1) EXDMAC Activation by Software

Setting the EXDMACn.EDMREQ.DCTG[1:0] bits to 00b enables the EXDMAC activation by software. To start DMA transfer by software, follow the procedure below.

1. Check that the EXDMACn.EDMREQ.SWREQ bit is 0.
2. Set the EXDMACn.EDMTMD.DCTG[1:0] bits to 00b.
3. Set the EXDMACn.EDMCNT.DTE bit to 1.
4. Set the EXDMACn.EDMREQ.SWREQ bit to 1.

When the EXDMACn.DMREQ.CLRS bit is 0, the EXDMACn.EDMREQ.SWREQ bit is cleared to 0 after data transfer is started in response to a DMA transfer request. When the CLRS bit is 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) Activation by an External DMA Transfer Request Pin (EDREQ)

Setting the EXDMACn.EDMTMD.DCTG[1:0] bits to 10b enables the activation by an external DMA transfer request pin. To set the activation by an external DMA transfer request pin, follow the procedure below.

1. Set the detection mode by the EXDMACn.EDMRMD.DREQS[1:0] bits.
2. Set the EXDMACn.EDMTMD.DCTG[1:0] bits to 10b (external DMA transfer request pin).
3. Set the EXDMACn.EDMERF.EREQ flag to 1 to clear the flag.
4. Set the EXDMACn.EDMCNT.DTE bit to 1 (activation enabled).

When the falling edge or rising edge is selected by using the EXDMACn.EDMRMD.DREQS[1:0] bits, if the external DMA transfer request pin detects an edge, the EXDMACn.EDMERF.EREQ flag is set to 1. The EXDMACn.EDMERF.EREQ flag is cleared to 0 when the DMA transfer is started by the external request. Moreover, this flag is cleared to 0 by writing 1 to it.

When the low level detection is set by the EXDMACn.EDMRMD.DREQS[1:0] bits, if the external DMA transfer request pin is low level, the EXDMACn.EDMERF.EREQ flag is set to 1. If the external DMA transfer request pin is high level, the EXDMACn.EDMERF.EREQ flag is 0. In case of the low level detection, when the DMA transfer is started by the external request or 1 is written to the flag, the EXDMACn.EDMERF.EREQ flag is not cleared to 0.

When the EDMAST.DMST bit and the EXDMACn.EDMCNT.DTE bit are set to 1 while the EXDMACn.EDMERF.EREQ flag is 1, the DMA transfer is started.

The value of the EXDMACn.EDMERF.EREQ flag is retained regardless of the settings in the EDMAST.DMST and EXDMACn.EDMCNT.DTE bits.

Figure 15.22 and Figure 15.23 show the timing of an external DMA transfer request by falling-edge detection and by low-level detection, respectively.

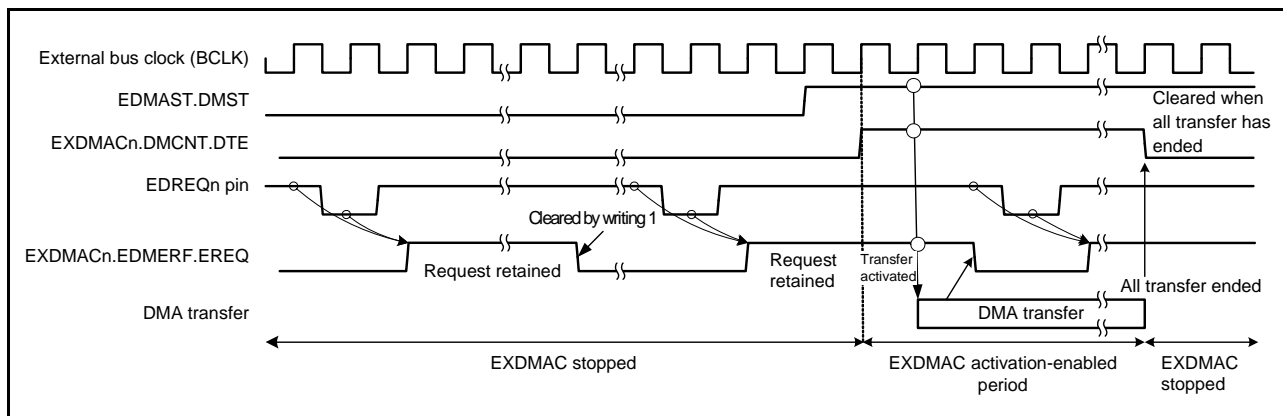


Figure 15.22 Timing of an External DMA Transfer Request by Falling-Edge Detection

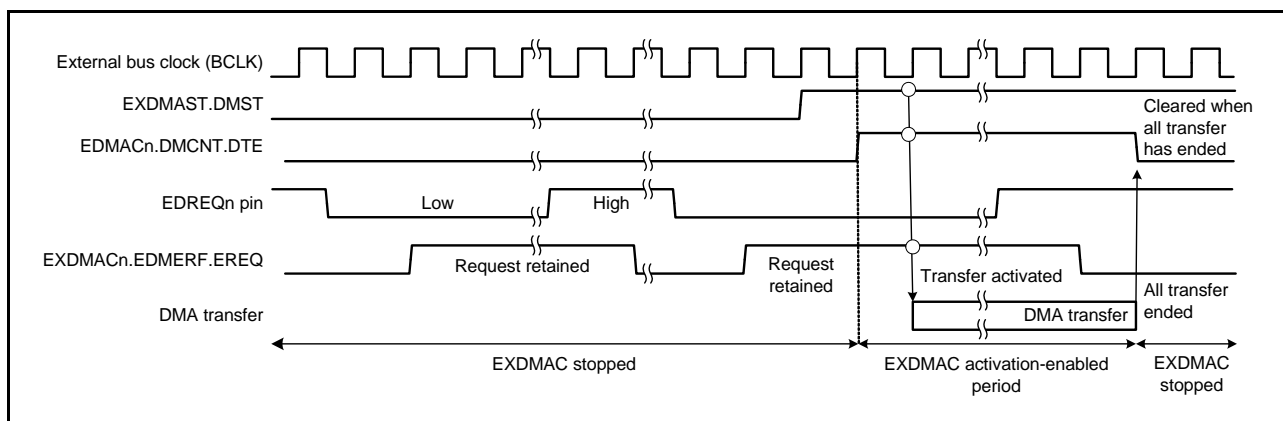


Figure 15.23 Timing of an External DMA Transfer Request by Low-Level Detection

(3) EXDMAC Activation by DMA Transfer Requests from On-Chip Peripheral Modules (MTU1 Compare Match)
 Setting the DCTG[1:0] bits in EDMTMD of EXDMACn to 11b enables the EXDMAC activation by DMA requests from the on-chip peripheral modules (compare match of MTU1). To start DMA transfer by DMA requests from the on-chip peripheral modules (compare match of MTU1), follow the procedure below.

1. Set the EXDMACn.EDMTMD.DCTG[1:0]bits to 11b (compare match of MTU1).
2. Set the EXDMACn.EDMPRF.PREQ flag to 1 to clear the flag.
3. Set the EXDMACn.EDMCNT.DTE bit to 1 (activation enable).

When a DMA transfer request is input from the on-chip peripheral modules (compare match of MTU1), the EXDMACn.EDMPRF.PREQ flag in EXDMACn is set to 1. The EXDMACn.EDMPRF.PREQ flag is cleared to 0 when the DMA transfer is started by the on-chip peripheral module request. This flag is cleared to 0 by writing 1 to it.

When the EDMAST.DMST bit is set to 1 while the EXDMACn.EDMPRF.PREQ flag is 1, the DMA transfer is started.

The value of the EXDMACn.EDMPRF.PREQ flag is retained regardless of the settings in the EXDMACn.EDMCNT.DTE and EXDMACn.EDMPRF.PREQ bits.

15.5.2 Activating the EXDMAC

Figure 15.24 shows the register setting procedure.

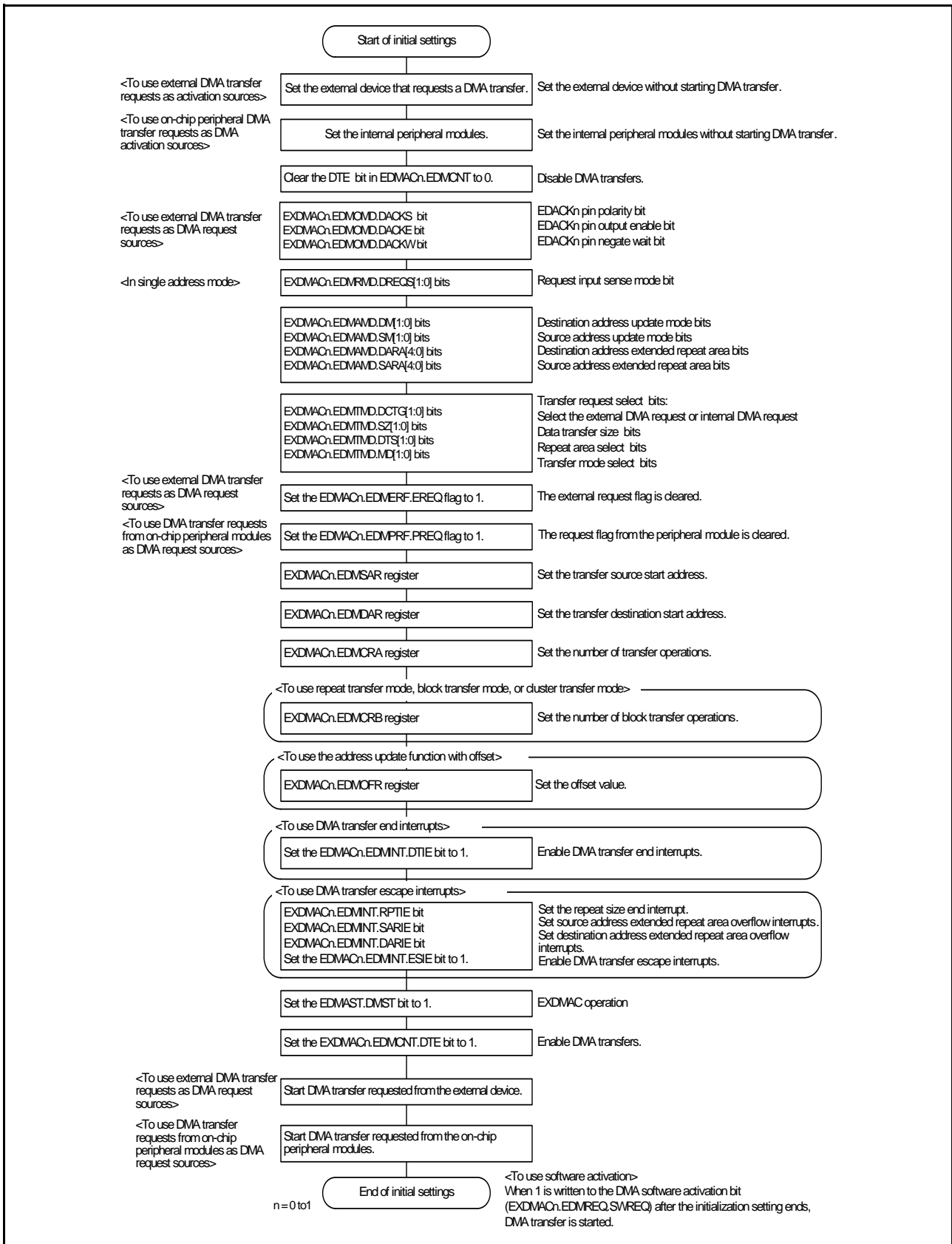


Figure 15.24 Register Setting Procedure

15.5.3 Starting DMA Transfer

Setting the DTE bit in EDMCNT of EXDMACn to 1 (DMA transfer enabled) and setting the DMST bit in EDMAST to 1 (EXDMAC start) enable DMA transfer of channel n (n = 0, 1).

When DMA transfer requests are generated, channel arbitration is performed where a DMA transfer request of higher-priority channel is accepted and DMA transfer of the channel starts. When a DMA transfer request is accepted and DMA transfer starts, the ACT flag in EDMSTS of EXDMACn is set to 1 (DMA transfer is in progress).

15.5.4 Registers during DMA Transfer

The EXDMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are EDMSAR, EDMDAR, EDMCRA, EDMCRB, EDMCNT, and EDMSTS of EXDMACn.

(1) DMA Source Address Register (EXDMACn.EDMSAR)

When data has been transferred in response to one transfer request, the contents of EDMSAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, refer to Table 15.5 to Table 15.8.

(2) DMA Destination Address Register (EXDMACn.EDMDAR)

When data has been transferred in response to one transfer request, the contents of EDMDAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, refer to Table 15.5 to Table 15.8.

(3) DMA Transfer Count Register (EXDMACn.EDMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, refer to Table 15.5 to Table 15.8.

(4) DMA Block Transfer Count Register (EXDMACn.EDMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, refer to Table 15.5 to Table 15.8.

(5) DMA Transfer Enable Bit (EXDMACn.EDMCNT.DTE)

Although the DTE bit in EDMCNT of EXDMACn enables or disables data transfer by the CPU write access, it is automatically cleared to 0 by the EXDMAC according to the DMA transfer state.

The conditions for clearing this bit by the EXDMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt

(6) DMA Active Flag (EXDMACn.EDMSTS.ACT)

The ACT flag in EDMSTS of EXDMAC indicates whether the EXDMAC is in the idle or active state.

This flag is set to 1 when the EXDMAC starts data transfer, and is cleared to 0 when data transfer in response to one transfer request is completed.

Even when DMA transfer is stopped by writing 0 to the DTE bit in EDMCNT of EXDMACn, this flag remains 1 until DMA transfer is completed.

(7) Transfer End Interrupt Flag (EXDMACn.EDMSTS.DTIF)

The DTIF bit in EDMSTS of EXDMACn is set to 1 after transfer of the total transfer size of data is completed.

When both this flag and the DTIE bit in EDMINT of EXDMACn are set to 1, a transfer end interrupt is requested.

This flag is set to 1 when the DMA transfer bus cycle is completed and the ACT flag in EDMSTS of EXDMACn is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DTE bit in EDMCNT of EXDMACn is set to 1 during the interrupt handling.

(8) Transfer Escape End Interrupt Flag (EXDMACn.EDMSTS.ESIF)

The ESIF flag in EDMSTS of EXDMACn is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this flag and the ESIE bit in EDMINT of EXDMACn are set to 1, a transfer escape end interrupt is requested.

This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the ACT flag in EDMSTS of EXDMACn is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DTE bit in EDMCNT of EXDMACn is set to 1 during an interrupt handling.

Before sending an interrupt request from the EXDMACn to the CPU or the DTC, the interrupt control register must be set.

For details, see section 11, Interrupt Control Unit (ICUa).

15.5.5 Channel Priority

When multiple DMA transfer requests are present, the EXDMAC determines the priority of channels that have DMA transfer requests. The channel priority is fixed as channel 0 > channel 1.

When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer transfer of the higher-priority channel starts.

15.6 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DTE bit in EDMCNT and the ACT flag in EDMSTS of EXDMACn are changed from 1 to 0, indicating that DMA transfer has ended.

15.6.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (EXDMACn.EDMTMD.MD[1:0] = 00b)

When the value of the EXDMACn.EDMCRAL register changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in EDMCNT of EXDMACn is cleared to 0 and the DTIF flag in EDMSTS of EXDMACn is set to 1 at the same time. If the DTIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (EXDMACn.EDMTMD.MD[1:0] = 01b)

When the value of DMCRB of EXDMACn changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in EDMCNT of EXDMACn is cleared to 0 and the DTIF flag in EDMSTS of EXDMACn is set to 1 at the same time. If the DTIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the EXDMACn to the CPU or the DTC, the interrupt control register must be set. For details, see section 11, Interrupt Control Unit (ICUa).

(3) In Block Transfer Mode (EXDMACn.EDMTMD.MD[1:0] = 10b)

When the value of EDMCRB of EXDMACn changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in EDMCNT of EXDMACn is cleared to 0 and the DTIF flag in EDMSTS of EXDMACn is set to 1 at the same time. If the DTIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the EXDMACn to the CPU or the DTC, the interrupt control register must be set. For details, see section 11, Interrupt Control Unit (ICUa).

(4) In Cluster Transfer Mode (EXDMACn.EDMTMD.MD[1:0] = 11b)

When the value of EDMCRB of EXDMACn changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in EDMCNT of EXDMACn is cleared to 0 and the DTIF flag in EDMSTS of EXDMACn is set to 1 at the same time. If the DTIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the EXDMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 11, Interrupt Control Unit (ICUa).

15.6.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the RPTIE bit in EDMINT of EXDMACn is set to 1. When the interrupt is requested to complete DMA transfer, the DTE bit in EDMCNT of EXDMACn is cleared to 0 and the ESIF flag in EDMSTS of EXDMACn is set to 1. If the ESIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DTE bit in EDMCNT of EXDMACn.

A repeat size end interrupt can be requested also in block transfer mode (or cluster transfer mode). In block transfer mode (or cluster transfer mode), the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size (or 1-cluster) data is completed.

Before sending an interrupt request from the EXDMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 11, Interrupt Control Unit (ICUa).

15.6.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the SARIE or DARIE bit in EDMINT of EXDMACn is set to 1, an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DTE bit in EDMCNT of EXDMACn is cleared to 0, and the ESIF flag in EDMSTS of EXDMACn is set to 1. If the ESIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.

In block transfer mode (or cluster transfer mode), even if an interrupt by an extended repeat area overflow is requested during a 1-block (or 1-cluster) transfer, the remaining data in the block (or the cluster) is transferred; transfer is terminated after a block (or cluster) transfer.

Before sending an interrupt request from the EXDMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 11, Interrupt Control Unit (ICUa).

15.7 Interrupts

The EXDMAC can output one interrupt request to the CPU or the DTC for each channel. Table 15.11 shows the relation among the interrupt sources, the interrupt status bits, and the interrupt enable bits. Figure 15.25 shows the schematic logic diagram of interrupt outputs. Procedures for restarting or suspending DMA transfer from within the processing routine for the EXDMAC interrupt are shown in Figure 15.26.

Table 15.11 Relation among Interrupt Sources, Interrupt Status Bits, and Interrupt Enable Bits

Interrupt Sources	Interrupt Enable Bits	Interrupt Status Bits	Request Output Enable Bits
Transfer end	—	EXDMACn.EDMSTS.DTIF	EXDMACn.EDMINT.DTIE
Escape transfer end	Repeat size end	EXDMACn.EDMINT.RPTIE	EXDMACn.EDMINT.ESIE
	Source address extended repeat area overflow	EXDMACn.EDMINT.SARIE	
	Destination address extended repeat area overflow	EXDMACn.EDMINT.DARIE	

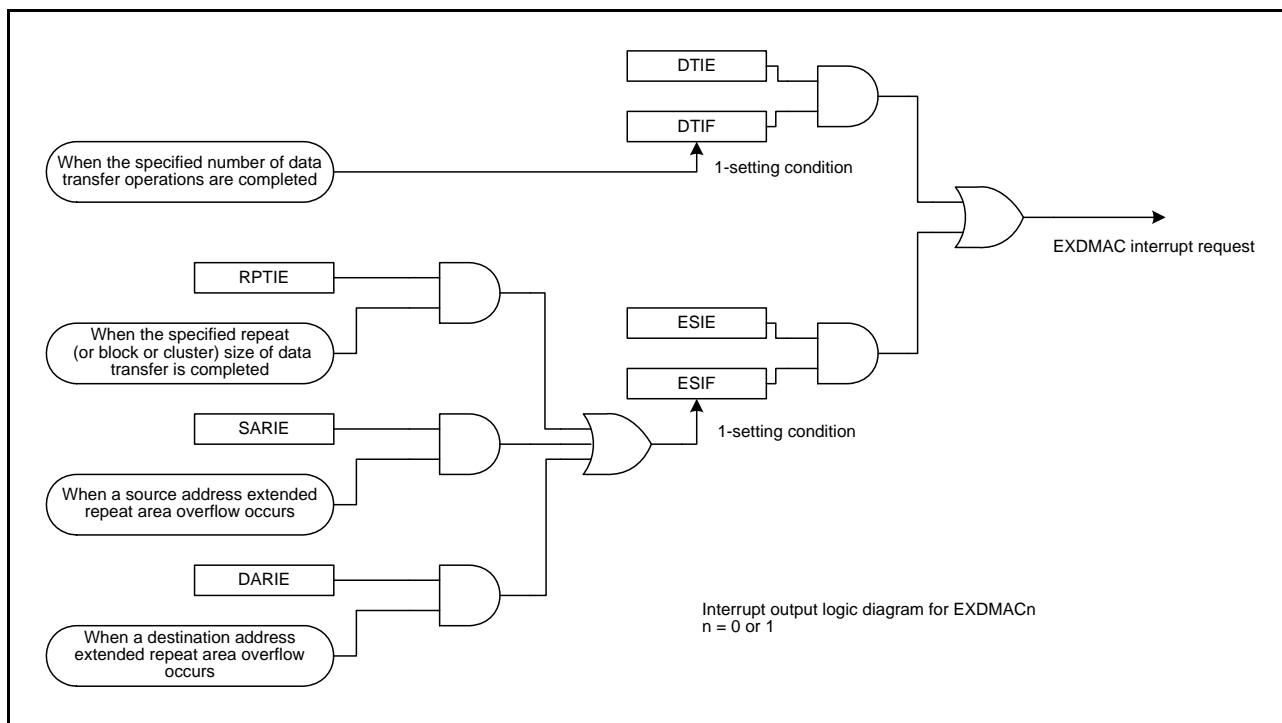


Figure 15.25 Schematic Logic Diagram of Interrupt Outputs

Procedures for EXDMAC interrupt processing differ according to whether DMA transfer is completed or to be suspended on the one hand or continued on the other.

(1) When Discontinuing or Terminating DMA Transfer

In the case of a transfer-completed interrupt, clear the interrupt source by writing 0 to the EXDMACn.EDMSTS.DTIF flag. In the case of a repeat-size interrupt or extended repeat-area overflow interrupt, clear the interrupt source by writing 0 to the EXDMACn.EDMSTS.ESIF flag. The EXDMAC remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DTE bit in EDMCNT of EXDMACn to 1.

(2) When Continuing DMA Transfer

Write 1 to the DTE bit in EDMCNT of EXDMACn. The ESIF bit in EDMSTS of EXDMACn is automatically cleared to 0 (interrupt source cleared), and DMA transfer is resumed.

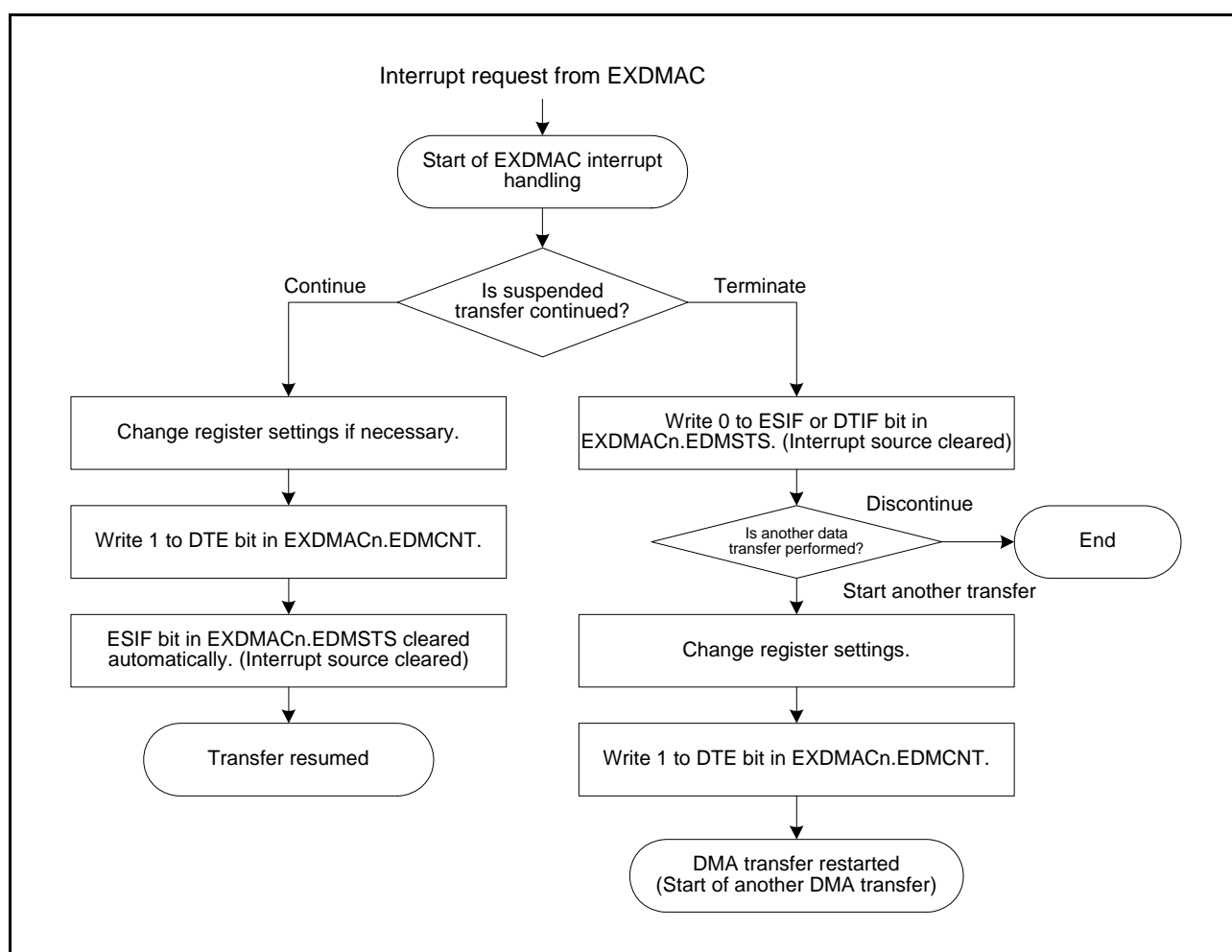


Figure 15.26 Restarting or Suspending DMA Transfer from within the Processing Routine for the EXDMAC Interrupt

15.8 Low-Power Consumption Function

To place the EXDMAC in the module-stop state, all-module clock stop mode, software-standby mode, or deep software-standby mode, clear the EDMAST.DMST bit to 0 (EXDMAC stopped), and then perform the following processing.

(1) Module Stop Function

Writing 1 to the MSTPCRA.MSTPA29 bit (transition to the module-stop state) enables the module-stop function of the EXDMAC. If DMA transfer is in progress at the time a 1 is written to the MSTPA29 bit, the transition to the module-stop state proceeds after DMAC transfer has ended.

Writing 0 to the MSTPA29 bit releases the EXDMAC from the module-stop state.

(2) All-Module Clock Stop Mode

Writing 1 to the MSTPCRA.ACSE bit (all-module clock stop mode enabled), writing 1 to all the bits in MSTPCRA and MSTPCRB, including the MSTPA29 bit (transition to the module-stop state), and executing a WAIT instruction causes a transition to the all-module clock stop mode. If DMA transfer is in progress at the time the WAIT instruction is executed, the EXDMAC can enter all-module clock stop mode after completion of the current DMA transfer.

After the EXDMAC returns from all-module clock stop mode, writing 0 to the MSTPA29 bit releases the EXDMAC from the module-stop state.

(3) Software Standby and Deep Software Standby Modes

Writing 1 to the SBYCR.SSBY bit (transition to software standby mode after WAIT instruction execution) and 0 to the DPSBYCR.DPSBY bit (transition to software standby mode after WAIT instruction execution), executing a WAIT instruction places the EXDMAC in software standby mode.

If DMA transfer is in progress at the time the WAIT instruction is executed, the EXDMAC enters software standby mode after completion of the current DMA transfer.

The EXDMAC enters deep software standby mode when the DPSBYCR.DPSBY bit is set to 1 (transition to deep software standby mode after WAIT instruction execution).

(4) Notes on Low-Power Consumption Function

For the timing of WAIT instruction execution and register settings, see section 9.7.7, Timing of Wait Instructions.

To perform DMA transfer after returning from low-power consumption mode, set the EDMAST.DMST bit to 1 again.

15.9 EDACK Operation in Single Address Mode

In single address mode, EDACK is output to one of the external transfer-source or transfer-destination devices and the address is simultaneously output to the other transfer device for access.

When the external device receiving EDACK transfers data to/from the CS area, the EDACKn negation timing can be adjusted by setting the DACKW bit in EDMOMD of EXDMACn. Specifically, the timing can be advanced by one BCLK cycle if the external device is a transfer destination and delayed by one BCLK cycle if the external device is a transfer source. When the external device receiving EDACK transfers data to/from the SDRAM, the EDACKn negation timing cannot be adjusted (setting the DACKW bit is invalid). For the CS area addresses and SDRAM area addresses, refer to section 4, Address Space.

The following sections show EDACK operation examples in single address mode, in which data is transferred to/from the CS and SDRAM areas in normal and block transfer modes.

15.9.1 EDACK Operation Example in Normal-Transfer (CS Area) Single Address Mode

Figure 15.27 shows the operation example in which data is transferred from the CS area to the device with EDACK in normal transfer mode. Setting the EXDMACn.EDMOMD.DACKW bit to 1 allows EDACK to be negated one BCLK cycle before the data read signal is negated.

Figure 15.28 shows the operation example in which data is transferred from the device with EDACK to the CS area in normal transfer mode. Setting the EXDMACn.EDMOMD.DACKW bit to 1 allows EDACK to be negated one BCLK cycle after the data write signal is negated.

For the data read signal, data write signal, and CS area access timing setting registers, refer to section 12, Buses.

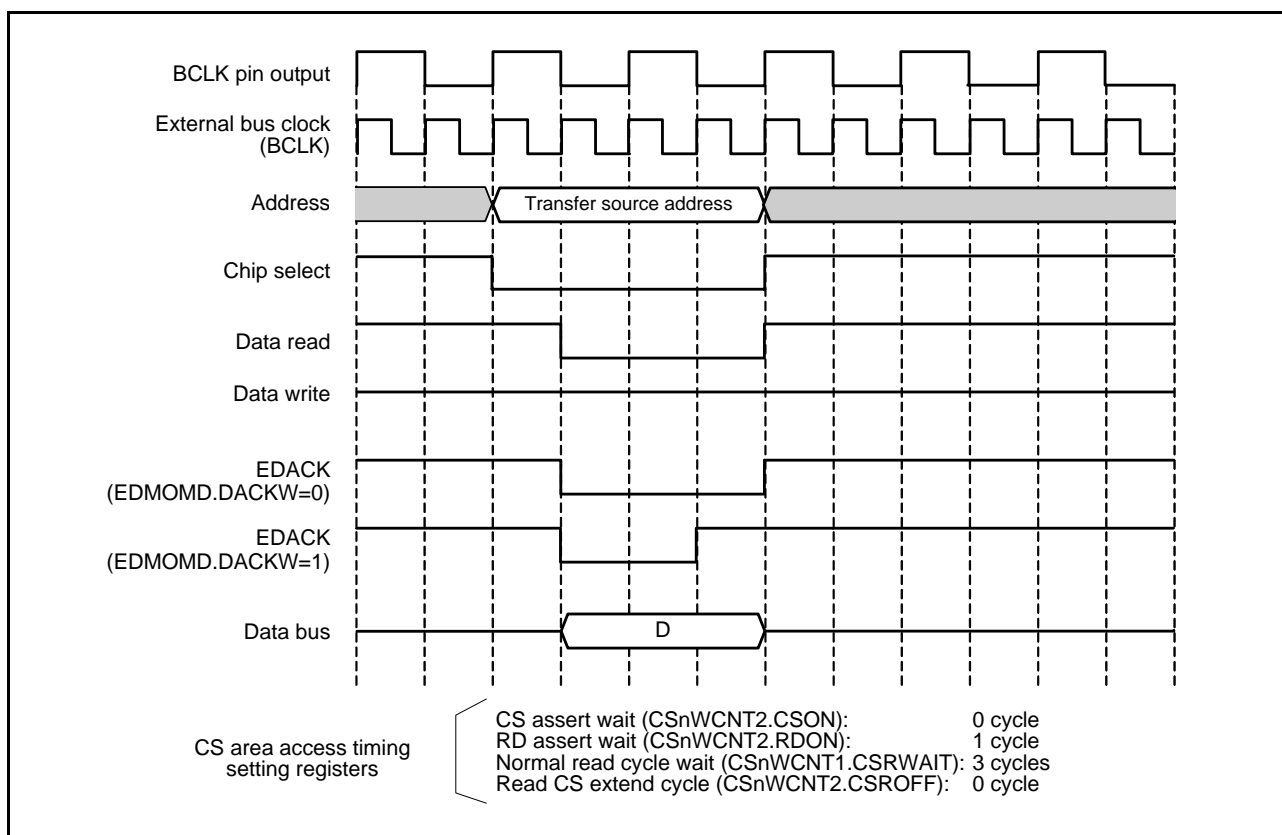


Figure 15.27 Operation Example in Normal-Transfer (CS Area Read) Single Address Mode

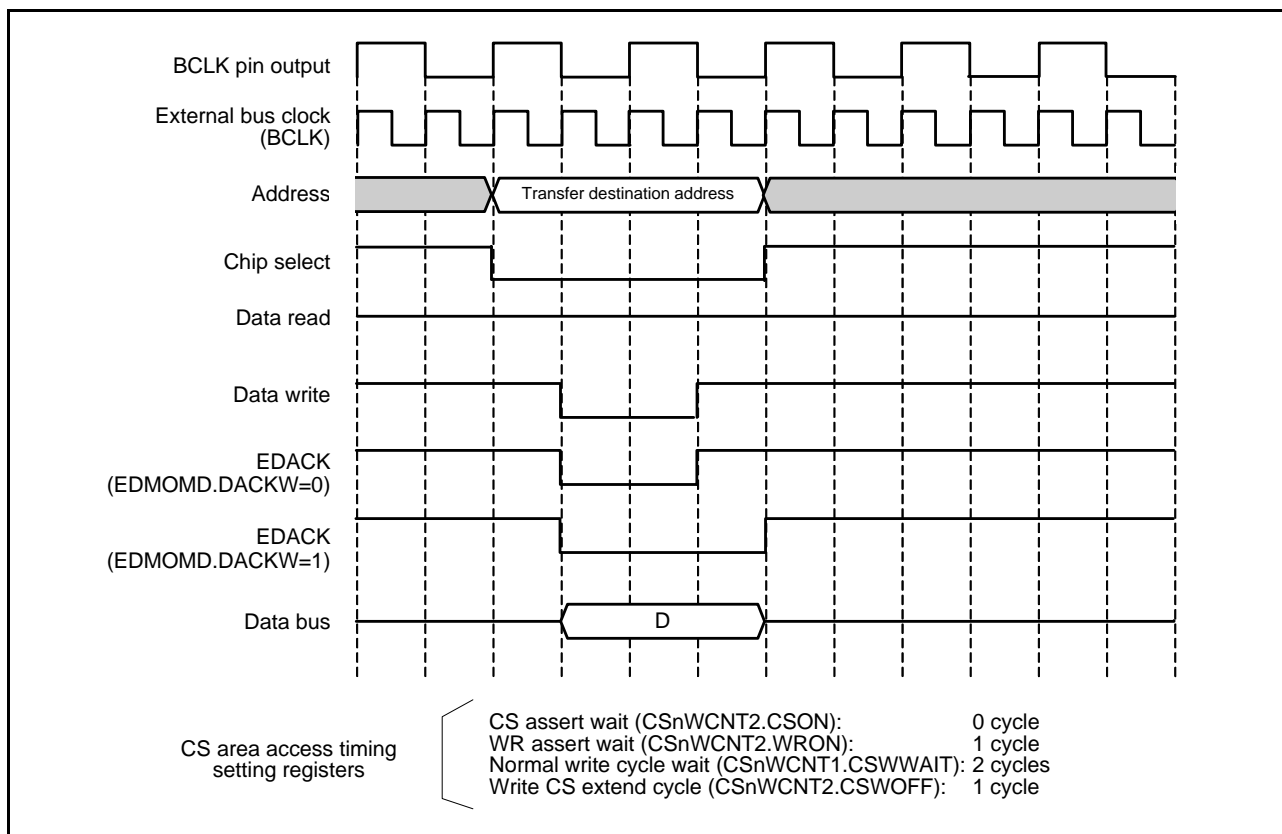


Figure 15.28 Operation Example in Normal-Transfer (CS Area Write) Single Address Mode

15.9.2 EDACK Operation Example in Normal-Transfer (SDRAM Area) Single Address Mode

Figure 15.29 shows the operation example in which data is transferred from SDRAM to the device with EDACK in normal transfer mode.

EDACK is asserted while SDRAM is outputting data.

Figure 15.30 shows the operation example in which data is transferred from the device with EDACK to SDRAM in normal transfer mode.

EDACK is asserted while SDRAM is writing data.

For the SDRAM commands and SDRAM access timing setting registers, refer to section 12, Buses.

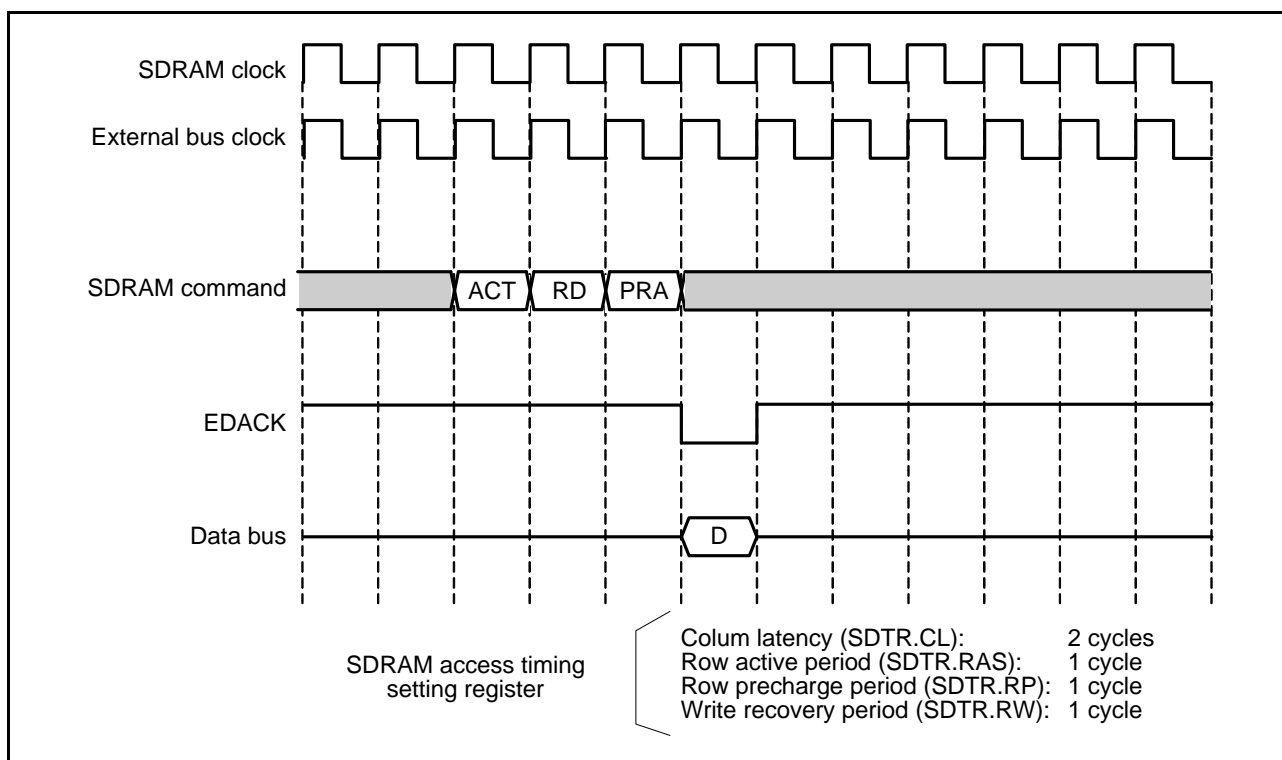


Figure 15.29 Operation Example in Normal-Transfer (SDRAM Area Read) Single Address Mode

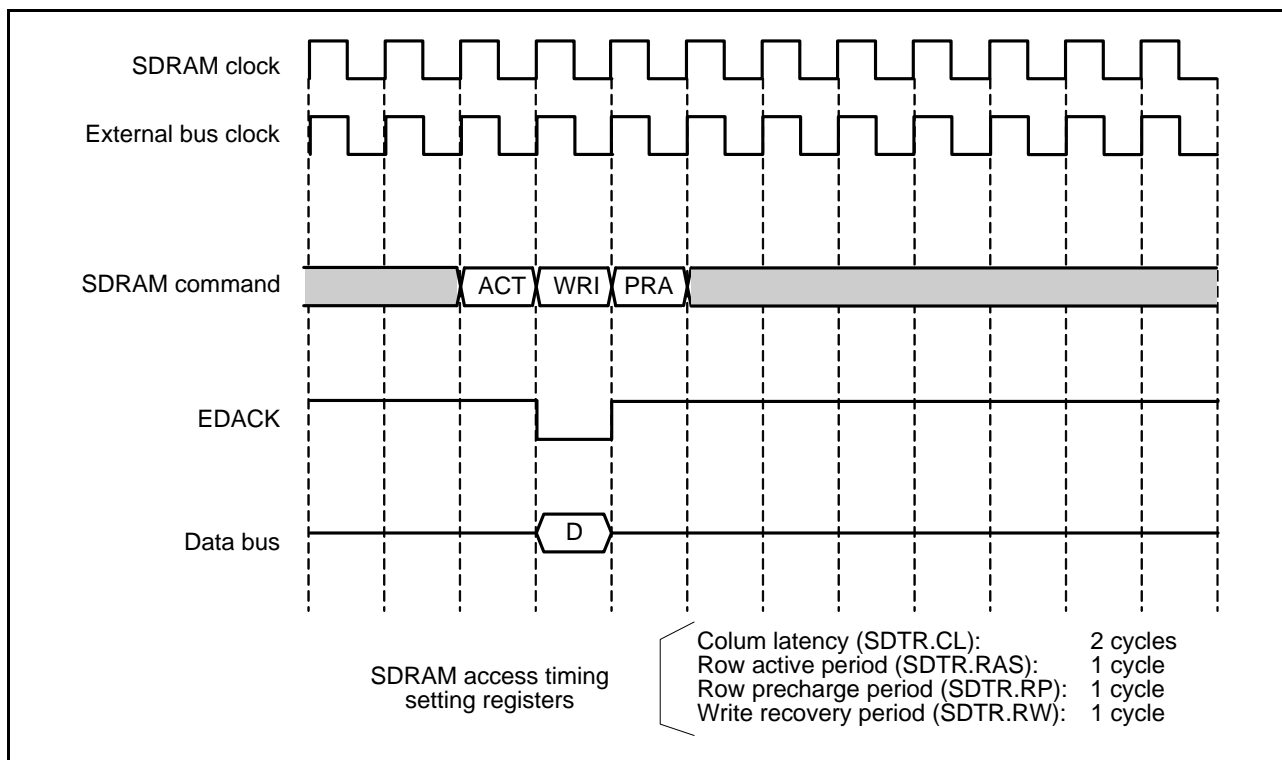


Figure 15.30 Operation Example in Normal-Transfer (SDRAM Area write) Single Address Mode

15.9.3 EDACK Operation Example in Block-Transfer (CS Area) Single Address Mode

Figure 15.31 shows the operation example in which data is transferred from the CS area to the device with EDACK in block transfer mode (block size = two). Setting the EXDMACn.EDMOMD.DACKW bit to 1 allows EDACK to be negated one BCLK cycle before the data read signal is negated.

Figure 15.32 shows the operation example in which data is transferred from the device with EDACK to the CS area in block transfer mode (block size = two). Setting the EXDMACn.EDMOMD.DACKW bit to 1 allows EDACK to be negated one BCLK cycle after the data write signal is negated.

For the data read signal, data write signal, and CS area access timing setting registers, refer to section 12, Buses.

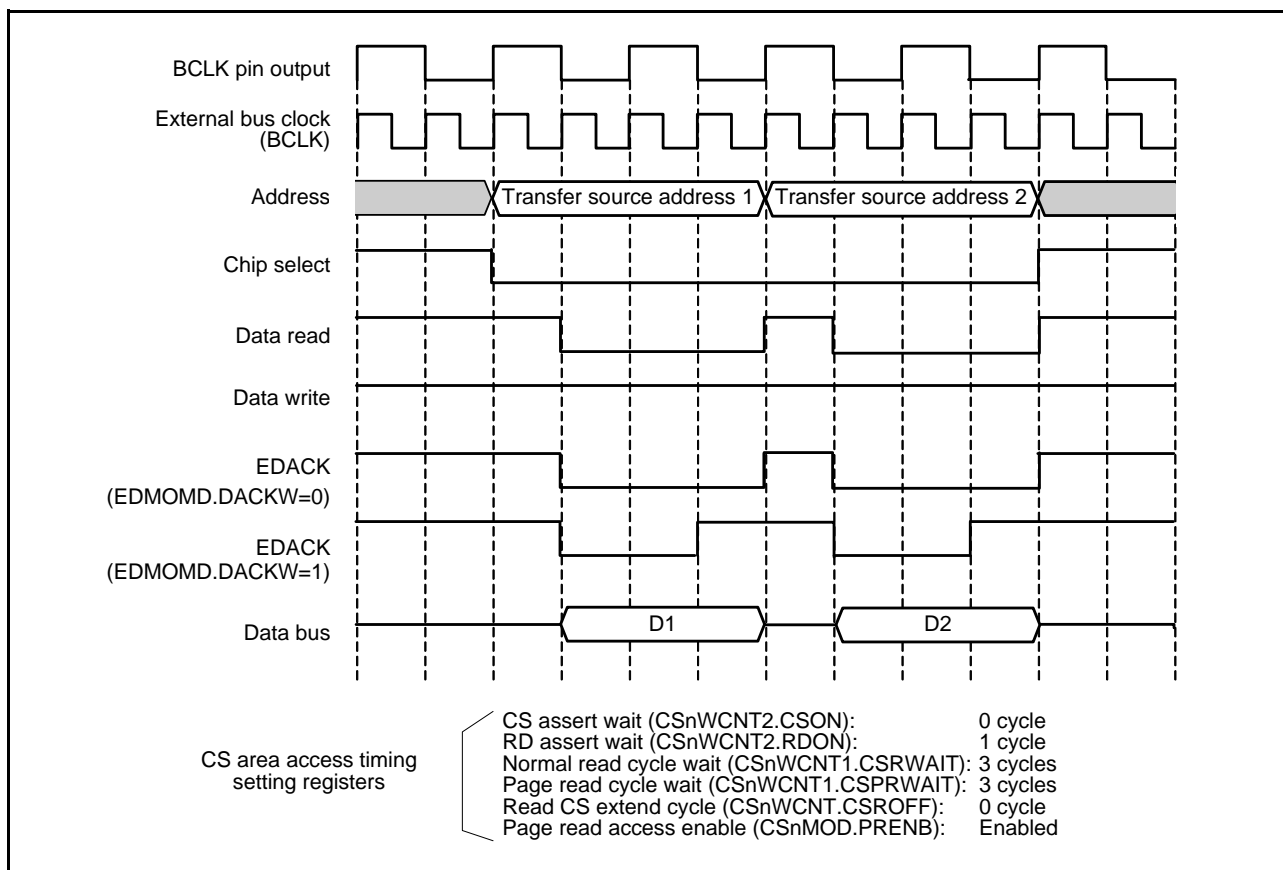


Figure 15.31 Operation Example in Block-Transfer (CS Area Read) Single Address Mode

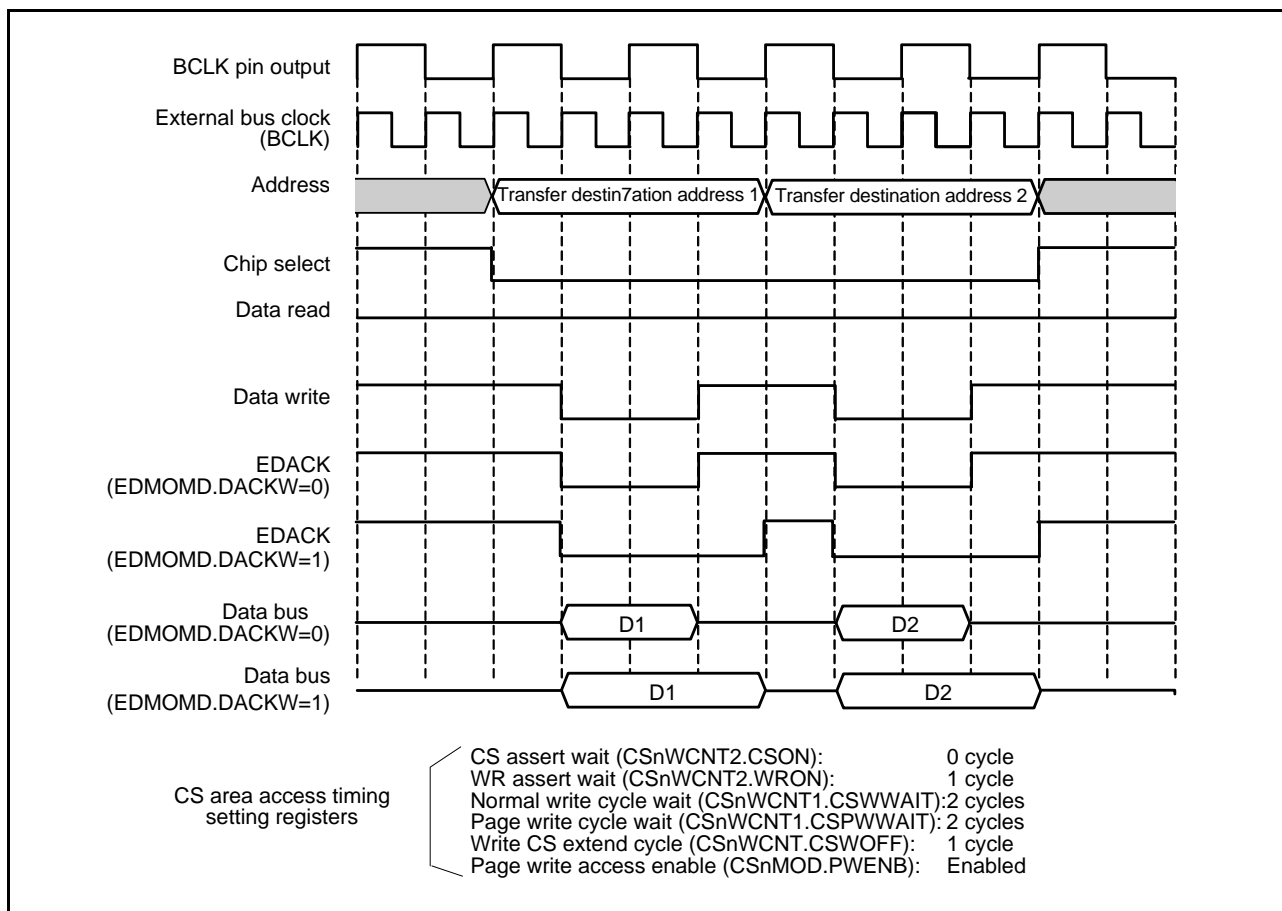


Figure 15.32 Operation Example in Block-Transfer (CS Area Write) Single Address Mode

15.9.4 EDACK Operation Example in Block-Transfer (SDRAM Area) Single Address Mode

Figure 15.33 shows the operation example in which data is transferred from SDRAM to the device with EDACK in block transfer mode (block size = four) with the BE bit in SDAMOD set to 1 to enable consecutive SDRAM access. EDACK is asserted while SDRAM is outputting data.

Figure 15.34 shows the operation example in which data is transferred from the device with EDACK to SDRAM in block transfer mode (block size = four) with the BE bit in SDAMOD set to 1 to enable consecutive SDRAM access. EDACK is asserted while SDRAM is writing data.

Figure 15.35 shows the operation example in which data is transferred from SDRAM to the device with EDACK in block transfer mode (block size = two) with the BE bit in SDAMOD set to 0 to disable consecutive SDRAM access. EDACK is asserted while SDRAM is outputting data.

Figure 15.36 shows the operation example in which data is transferred from the device with EDACK to SDRAM in block transfer mode (block size = two) with the BE bit in SDAMOD set to 0 to enable consecutive SDRAM access. EDACK is asserted while SDRAM is writing data.

For the SDRAM commands and SDRAM access timing setting registers, refer to section 12, Buses.

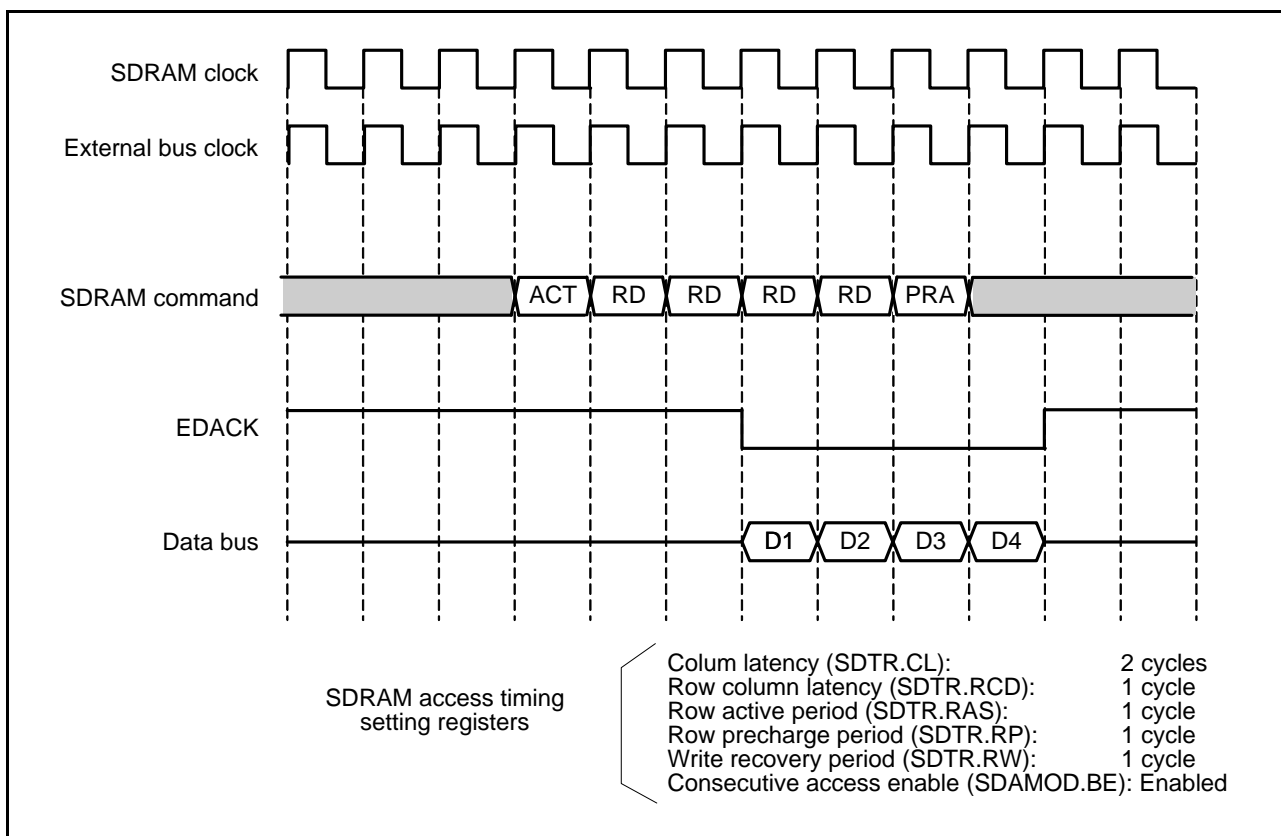


Figure 15.33 Operation Example in Block-Transfer (SDRAM Area Read: Consecutive Access Enabled) Single Address Mode

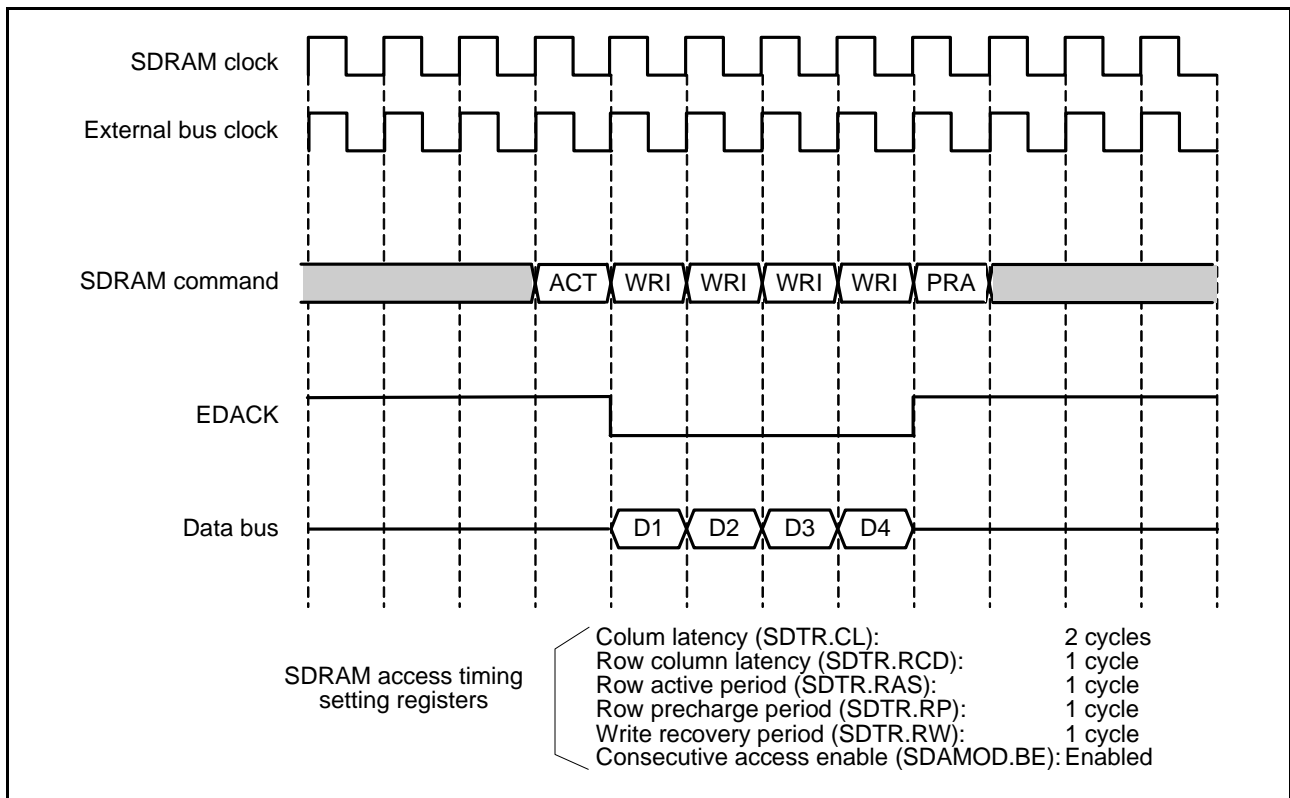


Figure 15.34 Operation Example in Block-Transfer (SDRAM Area Write: Consecutive Access Enabled) Single Address Mode

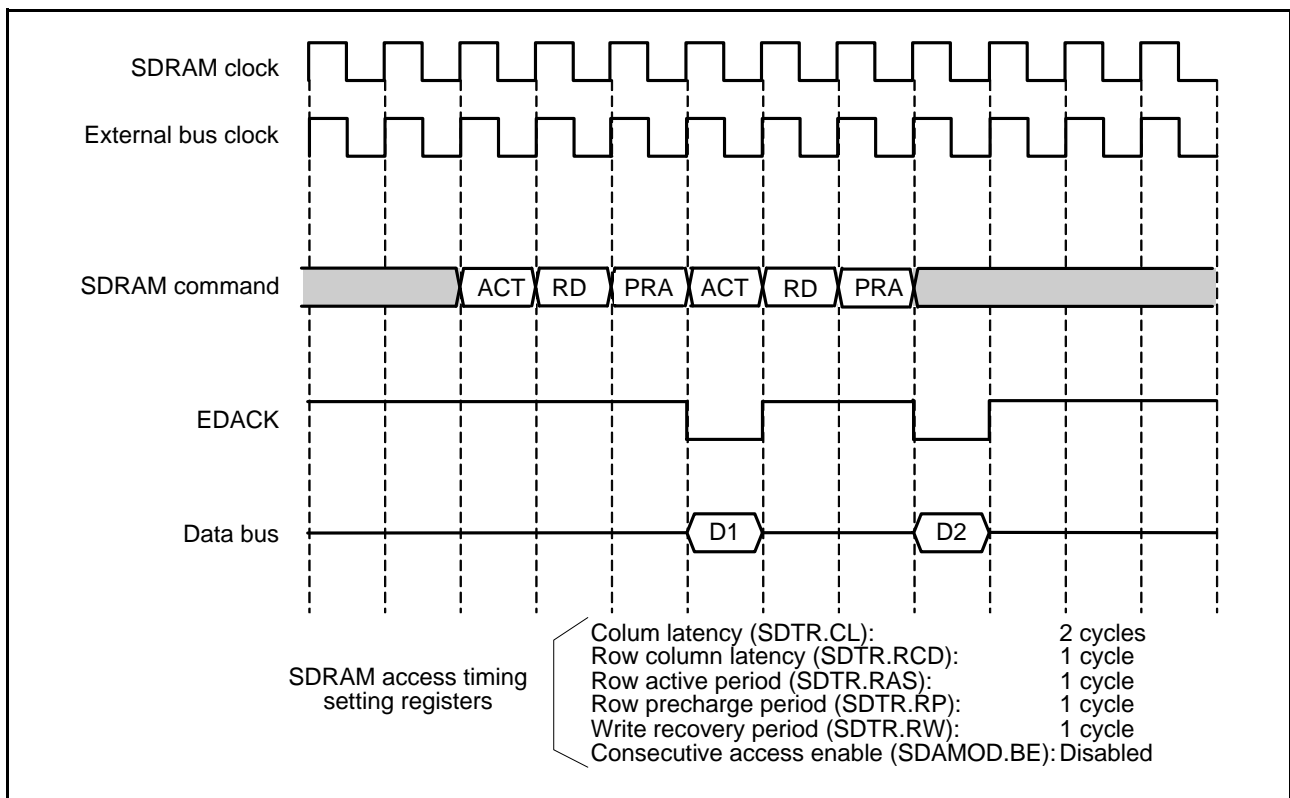


Figure 15.35 Operation Example in Block-Transfer (SDRAM Area Read: Consecutive Access Disabled) Single Address Mode

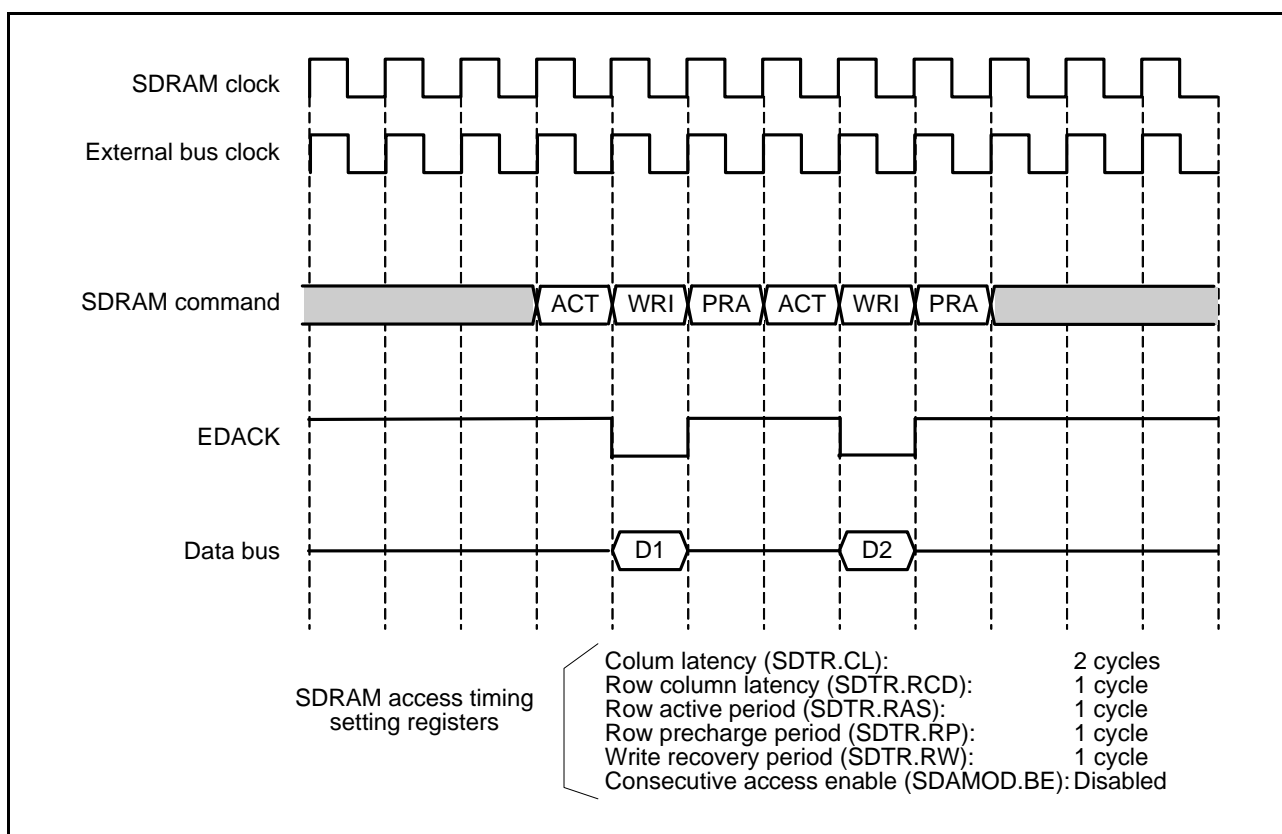


Figure 15.36 Operation Example in Block-Transfer (SDRAM Area Write: Consecutive Access Disabled) Single Address Mode

15.10 Usage Notes

15.10.1 Cluster Buffers

The EXDMAC provides seven 32-bit cluster buffers (CLSBR0 to CLSBR6), in which data is stored in the different manner depending on the transfer size setting (SZ bits in EDMTMD of EXDMACn).

Figure 15.37 shows how data is stored in cluster buffers.

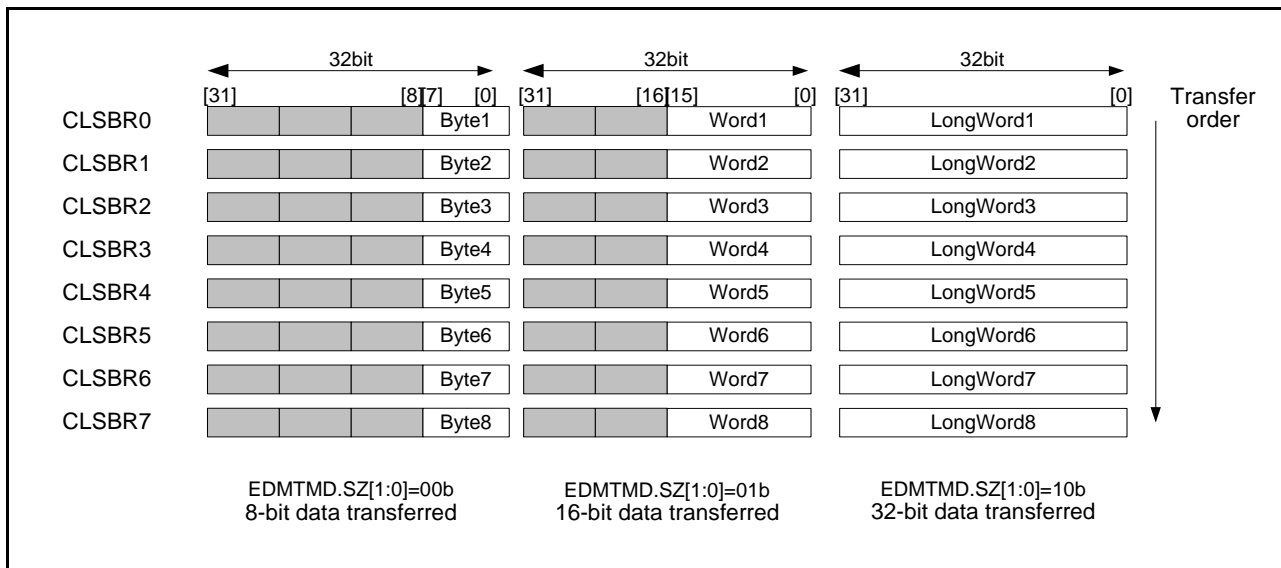


Figure 15.37 Data Storage in Cluster Buffers

15.10.2 Access to the Registers during DMA Transfer

The EDMSAR, EDMDAR, EDMCRA, EDMCRB, EDMTMD, EDMOMD, EDMINT, EDMAMD, EDMOFR, and EDMRMD registers of EXDMACn must not be accessed while the ACT bit in EDMSTS of the same channel is set to 1 (DMA operating state) or the DTE bit in EDMCNT of the same channel is set to 1 (DMA transfer enabled).

15.10.3 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

16. Data Transfer Controller (DTCa)

The RX62N/RX621 Group incorporates a data transfer controller (DTC).

The DTC is activated by an interrupt request to control data transfer.

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 mode	<ul style="list-style-type: none"> • Normal transfer mode A single activation leads to a single data transfer. • Repeat transfer mode A single activation leads to a single data transfer. The transfer address is returned to the transfer start address after the number of data transfers corresponding to "repeat size". The maximum repeat size is 256 data. • Block transfer mode A single activation leads to the transfer of a single block. The maximum block size is 255 data.
Transfer channel	<ul style="list-style-type: none"> • Channel transfer corresponding to the interrupt source is possible (transferred by DTC activation request from the ICU). • Data of multiple channels can be transferred on a single activation source (chain transfer). • Either "executed when the counter is 0" or "always executed" can be selected for chain transfer.
Transfer space	<ul style="list-style-type: none"> • In short-address mode: 16 Mbytes (Areas from 0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh excepting reserved areas) • In full-address mode: 4 Gbytes (Area from 0000 0000h to FFFF FFFFh excepting reserved areas)
Data transfer units	<ul style="list-style-type: none"> • Length of a single data: 8, 16, or 32 bits • Number of data for a single block: 1 to 255 data
CPU interrupt source	<ul style="list-style-type: none"> • An interrupt request can be generated to the CPU on a DTC activation interrupt. • An interrupt request can be generated to the CPU after a single data transfer. • An interrupt request can be generated to the CPU after data transfer of specified volume.
Read skip	Transfer data read skip can be specified.
Write-back skip	When "fixed" is selected for transfer source address and/or transfer destination address, write-back skip execution is provided.

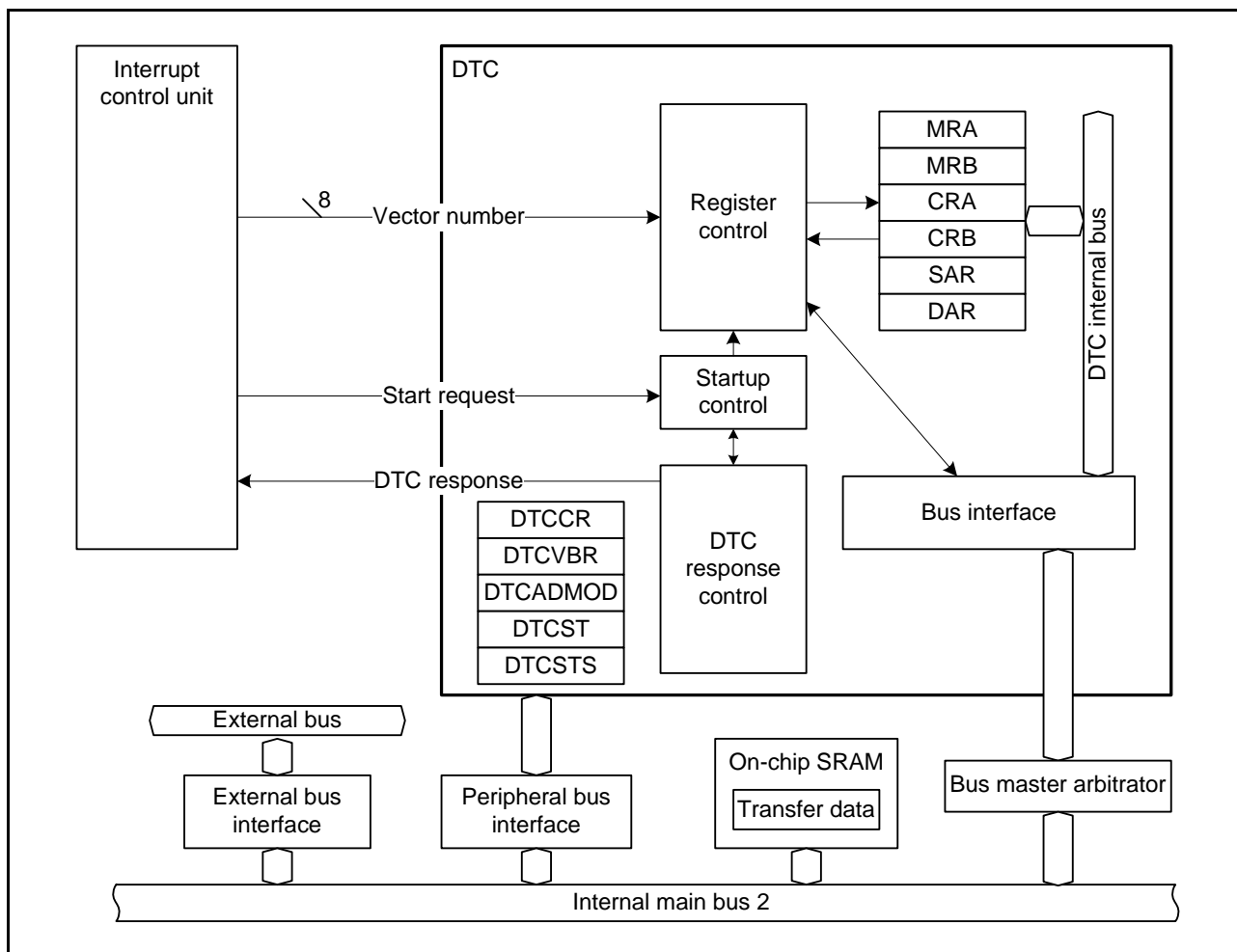


Figure 16.1 Block Diagram of the DTC

16.2 Register Descriptions

Table 16.2 lists the registers of the DTC.

Registers MRA, MRB, SAR, DAR, CRA, and CRB are DTC internal registers, which cannot be directly accessed from the CPU. Values to be set in the DTC internal registers are placed in the RAM area as transfer information data. When an activation request is generated, the DTC reads the transfer information data from the RAM area and set them in the internal registers. After the data transfer ends, the internal register contents are written back to the RAM area as transfer information data.

Table 16.2 Registers of the DTC

Register Name	Symbol	Value after Reset	Address	Access Size(Bits)
DTC mode register A	MRA	xxh	—	—
DTC mode register B	MRB	xxh	—	—
DTC transfer source address register	SAR	xxxxxxxxh	—	—
DTC transfer destination address register	DAR	xxxxxxxxh	—	—
DTC transfer count register A	CRA	xxxxh	—	—
DTC transfer count register B	CRB	xxxxh	—	—
DTC control register	DTCCR	08h	0008 2400h	8 bits
DTC vector base register	DTCVBR	00000000h	0008 2404h	32 bits
DTC address mode register	DTCADMOD	00h	0008 2408h	8 bits
DTC module start register	DTCST	00h	0008 240Ch	8 bits
DTC status register	DTCSTS	0000h	0008 240Eh	16 bits

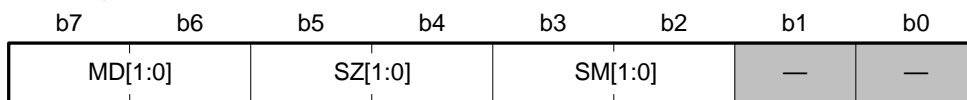
Note: To activate the DTC, a setting of the DTCERi.DTCE bit (i = interrupt vector number) and IERm.IENj (m = 02h to 1Fh, j = 7 to 0) bits in the interrupt control unit (ICU) is required. For details, refer to section 11, Interrupt Control Unit (ICUa).

[Legend]

x: Undefined value

16.2.1 DTC Mode Register A (MRA)

Address (inaccessible directly from the CPU)



Value after reset:

[Legend] x: Undefined

Bit	Symbol	Bit	Symbol	R/W
b1, b0	—	Reserved	These bits are always read as undefined. The write value should be 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 0 0: SAR value is address fixed (Write-back to SAR is skipped) 0 1: SAR value is address 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 transfer 0 1: Word transfer 1 0: Longword transfer 1 1: Setting prohibited	—
b7, b6	MD[1:0]	DTC Transfer Mode Select	b7 b6 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

MRA is used to select the operating mode of the DTC.

MRA cannot be accessed directly from the CPU.

SM[1:0] Bits (Transfer Source Address Addressing Mode)

These bits specify the SAR operation after data transfer.

SZ[1:0] Bits (DTC Data Transfer Size)

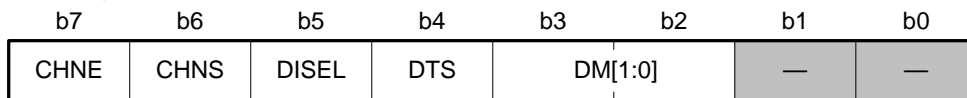
These bits specify the transfer data size.

MD[1:0] Bits (DTC Transfer Mode Select)

These bits specify the transfer mode of the DTC.

16.2.2 DTC Mode Register B (MRB)

Address (inaccessible directly from the CPU)



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

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are always read as undefined. The write value should be 0.	—
b3, b2	DM[1:0]	Transfer Destination Address Addressing Mode	b3 b2 0 0: DAR value is address fixed (Write-back to DAR is skipped) 0 1: DAR value is address fixed (Write-back to DAR is skipped) 1 0: DAR value is incremented after data transfer (+1 when SZ[1:0] bits in MRA = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b) 1 1: DAR value is decremented after data transfer (-1 when SZ[1:0] bits in MRA = 00b, -2 when SZ[1:0] bits = 01b, -4 when SZ[1:0] bits = 10b)	—
b4	DTS	DTC Transfer Mode Select	0: Transfer destination side is repeat area or block area 1: Transfer source side is repeat area or block area	—
b5	DISEL	DTC Interrupt Select	0: An interrupt request to the CPU is generated when specified data transfer is completed 1: An interrupt request to the CPU is generated each time DTC data transfer is performed	—
b6	CHNS	DTC Chain Transfer Select	0: Chain transfer is performed continuously 1: Chain transfer is performed only when the transfer counter is 0	—
b7	CHNE	DTC Chain Transfer Enable	0: Chain transfer is disabled 1: Chain transfer is enabled	—

MRB is used to select the operating mode of the DTC.

MRB cannot be accessed directly from the CPU.

DM[1:0] Bits (Transfer Destination Address Addressing Mode)

These bits specify the DAR operation after data transfer.

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.

DISEL Bit (DTC Interrupt Select)

The DISEL bit specifies whether to generate an interrupt request to the CPU each time DTC data transfer is performed or when specified data transfer is completed.

CHNS Bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition.

When the next transfer is chain transfer, completion of specified transfer count is not checked and the interrupt status flag is not cleared. Moreover, an interrupt request to the CPU is not generated.

CHNE Bit (DTC Chain Transfer Enable)

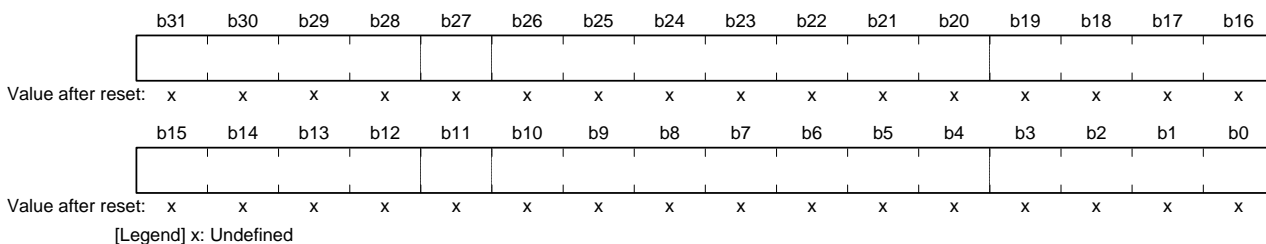
The CHNE bit enables or disables chain transfer.

The chain transfer condition is selected by the CHNS bit.

For details of chain transfer, see section 16.4.6, Chain Transfer.

16.2.3 DTC Transfer Source Address register (SAR)

Address (inaccessible directly from the CPU)



SAR is used to set the transfer source start address.

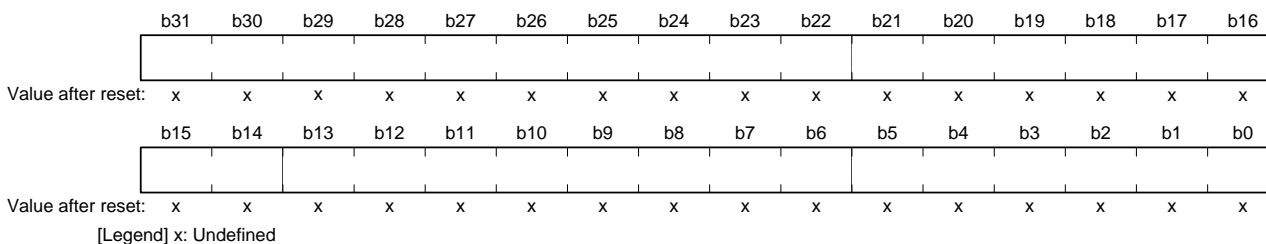
In full-address mode, 32 bits are valid.

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 Address Register (DAR)

Address (inaccessible directly from the CPU)



DAR is used to set the transfer destination start address.

In full-address mode, 32 bits are valid.

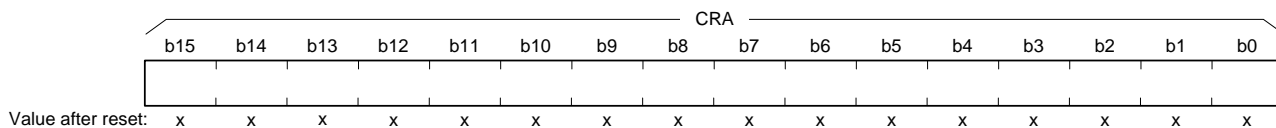
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.

DAR cannot be accessed directly from the CPU.

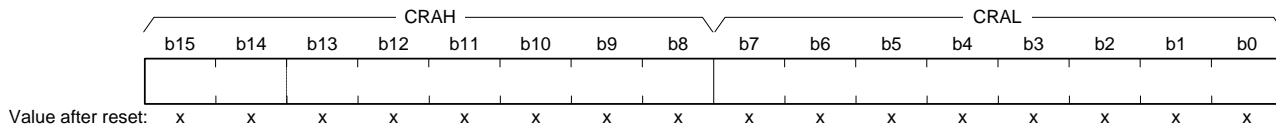
16.2.5 DTC Transfer Count Register A (CRA)

Address (inaccessible directly from the CPU)

- Normal transfer mode



- Repeat transfer mode/block transfer mode



Notes: 1. The function depends on transfer mode.
 2. x: Undefined

Symbol	Register Name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count.	—
CRAH	Transfer Counter A Upper Register		—

Note : Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

CRA is used to set the transfer count of the DTC.

The function of this register depends on transfer mode.

CRA cannot be accessed directly from the CPU.

(1) Normal transfer mode (MD[1:0] bits in MRA = 00b)

CRA functions as a 16-bit transfer counter in normal transfer mode.

The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.

The CRA value is decremented (-1) at each data transfer.

(2) Repeat transfer mode (MD[1:0] bits in MRA = 01b)

The CRAH register retains transfer count and the CRAL register functions as an 8-bit transfer counter.

The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.

The CRAL value is decremented (-1) at each data transfer. When it reaches 00h, the CRAH value is transferred to CRAL.

(3) Block transfer mode (MD[1:0] bits in MRA = 10b)

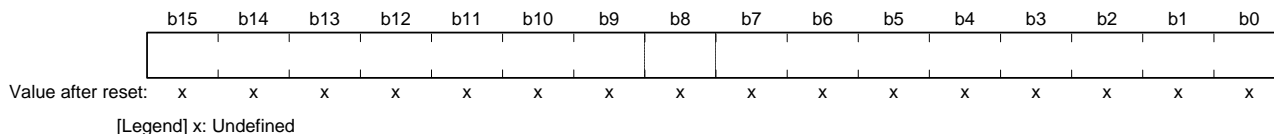
The CRAH register retains block size and the CRAL register functions as an 8-bit block size counter.

The transfer count is 1 and 255 when the set value is 01h and FFh, respectively. Setting a value of 00h is prohibited.

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 (inaccessible directly from the CPU)



CRB is used to set the block transfer count for block transfer mode.

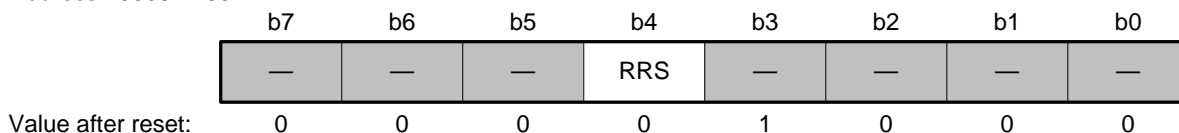
The block transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively. The CRB value is decremented (-1) at each data transfer.

When normal transfer mode or repeat transfer mode is selected, set a value of FFFFh to the CRB.

CRB cannot be accessed directly from the CPU.

16.2.7 DTC Control Register (DTCCR)

Address: 0008 2400h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b3	—	Reserved	This bit is always read as 1. The write value should be 1.	R/W
b4	RRS	DTC Transfer Data Read Skip Enable	0: Transfer data read is not skipped 1: Transfer data read is skipped when vector numbers match.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

DTCCR is used to specify the control of the DTC.

RRS Bit (DTC Transfer Data Read Skip Enable)

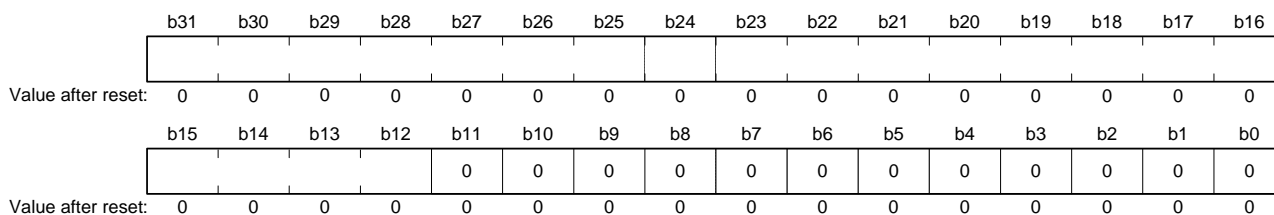
The DTC vector number is always compared with the vector number in the previous startup process.

When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred data. However, when the previous transfer was chain transfer, the transferred data is always read regardless of the value of RRS bit.

Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, the transferred data is always read regardless of the value of RRS bit.

16.2.8 DTC Vector Base Register (DTCVBR)

Address: 0008 2404h



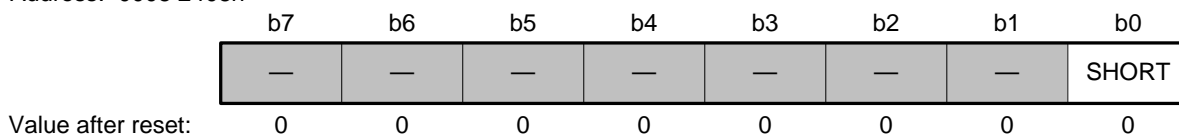
DTCVBR is used to set the base address for calculating the DTC vector table address.

The lower 12 bits (b11 to b0) are always 0 and cannot be modified.

The upper 4 bits (b31 to b28) are ignored, and the address of this register is extended by the value specified by b27.

16.2.9 DTC Address Mode Register (DTCADM0D)

Address: 0008 2408h



Bit	Symbol	Bit Name	Description	R/W
b0	SHORT	Short-Address Mode	0: Full-address mode 1: Short-address mode	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

DTCADM0D is used to select an address mode that specifies DTC's accessible area.

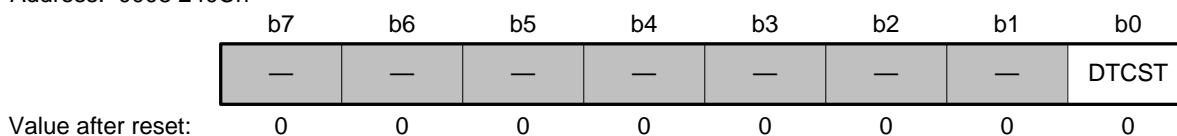
SHORT Bit (Short-Address Mode)

Full-address mode allows the DTC to access to a 4GB space (00000000h to FFFFFFFFh).

Short-address mode allows the DTC to access to a 16MB space (00000000h to 007FFFFFh and FF800000h to FFFFFFFFh).

16.2.10 DTC Module Start Register (DTCST)

Address: 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 always read as 0. The write value should be 0.	R/W

DTCST is used to start or stop the DTC module.

DTCST Bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept start requests. When this bit is cleared to 0, start requests are no longer accepted.

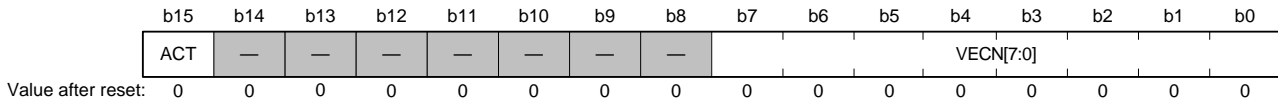
If this bit is cleared to 0 during data transfer, the accepted start request is active until the processing is completed.

To enable transitions to the module-stop state, all-module clock-stop state, software-standby mode, or deep software-standby mode, the DTCST bit must be set to "0".

For details on the facilities for transitions to the module-stop state, all-module clock-stop state, software-standby mode, and deep software-standby mode, refer to section 16.8, Low-Power Consumption Function, and section 9, Low Power Consumption.

16.2.11 DTC Status Register (DTCSTS)

Address: 0008 240Eh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	VECN[7:0]	DTC-Activating Vector Number Monitoring	These bits indicate the vector number for the activating source when DTC transfer is in progress. The value is only valid if DTC transfer is in progress (the value of the ACT flag is 1).	R
b14 to b8	—	Reserved	These bits are always read as 0. Writing to this bit has no effect.	R
b15	ACT	DTC Active Flag	0: DTC transfer operation is not in progress. 1: DTC transfer operation is in progress.	R

The DTCSTS is used to indicate the state of DTC transfer operation.

VECN[7:0] Bits (DTC-Activating Vector Number Monitoring)

While transfer by the DTC is in progress, these bits indicate the vector number corresponding to the activating source for the transfer.

When the DTCSTS register is read, the value read from the VECN[7:0] bits is valid if the value of the ACT flag was 1 (indicating DTC transfer in progress) and invalid if the value of the ACT flag was 0 (indicating no current DTC transfer). See Table 16.3 for the correspondence between activating sources and vector numbers.

ACT Bit (DTC Active Flag)

This bit indicates the state of DTC transfer operation.

[Setting condition]

- When the DTC is activated by a start request

[Clearing condition]

- When transfer by the DTC is completed in response to a start request.

16.3 Sources of Activation

The DTC is activated by an interrupt request. Setting the DTCERn.DTCE bit (where n is the interrupt vector number of the given interrupt) of the ICU to 1 selects the corresponding interrupt as an activation source for the DTC.

For the software activation (SWINT), see section 11, Interrupt Control Unit (ICUa).

On completion of a single round of data transfer (or the last of the consecutive transfers in the case of a chained transfer), follow the procedure below.

- On completion of a specified round of data transfer, the DTCERn.DTCE bit is cleared to 0 and an interrupt is requested to the CPU
- On completion of other transfers, if the MRB.DISEL bit is 1, an interrupt is requested to the CPU. If the MRB.DISEL bit is 0, the interrupt status flag (IRi.IR) of the activation source is cleared to 0.

16.3.1 Allocating Transfer Data and DTC Vector Table

The DTC reads the start address of transfer information data from the vector table for each activation source.

The vector table should be allocated so that the lower 12 bits of the base address (start address) are 0. The base address of the DTC vector table should be set in the DTC vector base register (DTCVBR).

Transfer information data is allocated in the RAM area. The start address of transfer information data (n) of vector No. n should be “vector table base address plus 4n”.

Transfer information data can be allocated with 3 longwords (short-address mode) or 4 longwords (full-address mode). Use the SHORT bit in DTCADM0D to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).

Figure 16.2 shows the correspondence between the DTC vector table and transfer information data. Figure 16.3 shows allocation of transfer information data in the RAM area. The lower address differs according to the endian mode of the allocation area. For details, see Figure 16.16.

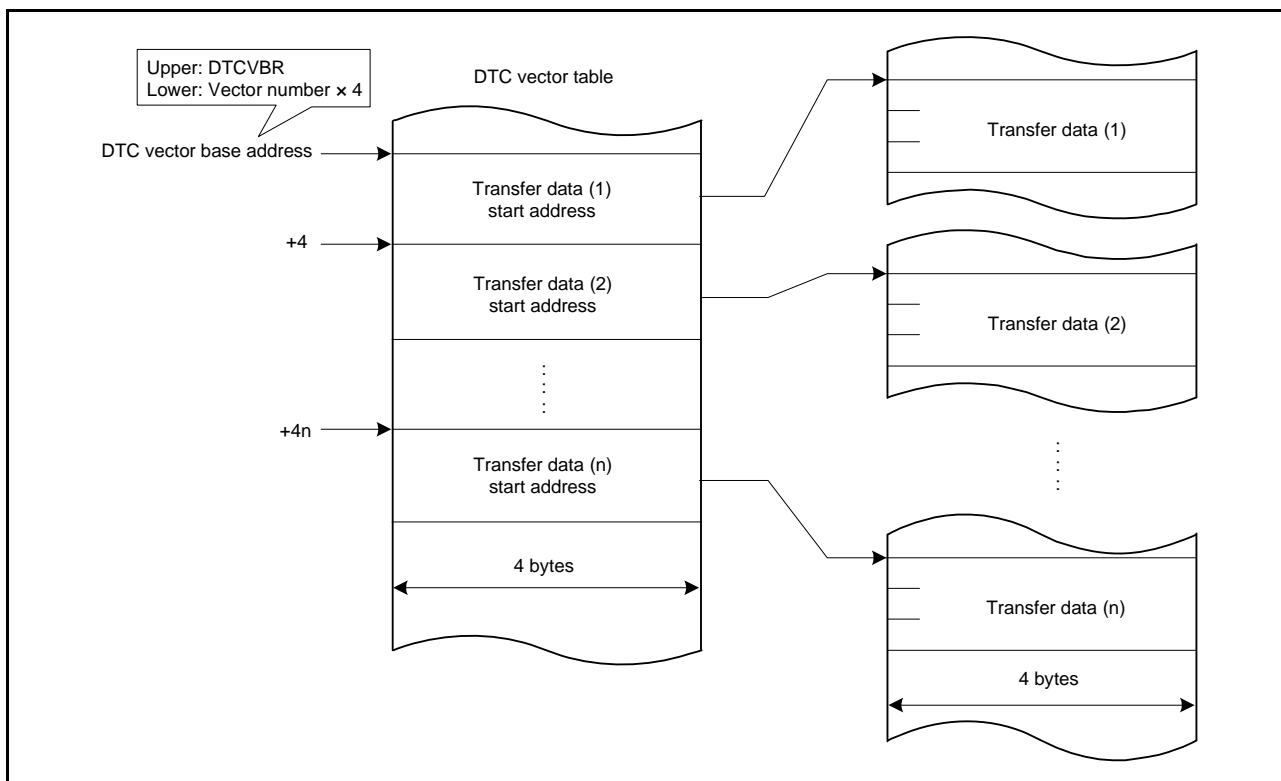


Figure 16.2 DTC Vector Table and Transfer Data

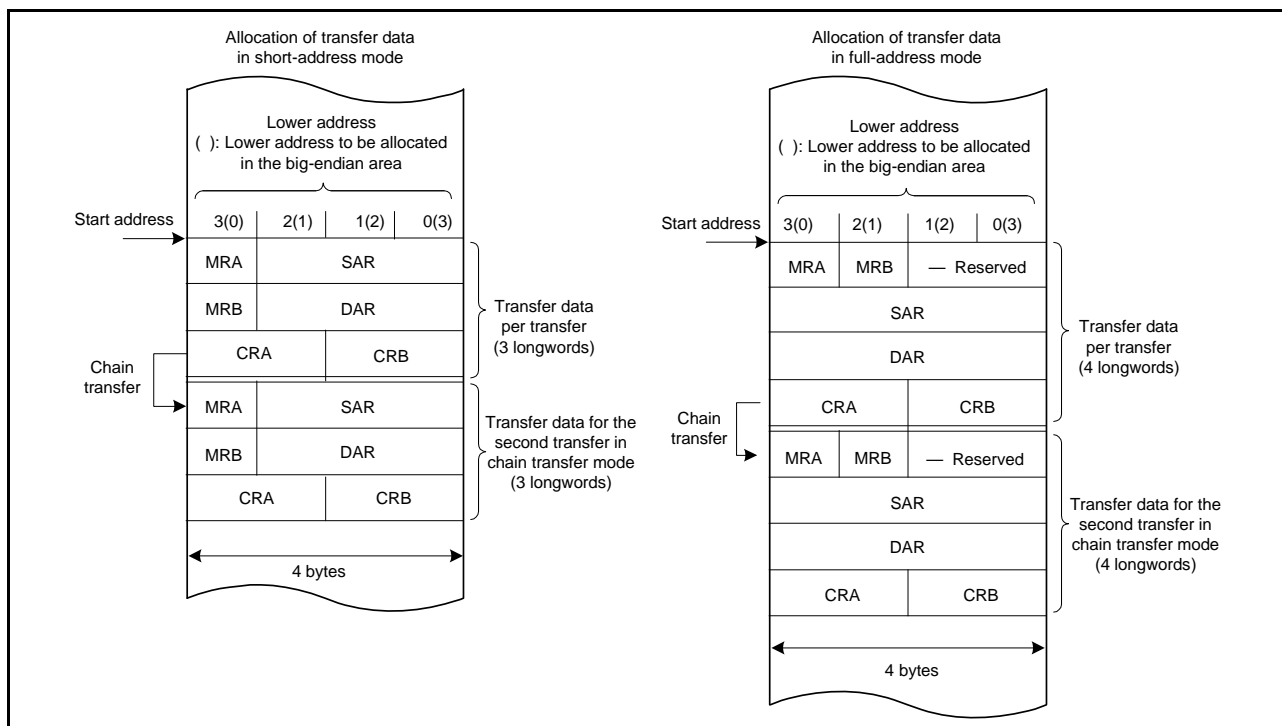


Figure 16.3 Allocation of Transfer Data in the RAM Area

16.3.2 Startup source and Vector Address

Table 16.3 shows the correspondence between DTC startup sources and vector addresses.

Table 16.3 Correspondence between Interrupt Sources, DTC Vector Addresses, and the ICU.DTCERn Register (1 / 3)

Startup Source Generation Source	Startup Source	Vector Number	DTC Vector Address Offset	ICU.DTCERn	Priority *
ICU	SWINT	27	006Ch	ICU.DTCER027	↑ High
CMT0	CMI0	28	0070h	ICU.DTCER028	
CMT1	CMI1	29	0074h	ICU.DTCER029	
CMT2	CMI2	30	0078h	ICU.DTCER030	
CMT3	CMI3	31	007Ch	ICU.DTCER031	
USB0	D0FIFO0	36	0090h	ICU.DTCER036	
	D1FIFO0	37	0094h	ICU.DTCER037	
USB1	D0FIFO1	40	00A0h	ICU.DTCER040	
	D1FIFO1	41	00A4h	ICU.DTCER041	
RSPI0	SPRI0	45	00B4h	ICU.DTCER045	
	SPTI0	46	00B8h	ICU.DTCER046	
RSPI1	SPRI1	49	00C4h	ICU.DTCER049	
	SPTI1	50	00C8h	ICU.DTCER050	
External pins	IRQ0	64	0100h	ICU.DTCER064	
	IRQ1	65	0104h	ICU.DTCER065	
	IRQ2	66	0108h	ICU.DTCER066	
	IRQ3	67	010Ch	ICU.DTCER067	
	IRQ4	68	0110h	ICU.DTCER068	
	IRQ5	69	0114h	ICU.DTCER069	
	IRQ6	70	0118h	ICU.DTCER070	
	IRQ7	71	011Ch	ICU.DTCER071	
	IRQ8	72	0120h	ICU.DTCER072	
	IRQ9	73	0124h	ICU.DTCER073	
	IRQ10	74	0128h	ICU.DTCER074	
	IRQ11	75	012Ch	ICU.DTCER075	
	IRQ12	76	0130h	ICU.DTCER076	
	IRQ13	77	0134h	ICU.DTCER077	
	IRQ14	78	0138h	ICU.DTCER078	
	IRQ15	79	013Ch	ICU.DTCER079	
AD0	ADI0	98	0188h	ICU.DTCER098	
AD1	ADI1	99	018Ch	ICU.DTCER099	
S12AD	S12ADI0	102	0198h	ICU.DTCER102	
MTU0	TGIA0	114	01C8h	ICU.DTCER114	
	TGIB0	115	01CCh	ICU.DTCER115	
	TGIC0	116	01D0h	ICU.DTCER116	
	TGID0	117	01D4h	ICU.DTCER117	Low

Table 16.3 Correspondence between Interrupt Sources, DTC Vector Addresses, and the ICU.DTCERn Register (3 / 3)

Startup Source Generation Source	Startup Source	Vector Number	DTC Vector Address Offset	ICU.DTCERn	Priority *
TMR3	CMIA3	183	02DCh	ICU.DTCER183	High ↑
	CMIB3	184	02E0h	ICU.DTCER184	
DMACA	DMACI0	198	0318h	ICU.DTCER198	
	DMACI1	199	031Ch	ICU.DTCER199	
	DMACI2	200	0320h	ICU.DTCER200	
	DMACI3	201	0324h	ICU.DTCER201	
EXDMAC	EXDMACI0	202	0328h	ICU.DTCER202	
	EXDMACI1	203	032Ch	ICU.DTCER203	
SCI0	RXI0	215	035Ch	ICU.DTCER215	
	TXI0	216	0360h	ICU.DTCER216	
SCI1	RXI1	219	036Ch	ICU.DTCER219	
	TXI1	220	0370h	ICU.DTCER220	
SCI2	RXI2	223	037Ch	ICU.DTCER223	
	TXI2	224	0380h	ICU.DTCER224	
SCI3	RXI3	227	038Ch	ICU.DTCER227	
	TXI3	228	0390h	ICU.DTCER228	
SCI5	RXI5	235	03ACh	ICU.DTCER235	
	TXI5	236	03B0h	ICU.DTCER236	
SCI6	RXI6	239	03BCh	ICU.DTCER239	
	TXI6	240	03C0h	ICU.DTCER240	
RIIC0	ICRXI0	247	03DCh	ICU.DTCER247	
	ICTXI0	248	03E0h	ICU.DTCER248	
RIIC1	ICRXI1	251	03ECh	ICU.DTCER251	Low
	ICTXI1	252	03F0h	ICU.DTCER252	

Note: Once the DTC has accepted an activation request, it does not accept further activation requests until transfer for that single request is completed, regardless of the priority order of the requests. However, if multiple activation requests are generated while DMAC or DTC transfer is in progress, the DTC accepts the request with the highest priority when the transfer ends. If multiple activation requests are generated when the DTC module start register (DTCST) value is 0, the DTC accepts the request with the highest priority when the DTC enters the activation-enabled state (by setting DTCST to 1).

16.4 Operation

The DTC transfers data in accordance with the transfer data. Storage of the transfer data in the RAM area is required before DTC operation.

When the DTC is activated, it reads the DTC vector corresponding to the vector number. Then the DTC reads transfer data from the transfer data store address pointed by the DTC vector, transfers data, and then writes back the transfer data after the data transfer. Storing transfer data in the RAM area allows data transfer of arbitrary number of channels.

There are three transfer modes: normal transfer mode, repeat transfer mode, and block transfer mode.

The DTC specifies a transfer source address in SAR and a transfer destination address in DAR. The values of these registers are incremented, decremented, or remain unchanged independently after data transfer.

Table 16.4 lists transfer modes of the DTC.

Table 16.4 Transfer Modes of the DTC

Transfer Mode	Data Size Transferred on a Single Activation Request	Increment/Decrement of Memory Address	Settable Transfer Count
Normal transfer mode	One byte/word/longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536
Repeat transfer mode*1	One byte/word/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 255 bytes/words/longwords)	Incremented/decremented by 1, 2 or, 4 or address fixed	1 to 65536

Note 1. Set transfer source or transfer destination in the repeat area.

Note 2. Set transfer source or transfer destination in the block area.

Note 3. After data transfer of the specified count, the initial state is restored and the operation is continued (repeated).

Setting the CHNE bit in MRB to 1 allows multiple transfers (chain transfer) on a single startup source. Chain transfer is enabled when transfer counter = 0 by setting the CHNS bit in MRB to 1.

Figure 16.4 shows the operation flowchart of the DTC. Table 16.5 shows chain transfer conditions (excluding combinations of the second transfer and the third transfer, and combinations of subsequent transfers).

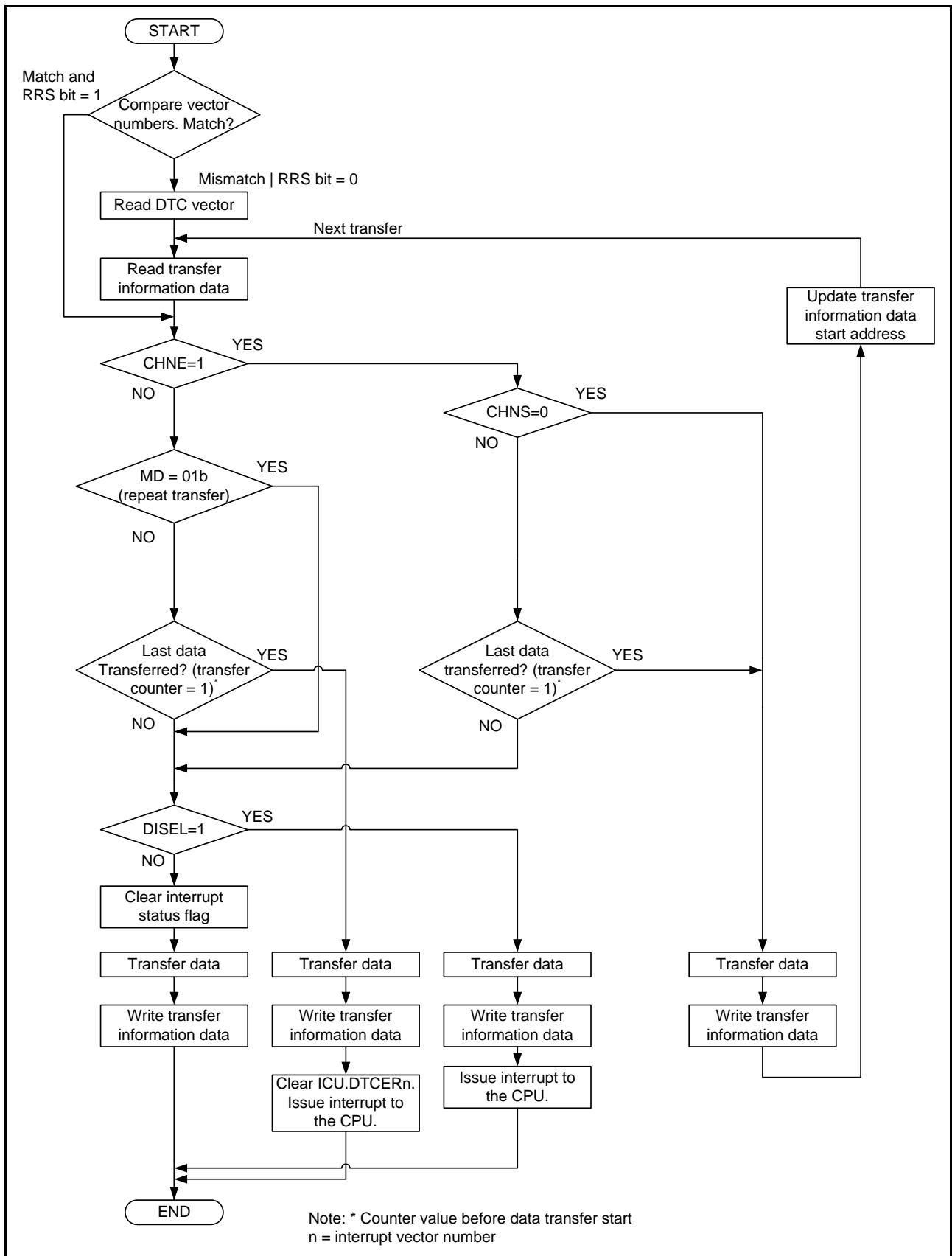


Figure 16.4 Operation Flowchart of the DTC

Table 16.5 Chain Transfer Conditions

First Transfer				Second Transfer*3				DTC Transfer
CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1*2	CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1*2	
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 registers to be used as transfer counters differ according to the transfer modes. Normal transfer mode: CRA, Repeat transfer mode: CRAL, Block transfer mode: CRB

Note 2. On the last data transfer, the counter value changes from 1 to 0 in normal or block transfer mode and changes from 1 to CRAH in repeat transfer mode. "1→*" " indicates both of these operations.

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 Data Read Skip Function

Vector address read and transfer data read can be skipped by the setting of the RRS bit in DTCCR.

When a DTC startup request is generated, the current DTC vector number is always compared with the DTC vector number in the previous startup process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the vector address and transfer data. However, when the previous transfer was chain transfer, the vector address and transfer data are always read. Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, transfer data is always read regardless of the value of RRS bit. Figure 16.13 shows an example of transfer data read skip.

To update the DTC vector table and transfer data, set the RRS bit to 0, update the DTC vector table and transfer data, and then set the RRS bit to 1. When the RRS bit is set to 0, the retained vector number is discarded and the DTC vector table and transfer data that are updated in the following startup process are read.

16.4.2 Transfer Data Write-Back Skip Function

When the SM[1:0] bits in MRA or the DM[1:0] bits in MRB are set to "address fixed", a part of transfer data is not written back. This function is performed independently of the setting of short-address mode or full-address mode. Table 16.6 lists transfer data write-back skip conditions and applicable registers.

The CRA and CRB registers are always written back independently of the setting of short-address mode or full-address mode. Furthermore, in full-address mode, write-back of the MRA and MRB registers are always skipped.

Table 16.6 Transfer Data Write-Back Skip Conditions and Applicable Registers

SM[1:0] Bits in MRA		DM[1:0] Bits in MRB		SAR Register	DAR Register
b3	b2	b3	b2		
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

16.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single startup source. The transfer count can be set to 1 to 65536.

Transfer source addresses and transfer destination addresses can be set to increment, decrement, or fixed independently. This mode enables an interrupt request to the CPU to be generated at the end of specified-count transfer.

Table 16.7 lists register functions in normal transfer mode, and Figure 16.5 shows the memory map of normal transfer mode.

Table 16.7 Register Functions in Normal Transfer Mode

Register	Description	Value Written Back by Writing Transfer Data
SAR	Transfer source address	Increment/decrement/fixed*
DAR	Transfer destination address	Increment/decrement/fixed*
CRA	Transfer counter A	CRA - 1
CRB	Transfer counter B	Not updated

Note: * Write-back operation is skipped in address-fixed mode.

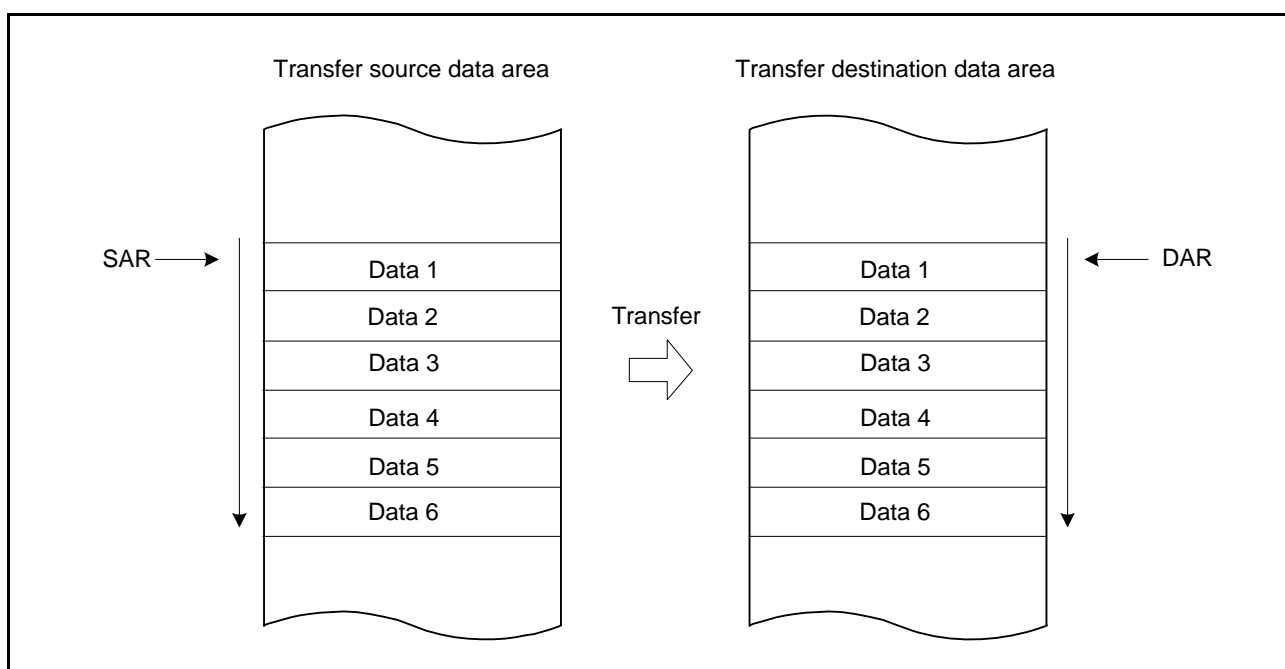


Figure 16.5 Memory Map of Normal Transfer Mode

16.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single startup source.

Specify either transfer source or transfer destination for the repeat area by the DTS bit in MRB. The transfer count can be set to 1 to 256. When the specified-count transfer is completed, the initial value of the address register specified in the transfer counter and the repeat area is restored and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.

When the transfer counter CRAL is decreased to 00h in repeat transfer mode, the CRAL value is updated to the value set in CRAH. Thus the transfer counter does not become 00h, which inhibits generation of interrupt request to the CPU when the DIESEL bit in MRB is set to 0 (an interrupt request to the CPU is generated when specified data transfer is completed).

Table 16.8 lists the register functions in repeat transfer mode, and Figure 16.6 shows the memory map of repeat transfer mode.

Table 16.8 Register Functions in Repeat Transfer Mode

Register	Description	Value Written Back by Writing Transfer Data	
		When CRAL is not 1	When CRAL is 1
SAR	Transfer source address	Increment/decrement/fixe*d*	(When the DTS bit in MRB is 0) Increment/decrement/fixe*d* (When the DTS bit in MRB is 1) SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixe*d*	(When the DTS bit in MRB is 0) DAR register initial value (When the DTS bit in MRB is 1) Increment/decrement/fixe*d*
CRAH	Retains transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note: * Write-back is skipped in address-fixed mode.

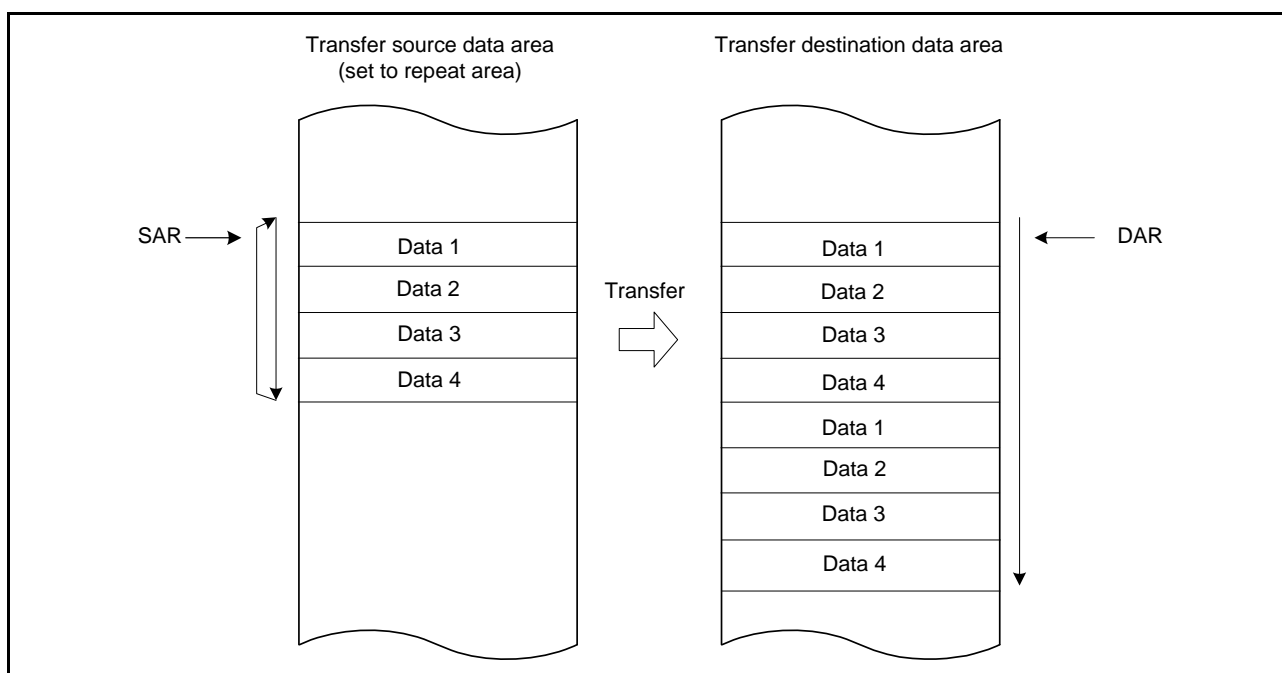


Figure 16.6 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

16.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single startup source.

Specify either transfer source or transfer destination for the block area by the DTS bit in MRB. The block size can be set to 1 to 255 bytes (or 1 to 255 words or 1 to 255 longwords).

When transfer of the specified one block is completed, the initial values of the block size counter CRAL and the address register (SAR when the DTS bit = 1 or DAR when the DTS bit = 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.

The block transfer 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 16.9 lists register functions in block transfer mode, and Figure 16.7 shows the memory map of block transfer mode.

Table 16.9 Register Functions in Block Transfer Mode

Register	Description	Value Written Back by Writing Transfer Data
SAR	Transfer source address	(When DTS bit in MRB is 0) Increment/decrement/fixed* (When DTS bit in MRB is 1) SAR register initial value
DAR	Transfer destination address	(When DTS bit in MRB is 0) DAR register initial value (When DTS bit in MRB is 1) Increment/decrement/fixed*
CRAH	Retains block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note: * Write-back is skipped in address-fixed mode.

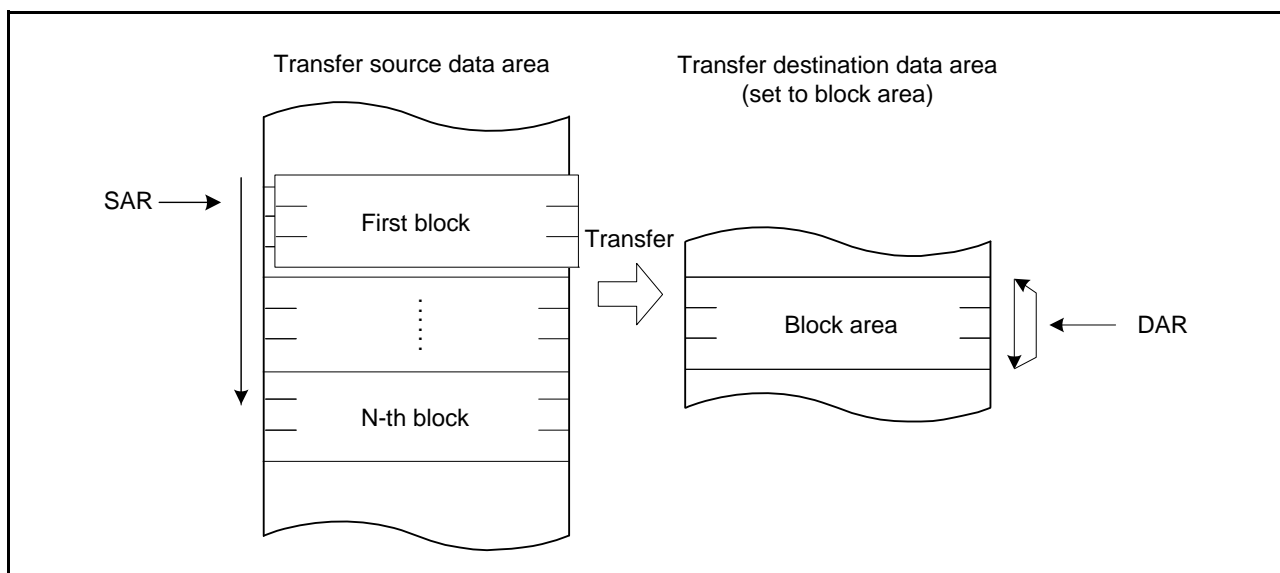


Figure 16.7 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

16.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 enables chained transfer. In chained transfer, multiple unit transfer operations are performed in response to a single request from an activation source.

When the MRB.CHNE bit is 1 and the MRB.CHNS bit is 0, an interrupt request is sent to the CPU on completion of the specified number of unit transfer operations unless the MRB.DISEL bit is 1 (selecting generation of an interrupt for the CPU after every round of data transfer), in which case an interrupt request for the CPU is not generated. Furthermore, the interrupt status flag is not affected when an interrupt signal is being used as an activation source.

Settings for the SAR, DAR, CRA, CRB, MRA, and MRB registers can be individually made.

Figure 16.8 shows operations in chained transfer.

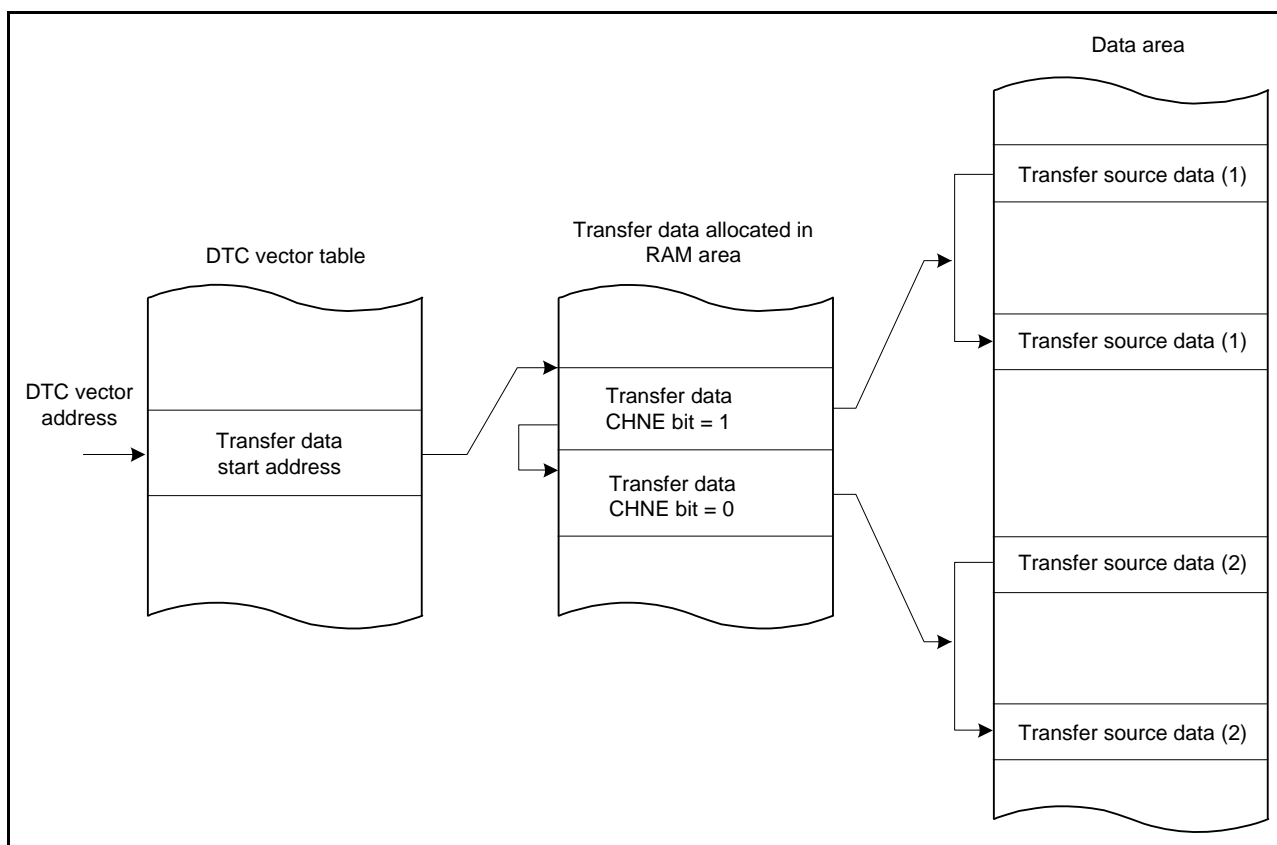


Figure 16.8 Chain Transfer Operation

When both the MRB.CHNE and MRB.CHNS bits are 1, chained transfer only proceeds on completion of the specified number of rounds of data transfer (that is, when the transfer counter becomes 0). Even if the transfer mode is repeat transfer mode, chained transfer only proceeds on completion of the specified number of rounds of data transfer.

See table 16.5 for details of the conditions for chained transfer.

16.4.7 Operation Timing

Figure 16.9 to Figure 16.13 show examples of DTC operation timing.

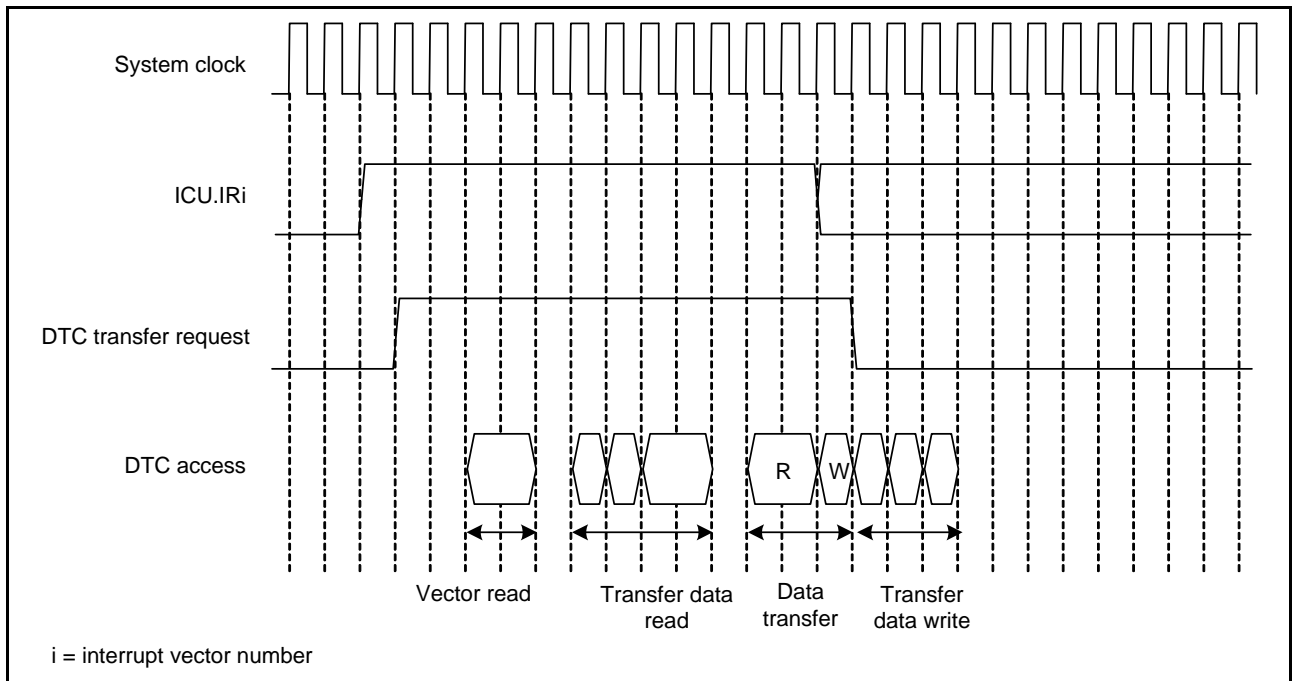


Figure 16.9 Example of DTC Operation Timing 1
 (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

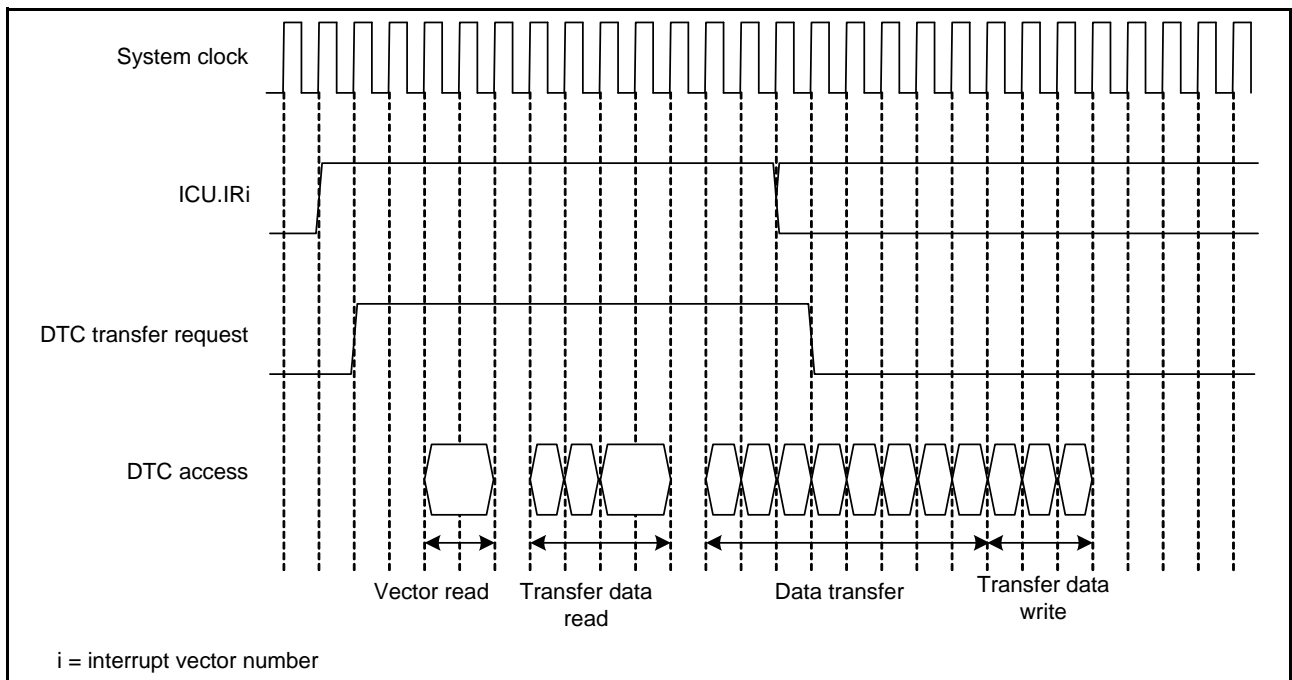


Figure 16.10 Example of DTC Operation Timing 2
 (Short-Address Mode, Block Transfer Mode, Block Size = 4)

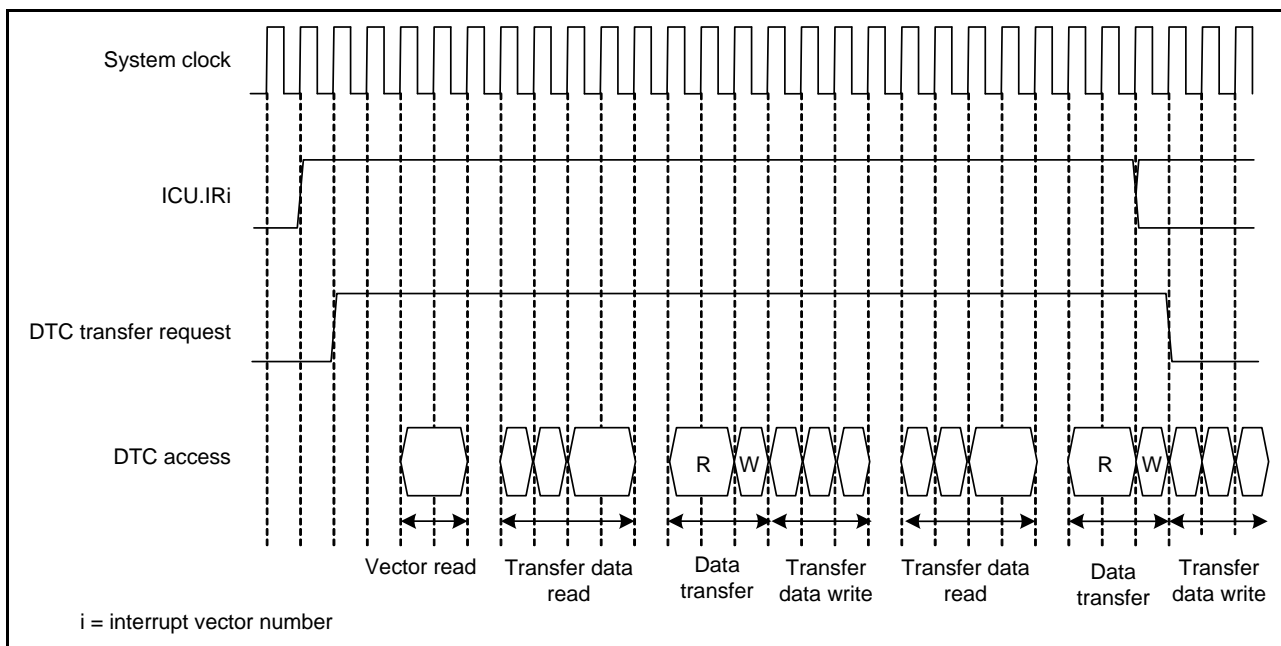


Figure 16.11 Example of DTC Operation Timing 3 (Short-Address Mode, Chain Transfer)

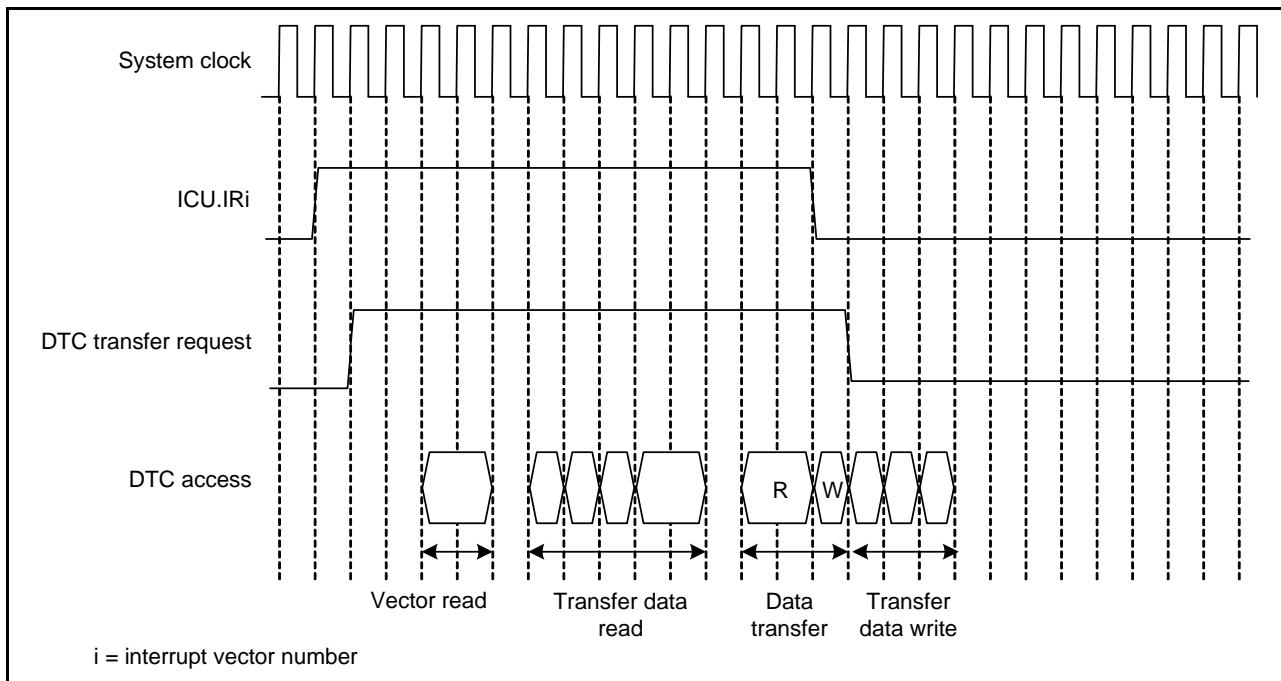


Figure 16.12 Example of DTC Operation Timing 4 (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

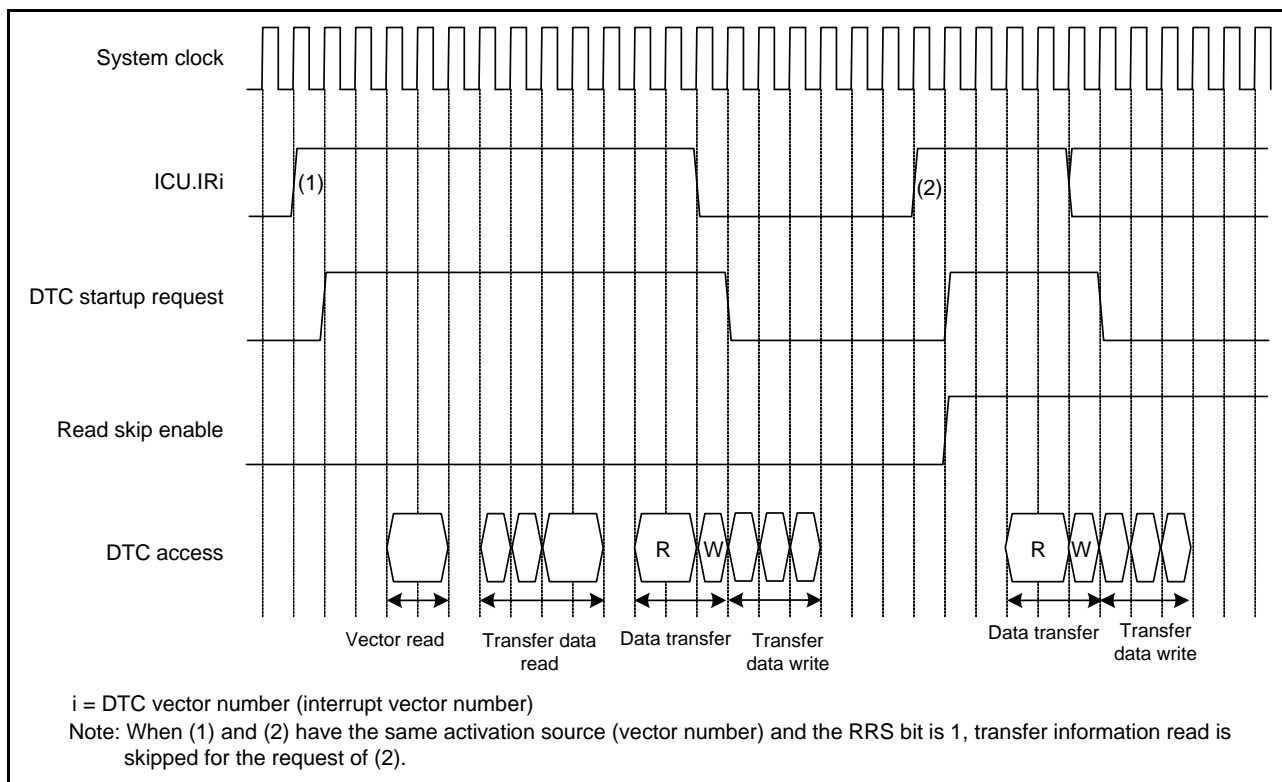


Figure 16.13 Example of Operation when Transfer Information Skip is Executed

16.4.8 Execution Cycle of the DTC

Table 16.10 lists the execution cycle of single data transfer of the DTC.

Table 16.10 Execution Cycle of the DTC

Transfer Mode	Vector Read		Transfer Data Read			Transfer Data Write			Data Read	Data Write	Internal Operation	
	Cv+1	0*1	4 × Ci + 1*2	3 × Ci + 1*3	0*1	3 × Ci*4	2 × Ci*5	Ci*6	Cr + 1	Cw	2	0*1
Normal	Cv+1	0*1	4 × Ci + 1*2	3 × Ci + 1*3	0*1	3 × Ci*4	2 × Ci*5	Ci*6	Cr + 1	Cw	2	0*1
Repeat	Cv+1	0*1	4 × Ci + 1*2	3 × Ci + 1*3	0*1	3 × Ci*4	2 × Ci*5	Ci*6	Cr + 1	Cw	2	0*1
Block*7	Cv+1	0*1	4 × Ci + 1*2	3 × Ci + 1*3	0*1	3 × Ci*4	2 × Ci*5	Ci*6	P × Cr	P × Cw	2	0*1

Note 1. Transfer data skip

Note 2. Full-address mode

Note 3. 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. This indicates the case when the block size is two or more. When the block size is one, the number of cycles is the same as for normal transfer.

[Legend]

P: Block size (set by CRAH and CRAL)

Cv: Access cycles for the vector information data storing destination

Ci: Access cycles for the transfer information data storing destination

Cr: Access cycles for the data read destination

Cw: Access cycles for the data write destination

(The unit of vector read, transfer information data read, "+1" of data transfer read, and "2" of internal operation is system clock cycles (ICLK).)

(Cv, Ci, Cr, and CW values depend on the access destination. For the number of cycles for the access destination, see section 37, RAM, section 38, ROM (Flash Memory for Code Storage), section 5, I/O Registers, and section 12.2.5, External Bus.

For execution timing in each transfer mode, see section 16.4.7, Operation Timing.)

16.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer data read and transfer data write. While transfer data is not read or written, bus arbitration is made according to the priority determined by the bus master arbitrator. For the bus arbitration, see section 12, Buses.

16.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR).

Follow the procedure shown in Figure 16.14 to set the DTC activation source.

Set the DTC module start bit (DTCST.DTCST) to 1.

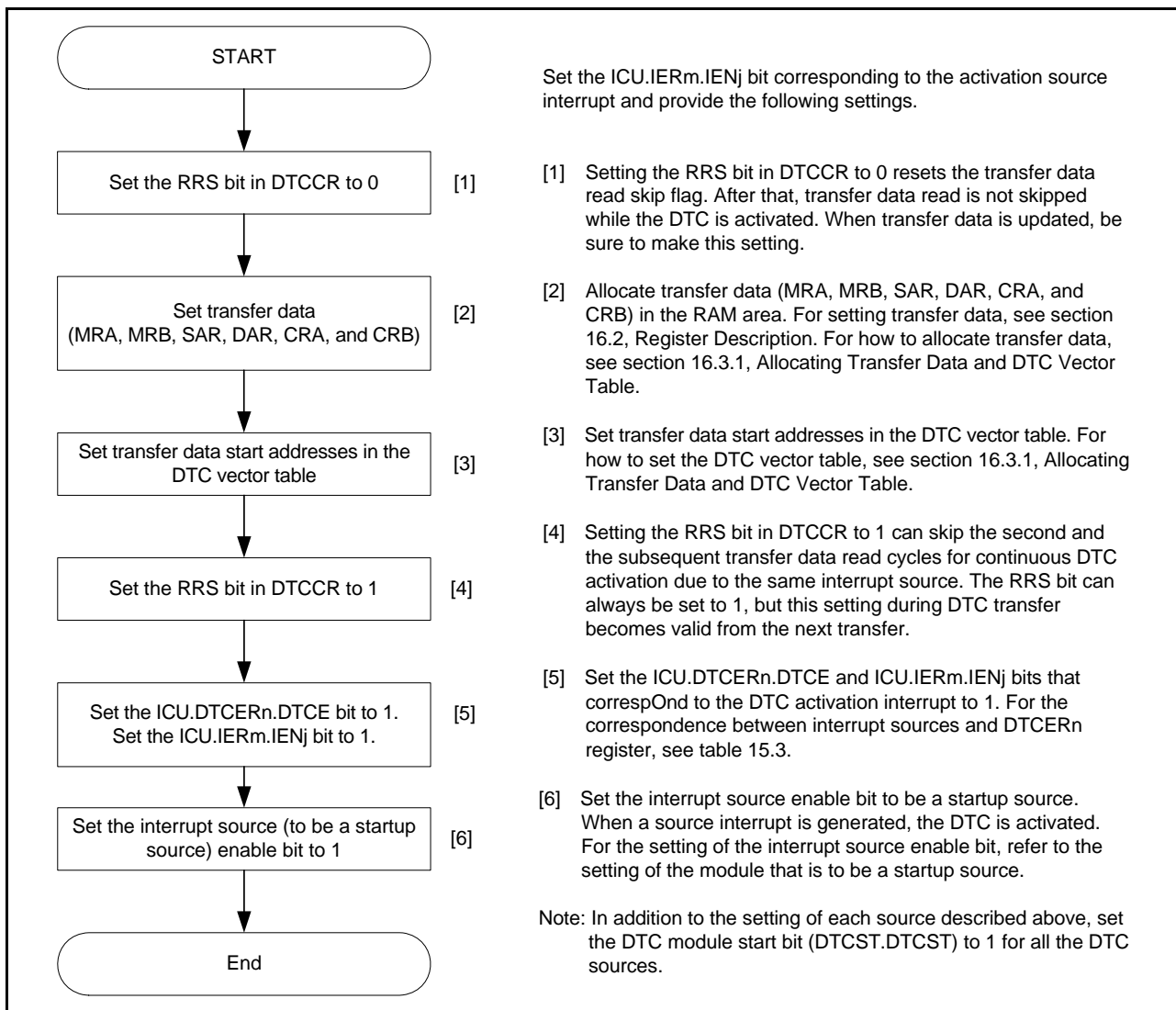


Figure 16.14 Procedure to set the DTC activation source

16.6 Examples of DTC Usage

16.6.1 Normal Transfer

As an example of DTC usage, its employment in the transfer of 128 bytes of data by an SCI is described below.

(1) Transfer Data Set

In the MRA register, make the settings to select a fixed source address (MRA.SM[1:0] = 00b), incrementation of the destination address (MRA.DM[1:0] = 10b), transfer in normal mode (MRA.MD[1:0] = 00b), and byte-sized transfer (MRA.SZ[1:0] = 00b). The MRB.DTS bit can be set to any value. For other bits of the MRB register, make the setting for one interrupt to initiate one round of transfer (MRB.CHNE = "0" and MRB.DISEL = "0"). Set the SAR to the address of the RDR for the given SCIn (n = 0 to 3, 5, 6), the DAR to the first address of the area in RAM where data are to be stored, and the CRA register to 128 (0080h). Set a value of FFFFh to the CRB.

(2) DTC Vector Table

The address where the transfer-control information for use with the RXI starts is set in the vector table for the DTC.

(3) ICU Set and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to "1" and the ICU.IERm.IENj bit to "1".

Set the DTCST.DTCST bit to "1".

(4) SCI Set

Enable reception-completed interrupts by setting the SCR.RIE bit in the given SCIn to "1". Also, so that further reception does not proceed if a reception error occurs while reception by the SCI is in progress, make the CPU able to accept reception-error interrupts.

(5) DTC Transfer

Every time the reception of one 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. Processing for completion is performed in the processing routine for this interrupt.

16.6.2 Chain Transfer

As an example of chained transfer by the DTC, its employment in the output of pulses by a PPG is described below.

Chained transfer is used to transfer pulse output data and vary the period of the output trigger for the PPG. The first half of the chained transfer is in repeated-transfer mode and the destinations for transfer are the PPGn.NDRH and PPGn.NDRL (where n = 0 or 1) registers. The second half of the chained transfer is in normal-transfer mode and the destinations for transfer are the MTUn.TGR registers (where n = 0 to 11). This is because clearing of the activation source and generation of an interrupt on completion of the specified number of rounds of transfer are restricted to the second half of the chained transfer (transfer while MRB.CHNE = "0").

An example of how to use the compare-match interrupt for an MTUn.TGRA register (for n = 0 to 4 or 6 to 10) as an activating source for the DTC is given below.

(1) First Transfer Data Set

Settings are made for transfer to the PPGn.NDRH and PPGn.NDRL registers. In the MRA register, make the settings to select incrementation of the source address (MRA.SM[1:0] = 10b), transfer in repeated-transfer mode (MRA.MD[1:0] = 01b), and word-sized transfer (MRA.SZ[1:0] = 01b). In the MRB register, make the settings for the destination address fixed (MRB.DM[1:0] = 00b) and for chained transfer (MRB.CHNE = "1", MRB.CHNS = "0", and MRB.DISEL = "0"). Set the source side on the repeat area (MRB.DTS = 1). Set the SAR to the first address of the data table, the DAR register to the address of the PPGn.NDRH register, and the CRAH and CRAL registers to the size of the data table. Set a value of FFFFh to the CRB.

(2) Second Transfer Data Set

Settings are made for transfer to the MTUn.TGRA register. In the MRA register, make the settings to select incrementation of the source address (MRA.SM[1:0] = 10b), transfer in normal mode (MRA.MD[1:0] = 00b), and word-sized transfer (MRA.SZ[1:0] = 01b). In the MRB register, make the settings for the destination address fixed (MRB.DM[1:0] = 00b) and for the single data transfer per interrupt (MRB.CHNE = 0, MRB.DISEL = 0). The MRB.DTS bit can be set to any value. Set the SAR register to the first address of the data table, the DAR register to the address of the MTUn.TGRA register, and the CRA register to the size of the data table. Set a value of FFFFh to the CRB.

(3) Transfer Data Assignment

Place the transfer-control information for use in transfer to the MTU immediately after the transfer-control information for use in transfer to the PPGn.NDRH and PPGn.NDRL registers.

(4) DTC Vector Table

In the DTC vector table, set the address where the transfer-control information for use in transfer to the PPGn.NDRH and PPGn.NDRL registers starts.

(5) ICU Set and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to "1" and the ICU.IERm.IENj bit to "1".
Set the DTCST.DTCST bit to "1".

(6) MTU Set

In the given MTUn, set the TIOR register so that the TGRA register operates as an output-compare register (with output disabled) and make the TIER setting to enable TGInA interrupt requests.

(7) PPG Set

Set the default output values in the PPGn.NDRH and PPGn.NDRL registers and the next output values in the PPGn.NDRH and PPGn.NDRL registers. In the PORTn.DDR of the corresponding port n (where n = 0 to 9 or A to G), set the appropriate bit to "1" so that output from the PPGn.NDRH and PPGn.NDRL registers proceeds. Also, select compare-match of the MTU as the output trigger in the PPGn.PCR of the corresponding port n (where n = A to E and G).

(8) MTU Activation

Set the MTUn.TSTR.CST[5:0] (n = A, B) bits to "1" to start counting operation of the MTU.TCNT counter.

(9) DTC Transfer

Every time a compare-match with the MTUn.TGRA register is generated, next output values are transferred to the PPGn.NDRH and PPGn.NDRL registers and the setting for the next output-trigger period is transferred to the MTUn.TGRA register.

(10) Interrupt Handling

After the specified number of rounds of data transfer has been completed (i.e. when the value in the CRA register of the MTU has become "0"), a TGIA interrupt request is issued for the CPU. Processing for completion is performed in the processing routine for this interrupt.

16.6.3 Chain Transfer when Counter = 0

The second data transfer is performed only when the counter = 0. Repeat transfer of a transfer count of 256 or more is enabled by the re-setting for the first data transfer.

The following shows an example of configuring a 128-kbyte input buffer, where the input buffer is set so that its lower address starts with 0000h. Figure 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, CRA = 0000h (65,536 times), CHNE bit = 1 (chain transfer enabled) in MRB, CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0) in MRB, and DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed) in MRB.
2. Prepare the upper 8-bit address of the start address at every 65,536 times of the transfer destination address for the first data transfer in another area (such as ROM). For example, when setting the input buffer to 200000h to 21FFFFh, prepare 21h and 20h.
3. For the second data transfer, set repeat transfer mode (source side: repeat area) for re-setting the transfer destination address of the first data transfer. Specify the upper 8 bits of DAR in the first transfer data area for the transfer destination. At this time, set CHNE bit = 0 (chain transfer disabled) in MRB and DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed) in MRB. When setting the input buffer mentioned above to 200000h to 21FFFFh, set the transfer counter to 2.
4. The first data transfer is performed by an interrupt 65,536 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 21h. The transfer counter (lower 16 bits) of the transfer destination address of the first data transfer is 0000h.
5. In succession, the first data transfer is performed by an interrupt 65,536 times specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 20h. The transfer counter (lower 16 bits) of the transfer destination address of the first data transfer is 0000h.
6. Steps 4 and 5 above are repeated infinitely. Since the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated

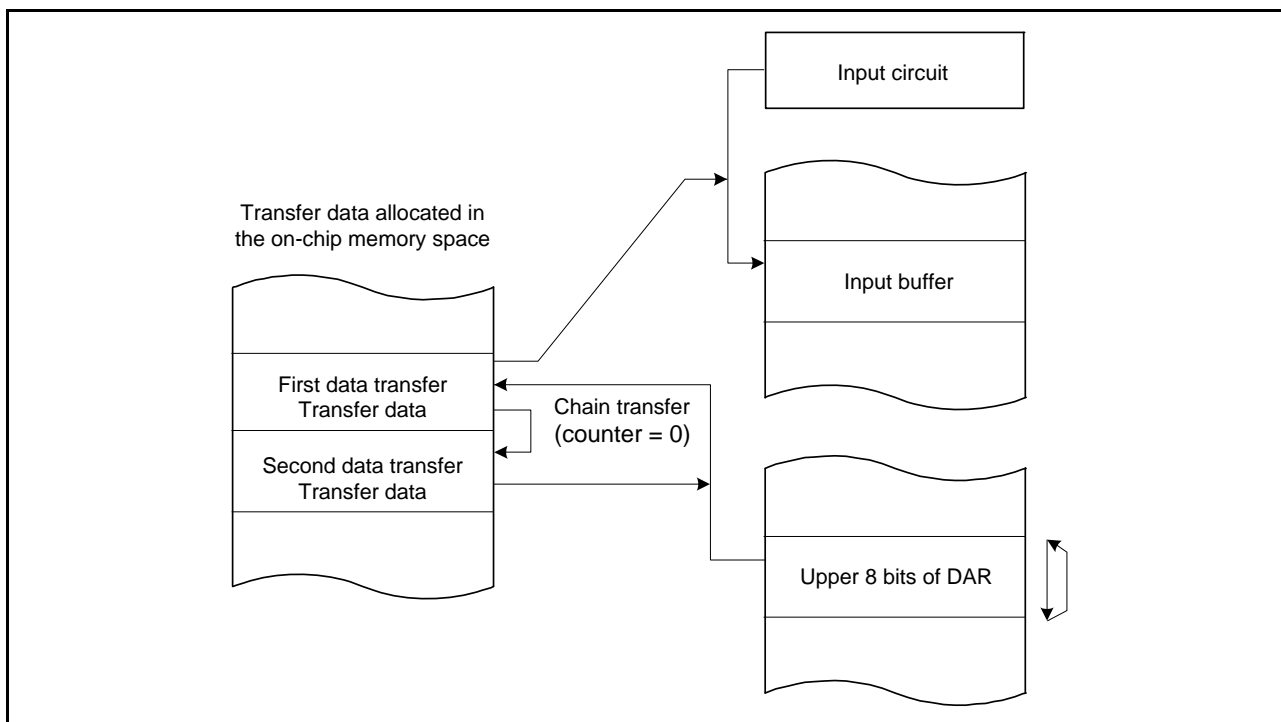


Figure 16.15 Chain Transfer when Counter = 0

16.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the DISEL bit in MRB set to 1 (an interrupt request to the CPU is generated each time DTC data transfer is performed) has been completed, an interrupt to the CPU is generated by the DTC startup source. Such interrupts to the CPU are controlled according to the PSWI bit (interrupt enable) of the CPU, the PSW.IPL[3:0] bits (processor interrupt priority level), and the priority level of the interrupt control unit.

16.8 Low-Power Consumption Function

To place the DTC in the module-stop state, all-module clock stop mode, software-standby mode, or deep software-standby mode, clear the DTCST.DTCST bit to 0 (DTC module stopped), and then perform the following processing.

(1) Module Stop Function

Writing 1 to the MSTPCRA.MSTPA28 bit (transition to the DMACA and DTC module-stop state) enables the module-stop function of the DTC. If DTC transfer is in progress at the time a 1 is written to the MSTPA28 bit, the transition to the module-stop state proceeds after DTC transfer has ended. Writing 0 to the MSTPA28 bit releases the DTC from the module-stop state.

(2) All-Module Clock Stop Mode

Writing 1 to the MSTPCRA.ACSE bit (all-module clock stop mode enabled), writing 1 to all the bits in MSTPCRA and MSTPCRB, including the MSTPA28 bit (transition to the DMACA and DTC module-stop state), and executing a WAIT instruction causes a transition to the all-module clock stop mode. If DTC transfer is in progress at the time the WAIT instruction is executed, the DTC can enter all-module clock stop mode after completion of the current DTC transfer.

After the DTC returns from all-module clock stop mode, writing 0 to the MSTPA28 bit releases the DTC from the module-stop state.

(3) Software Standby and Deep Software Standby Modes

Writing 1 to the SBYCR.SSBY bit (transition to software standby mode after WAIT instruction execution) and 0 to the DPSBYCR.DPSBY bit (transition to software standby mode after WAIT instruction execution), executing a WAIT instruction places the DTC in software standby mode.

If DTC transfer is in progress at the time the WAIT instruction is executed, the DTC enters software standby mode after completion of the current DTC transfer.

The DTC enters deep software standby mode when the DPSBYCR.DPSBY bit is set to 1 (transition to deep software standby mode after WAIT instruction execution).

(4) Notes on Low-Power Consumption Function

For the timing of WAIT instruction execution and register settings, see section 9.7.7, Timing of Wait Instructions.

To perform DTC transfer after returning from low-power consumption mode, set the DTCST.DTCST bit to 1 again.

16.9 Usage Notes

16.9.1 Transfer Information Data Start Address

Be sure to set multiples of 4 for the transfer information data start addresses in the DTC vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

16.9.2 Allocating Transfer Data

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.

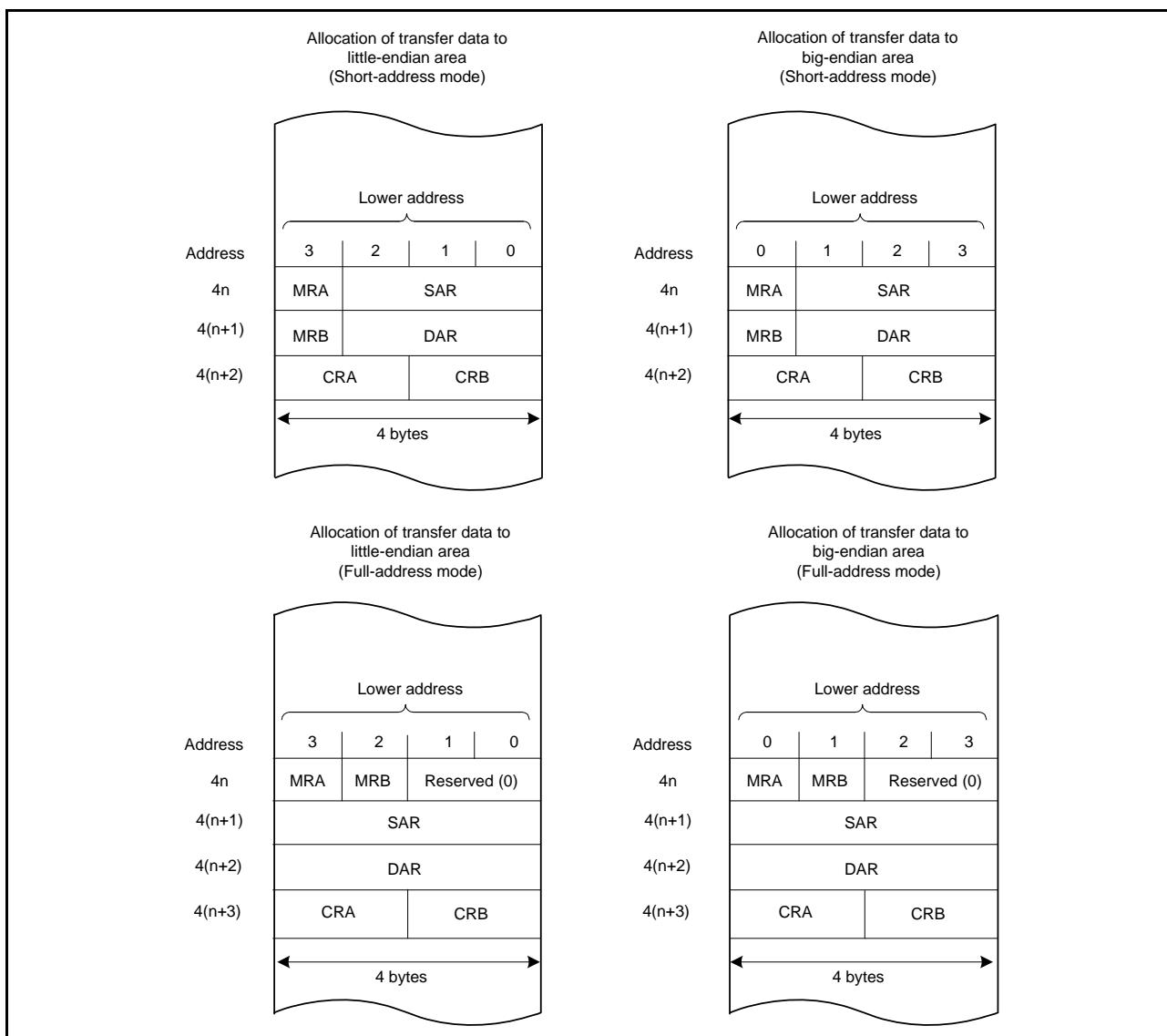


Figure 16.16 Allocation of Transfer Data

16.9.3 Setting the DTC Activation Enable Register (ICU.DTCERi) of the Interrupt Control Unit

While the DTCST.DTCST bit is 0 (DTC module stop), set the DTC activation enable register (ICU.DTCERi (i = interrupt vector number)). Moreover, the DMACA should not be activated by setting the DMACA activation request select register (ICU.DMRSRn (n = number of DMACA channel)) to the same vector number that has been specified by setting the ICU.DTCERi register 1 (DTC transfer enable). For details on the ICU.DTCERi and ICU.DMRSRn registers (n = number of DMACA channel), refer to section 11, Interrupt Control Unit (ICUa).

16.9.4 Selecting Communication Function Interrupt as DTC Activation Source

In the RX62N and RX621 Groups, caution should be used when the communication function (SCI, RIIC, RSPI, USB) is used in combination with the DTC or DMACA function. For details, see section 11.7, Usage Notes.

17. I/O Ports

The I/O ports of the RX62N/RX621 Group function as a programmable I/O port, an I/O pin of a peripheral module, an input pin for an interrupt, or a bus control pin.

Each pin is also configurable as an I/O pin of a peripheral module or an input pin for an interrupt. All pins function as input pins immediately after a reset, and pin functions are switched by register settings. The setting of each pin is specified by the registers for the corresponding I/O port and peripheral modules.

Each port has data direction registers (DDR) that control input and output, data registers (DR) that store data for output, port registers (PORT) for reading the pin states, and input buffer control registers (ICR) that enable or disable the input buffer.

As the configuration of the I/O ports differs with the package, the I/O ports for the respective packages are described separately in this section.

For the 176-pin LFBGA version, see section 17.1, I/O Ports [for 176-Pin LFBGA].

For the 145-pin TFLGA and 144-pin LQFP versions, see section 17.2, I/O Ports [for 145-Pin TFLGA/144-Pin LQFP].

For the 100-pin LQFP version, see section 17.3, I/O Ports [for 100-Pin LQFP].

For the 85-pin TFLGA version, see section 17.4, I/O Ports [for 85-Pin TFLGA].

17.1 I/O Ports [for 176-Pin LFBGA]

The RX62N/RX621 Group (176-Pin LFBGA) has 17 I/O ports (ports 0 to 9 and A to G), which handle 128 I/O pins.

17.1.1 Overview

Table 17.1 gives the specifications of the I/O ports and Table 17.2 lists I/O ports and pin functions.

Table 17.1 Specifications of I/O Ports (176-Pin LFBGA)

Item	Description
I/O pins	126
Input pins	2
Number of ports	17 (0 to 9 and A to G)
Built-in input pull-up resistor	Ports 9, A, B, C, D, E, G
Open drain outputs	Ports 0, 1, 2, 3 (P30 to P34), C
5-V tolerance	Port 0 (P00, P01, P02, P07), port 1 (P12, P13, P16, P17), port 2 (P20, P21), port 3 (P33)
Schmitt trigger input pins	All port inputs, CAN inputs, USB inputs, IRQ inputs, MTU inputs, POE inputs, TMR inputs, RIIC inputs, SCI inputs, and A/D trigger inputs
Others	<ul style="list-style-type: none"> • Each pin is capable of driving a capacitive load of 30 pF in the case of a TTL load. • When configured as an output, a pin is capable of driving a Darlington transistor.

Table 17.2 Port Functions (176-Pin LFBGA) (1 / 6)

Port	Description	Bit	Function			CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
			I/O	Input	Output				
Port 0	General I/O port pins, interrupt inputs, TMR inputs, SCI I/O signals, A/D converter inputs, and D/A converter outputs	0	P00	TMRI0-A/IRQ8-A	TxD6-A	—	All input functions	—	√
		1	P01	TMCI0-A/RxD6-A/IRQ9-A			All input functions		
		2	P02/SCK6-A	TMCI1-A/IRQ10-A			All input functions		
		3	P03	IRQ11-A	DA0		All input functions		
		5	P05	IRQ13-A	DA1		All input functions		
		7	P07	ADTRG0#-A/IRQ15-A			All input functions		
Port 1	General I/O port pins, USB I/O signals, MTU I/O signals, TMR I/O signals, interrupt inputs, SCI I/O signals, RIIC I/O signals, PPG I/O signals, and A/D converter inputs	0	P10	MTIC5W-A/TMRI3-A/IRQ0-B	USB1_DPUPE-A	—	All input functions	—	√
		1	P11/SCK2-A	MTIC5V-A /TMCI3-A /IRQ1-B	USB1_VBUSEN-A		All input functions		
		2	P12/SCL0	MTIC5U-A/TMCI1-B/RxD2-A/IRQ2-B			All input functions		
		3	P13 /SDA0	ADTRG1#/IRQ3-B	TMO3/TxD2-A		All input functions		
		4	P14	USB0_OVRCURA/TMRI2/IRQ4-B	USB0_DPUPE-B		All input functions		
		5	P15/MTIOC0B/SCK3-A	USB1_OVRCURA/TMCI2-A/IRQ5-B	PO13/USB1_DPUPE-B		All input functions		
		6	P16/MTIOC3C-A	USB0_VBUS/USB0_OVRCURB/RxD3-A/IRQ6-B	TMO2/PO14/USB0_VBUSEN-B		All input functions		
		7	P17/MTIOC3A	USB1_VBUS/USB1_OVRCURB/IRQ7-B	TxD3-A/PO15/USB1_VBUSEN-B		All input functions		

Table 17.2 Port Functions (176-Pin LFBGA) (2 / 6)

Port	Description	Bit	Function			CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
			I/O	Input	Output				
Port 2	General I/O port pins, bus control I/O signals, EXDMAC I/O signals, USB I/O signals, RSPI I/O signals, MTU I/O signals, PPG outputs, TMR I/O signals, SCI I/O signals, RIIC I/O signals, and A/D converter inputs	0	P20/MTIOC1A/SDA1	USB0_ID/ TMRI0-B	PO0/TxD0	—	All input functions	—	√
		1	P21/MTIOC1B/SCL1	TMCI0-B/RxD0	USB0_EXICEN/ PO1	—	All input functions	—	—
		2	P22/MTIOC3B-A/ SCK0	EDREQ0-B/ MTCLKC-A	USB0_DRPD/ PO2/ TMO0	EDRE Q0-B	P22, MTIOC3B-A SCK0, MTCLKC-A	—	—
		3	P23/MTIOC3D-A	MTCLKD-A	EDACK0-B/ USB0_DPUPE-A/ TxD3-B/PO3	—	All input functions	—	—
		4	P24/MTIOC4A-A/ SCK3-B	EDREQ1-B/ MTCLKA-A/TMRI1	CS4#-C/ USB0_VBUSEN-A/ PO4	EDRE Q1-B	P24, MTIOC4A-A, SCK3-B, MTCLKA-A, TMRI1	—	—
		5	P25/MTIOC4C-A	MTCLKB-A/ ADTRG0#-B/ RxD3-B	CS5#-C/ EDACK1-B/ USB0_DPRPD/ PO5	—	All input functions	—	—
		6	P26/MOSIB-A/ MTIOC2A	USB1_ID	CS6#-C/PO6/ TMO1/TxD1-A	MOSIB -A	P26, MTIOC2A, USB1_ID	—	—
Port 3	General I/O port pins, CAN I/O signals, USB outputs, RSPI I/O signals, MTU I/O signals, TMR inputs, SCI I/O signals, interrupt inputs, PPG outputs, and RTC outputs	7	P27/ RSPCKB-A/ MTIOC2B/ SCK1-A	—	CS7#-C/ USB1_EXICEN/ PO7	RSPC KB-A	P27, MTIOC2B, SCK1-A	—	—
		0	P30/MISOB-A/ MTIOC4B-A	TMRI3-B/RxD1-A/ IRQ0-A	USB1_DRPD/ PO8	MISOB -A	P30, MTIOC4B-A TMRI3-B, RxD1-A, IRQ0-A	—	√
		1	P31/SSLB0-A/ MTIOC4D-A	TMCI2-B/IRQ1-A	USB1_DPRPD/ PO9	SSLB0 -A	P31, MTIOC4D-A TMCI2-B, IRQ1-A	—	√
		2	P32/MTIOC0C	IRQ2-A	CTX0/TxD6-B/ PO10/RTCOUT	—	All input functions	—	√
		3	P33/MTIOC0D	CRX0/RxD6-B/ IRQ3-A	PO11	—	All input functions	—	√
		4	P34/MTIOC0A/ SCK6-B	TMCI3-B/IRQ4-A	PO12	—	All input functions	—	√
Port 4	General I/O port pins, interrupt inputs, and A/D converter inputs	5	—	P35/NMI	—	—	All input functions	—	—
		0	P40	AN0/IRQ8-B	—	—	P40, IRQ8-B	—	—
		1	P41	AN1/IRQ9-B	—	—	P41, IRQ9-B	—	—
		2	P42	AN2/IRQ10-B	—	—	P42, IRQ10-B	—	—
		3	P43	AN3/IRQ11-B	—	—	P43, IRQ11-B	—	—
		4	P44	AN4/IRQ12	—	—	P44, IRQ12	—	—
		5	P45	AN5/IRQ13-B	—	—	P45, IRQ13-B	—	—
		6	P46	AN6/IRQ14	—	—	P46, IRQ14	—	—
7	P47	AN7/IRQ15-B	—	—	P47, IRQ15-B	—	—		

Table 17.2 Port Functions (176-Pin LFBGA) (3 / 6)

Port	Description	Function				CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
		Bit	I/O	Input	Output				
Port 5	General I/O port pins, external bus clock outputs, bus control I/O signals, USB outputs, EXDMAC I/O signals, RSPI outputs, Ether I/O signals, MTU I/O signals, and SCI I/O signals	0	P50		WR0#/WR#/SSLB1-A/TxD2-B	—	All input functions	—	—
		1	P51/SCK2-B	WAIT#-D	WR1#/BC1#/SSLB2-A	WAIT#-D	P51, SCK2—B		
		2	P52	RxD2-B	RD#/SSLB3-A	—	All input functions		
		3		P53	BCLK	—	All input functions		
		4	P54/MTIOC4B-B	ET_LINKSTA	EDACK0-C	ET_LI NKSTA	P54, MTIOC4B-B		
		5	P55/MTIOC4D-B	WAIT#-B/EDREQ0-C	ET_EXOUT	WAIT#-B, EDREQ0-C	P55, MTIOC4D-B		
		6	P56/MTIOC3C-B		WR2#/BC2#/EDACK1-C	—	All input functions		
Port 6	General I/O port pins, SDRAM outputs, and bus control outputs	7	P57	WAIT#-A/EDREQ1-C	WR3#/BC3#	WAIT#-A, EDREQ1-C	P57		
		0	P60		CS0#-A	—	All input functions	—	—
		1	P61		CS1#-A/SDCS#		All input functions		
		2	P62		CS2#-A/RAS#		All input functions		
		3	P63		CS3#-A/CAS#		All input functions		
		4	P64		CS4#-A/WE#		All input functions		
		5	P65		CS5#-A/CKE		All input functions		
Port 7	General I/O port pins, SDRAM outputs, bus control outputs, and Ether I/O signals	6	P66		CS6#-A/DQM0		All input functions		
		7	P67		CS7#-A/DQM1		All input functions		
		0	P70		SDCLK	—	All input functions	—	—
		1	P71/ET_MDIO		CS1#-B	ET_MDIO	P71		
		2	P72		CS2#-B/ET_MDC	—	All input functions		
		3	P73		CS3#-B/ET_WOL		All input functions		
		4	P74	ET_ERXD1/RMII_RXD1	CS4#-B	ET_ERXD1/RMII_RXD1	P74		
5	P75	ET_ERXD0/RMII_RXD0	CS5#-B	ET_ERXD0/RMII_RXD0	P75				
6	P76	ET_RX_CLK/REF50CK	CS6#-B	ET_RX_CLK/REF50CK	P76				
7	P77	ET_RX_ER/RMII_RX_ER	CS7#-B	ET_RX_ER/RMII_RX_ER	P77				

Table 17.2 Port Functions (176-Pin LFBGA) (4 / 6)

Port	Description	Bit	Function				CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
			I/O	Input	Output					
Port 8	General I/O port pins, EXDMAC I/O signals, MTU I/O signals, and Ether I/O signals	0	P80/MTIOC3B-B	EDREQ0-A	ET_TX_EN/ RMII_TXD_EN	EDREQ0-A	P80, MTIOC3B-B	—	—	
		1	P81/MTIOC3D-B		EDACK0-A/ ET_ETXD0/ RMII_TXD0	—	All input functions			
		2	P82/MTIOC4A-B	EDREQ1-A	ET_ETXD1/ RMII_TXD1	EDREQ1-A	P82, MTIOC4A-B			
		3	P83/MTIOC4C-B	ET_CRS/ RMII_CRS_DV	EDACK1-A	ET_CRS, RMII_CRS_DV	P83, MTIOC4C-B			
		4	P84			—	All input functions			
		5	P85				All input functions			
Port 9	General I/O port pins, address outputs, and bidirectional data-bus lines	0	P90/D16		A16-B	D16	P90	√	—	
		1	P91/D17		A17-B	D17	P91			
		2	P92/D18		A18-B	D18	P92			
		3	P93/D19		A19-B	D19	P93			
		4	P94/D20		A20-B	D20	P94			
		5	P95/D21		A21-B	D21	P95			
		6	P96/D22		A22-B	D22	P96			
		7	P97/D23		A23-B	D23	P97			
Port A	General I/O port pins, address outputs, SDRAM outputs, bus control I/O signals, RSPI I/O signals, MTU I/O signals, and PPG outputs	0	PA0/MTIOC6A		A0/BC0#/DQM2/ SSLA1-B/PO16	—	All input functions	√	—	
		1	PA1/MTIOC6B		A1/DQM3/ SSLA2-B/PO17		All input functions			
		2	PA2/MTIOC6C		A2/SSLA3-B/ PO18		All input functions			
		3	PA3/MTIOC6D		A3/PO19		All input functions			
		4	PA4/SSLA0-B/ MTIOC7A		A4/PO20	SSLA0-B	PA4, MTIOC7A			
		5	PA5/RSPCKA-B/ MTIOC7B		A5/PO21	RSPCKA-B	PA5, MTIOC7B			
		6	PA6/MOSIA-B/ MTIOC8A		A6/PO22	MOSIA-B	PA6, MTIOC8A			
		7	PA7/MISOA-B/ MTIOC8B		A7/PO23	MISOA-B	PA7, MTIOC8B			
Port B	General I/O port pins, address outputs, MTU I/O signals, and PPG outputs	0	PB0/MTIOC9A		A8/PO24	—	All input functions	√	—	
		1	PB1/MTIOC9C		A9/PO25		All input functions			
		2	PB2/MTIOC9B	MTCLKG-B	A10/PO26		All input functions			
		3	PB3/MTIOC9D	MTCLKH-B	A11/PO27		All input functions			
		4	PB4/ MTIOC10A	MTCLKE-B	A12/PO28		All input functions			
		5	PB5/ MTIOC10C	MTCLKF-B	A13/PO29		All input functions			
		6	PB6/ MTIOC10B		A14/PO30		All input functions			
		7	PB7/ MTIOC10D		A15/PO31		All input functions			

Table 17.2 Port Functions (176-Pin LFBGA) (5 / 6)

Port	Description	Bit	Function				CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
			I/O	Input	Output					
Port C	General I/O port pins, address outputs, bus control outputs, RSPI I/O signals, Ether I/O signals, MTU inputs, and SCI I/O signals	0	PC0	ET_ERXD3/ MTCLKG-A	A16-A/SSLA1-A	ET_ERXD3	PC0, MTCLKG-A	√	√	
		1	PC1/SCK5	ET_ERXD2/ MTCLKH-A	A17-A/SSLA2-A	ET_ERXD2	PC1, SCK5, MTCLKH-A			
		2	PC2	ET_RX_DV/ MTCLKE-A/RxD5	A18-A/SSLA3-A	ET_RX_DV	PC2, MTCLKE-A, RxD5			
		3	PC3	ET_TX_ER/ MTCLKF-A	A19-A/TxD5	ET_TX_ER	PC3, MTCLKF-A			
		4	PC4/SSLA0-A	MTCLKC-B/ ET_TX_CLK	A20-A/CS3#-C	SSLA0-A, ET_TX_CLK	PC4, MTCLKC-B			
		5	PC5/ RSPCKA-A	WAIT#-C/ MTIC11W-A/ MTCLKD-B	A21-A/CS2#-C/ ET_ETXD2	RSPCKA-A, WAIT#-C	PC5, MTIC11W-A, MTCLKD-B			
		6	PC6/MOSIA-A	MTIC11V-A/ MTCLKA-B	A22-A/CS1#-C/ ET_ETXD3	MOSIA-A	PC6, MTIC11V-A MTCLKA-B			
		7	PC7/MISOA-A	ET_COL/ MTIC11U-A/ MTCLKB-B	A23-A/CS0#-B	MISOA-A, ET_COL	PC7, MTIC11U-A MTCLKB-B			
Port D	General I/O port pins, bidirectional data-bus lines, MTU inputs, and POE inputs	0	PD0/D0	POE7#		D0	PD0, POE7#	√	—	
		1	PD1/D1	POE6#		D1	PD1, POE6#			
		2	PD2/D2	MTIC11W-B/ POE5#		D2	PD2, MTIC11W-B, POE5#			
		3	PD3/D3	MTIC11V-B/ POE4#		D3	PD3, MTIC11V-B, POE4#			
		4	PD4/D4	MTIC11U-B/ POE3#		D4	PD4, MTIC11U-B, POE3#			
		5	PD5/D5	MTIC5W-B/ POE2#		D5	PD5, MTIC5W-B, POE2#			
		6	PD6/D6	MTIC5V-B/ POE1#		D6	PD6, MTIC5V-B, POE1#			
		7	PD7/D7	MTIC5U-B/ POE0#		D7	PD7, MTIC5U-B, POE0#			
Port E	General I/O port pins, bidirectional data-bus lines, RSPI I/O signals, interrupt inputs, and POE inputs	0	PE0/D8		SSLB1-B	D8	PE0	√	—	
		1	PE1/D9		SSLB2-B	D9	PE1			
		2	PE2/D10	POE9#		SSLB3-B	D10	PE2, POE9#		
		3	PE3/D11	POE8#			D11	PE3, POE8#		
		4	PE4/D12/ SSLB0-B				D12, SSLB0-B	PE4		
		5	PE5/D13/ RSPCKB-B	IRQ5-A			D13, RSPCKB-B	PE5, IRQ5-A		
		6	PE6/D14/ MOSIB-B	IRQ6-A			D14, MOSIB-B	PE6, IRQ6-A		
		7	PE7/D15/ MISOB-B	IRQ7-A			D15, MISOB-B	PE7, IRQ7-A		
Port F	General I/O port pins, on-chip emulator I/O signals, and SCI I/O signals	0	PF0		TDO/TxD1-B	—	All input functions	—	—	
		1	PF1/SCK1-B	TCK	—	TCK	PF1, SCK1-B			
		2	PF2	TDI/RxD1-B		—	All input functions			
		3	PF3	TMS		—	All input functions			
		4	PF4	TRST#		—	All input functions			

Table 17.2 Port Functions (176-Pin LFBGA) (6 / 6)

Port	Description	Bit	Function			CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
			I/O	Input	Output				
Port G	General I/O port pins, tracing outputs, and bidirectional data-bus lines	0	PG0/D24			D24	PG0	√	—
		1	PG1/D25			D25	PG1		
		2	PG2/D26		TRDATA0	D26	PG2		
		3	PG3/D27		TRDATA1	D27	PG3		
		4	PG4/D28		TRSYNC	D28	PG4		
		5	PG5/D29		TRCLK	D29	PG5		
		6	PG6/D30		TRDATA2	D30	PG6		
		7	PG7/D31		TRDATA3	D31	PG7		

17.1.2 Register Descriptions

Table 17.3 lists registers of I/O ports, and Table 17.4 lists valid bits in each register.

Table 17.3 Registers of I/O Ports (176-Pin LFBGA) (1 / 3)

Port Symbol	Register Name	Register Symbol	Value after Reset	Address	Access Size
PORT0	Data direction register	DDR	00h	0008 C000h	8
	Data register	DR	00h	0008 C020h	8
	Port register	PORT	Undefined	0008 C040h	8
	Input buffer control register	ICR	00h	0008 C060h	8
	Open drain control register	ODR	00h	0008 C080h	8
PORT1	Data direction register	DDR	00h	0008 C001h	8
	Data register	DR	00h	0008 C021h	8
	Port register	PORT	Undefined	0008 C041h	8
	Input buffer control register	ICR	00h	0008 C061h	8
	Open drain control register	ODR	00h	0008 C081h	8
PORT2	Data direction register	DDR	00h	0008 C002h	8
	Data register	DR	00h	0008 C022h	8
	Port register	PORT	Undefined	0008 C042h	8
	Input buffer control register	ICR	00h	0008 C062h	8
	Open drain control register	ODR	00h	0008 C082h	8
PORT3	Data direction register	DDR	00h	0008 C003h	8
	Data register	DR	00h	0008 C023h	8
	Port register	PORT	Undefined	0008 C043h	8
	Input buffer control register	ICR	00h	0008 C063h	8
	Open drain control register	ODR	00h	0008 C083h	8
PORT4	Data direction register	DDR	00h	0008 C004h	8
	Data register	DR	00h	0008 C024h	8
	Port register	PORT	Undefined	0008 C044h	8
	Input buffer control register	ICR	00h	0008 C064h	8
PORT5	Data direction register	DDR	00h	0008 C005h	8
	Data register	DR	00h	0008 C025h	8
	Port register	PORT	Undefined	0008 C045h	8
	Input buffer control register	ICR	00h	0008 C065h	8
PORT6	Data direction register	DDR	00h	0008 C006h	8
	Data register	DR	00h	0008 C026h	8
	Port register	PORT	Undefined	0008 C046h	8
	Input buffer control register	ICR	00h	0008 C066h	8
PORT7	Data direction register	DDR	00h	0008 C007h	8
	Data register	DR	00h	0008 C027h	8
	Port register	PORT	Undefined	0008 C047h	8
	Input buffer control register	ICR	00h	0008 C067h	8
PORT8	Data direction register	DDR	00h	0008 C008h	8
	Data register	DR	00h	0008 C028h	8
	Port register	PORT	Undefined	0008 C048h	8
	Input buffer control register	ICR	00h	0008 C068h	8

Table 17.3 Registers of I/O Ports (176-Pin LFBGA) (2 / 3)

Port Symbol	Register Name	Register Symbol	Value after Reset	Address	Access Size
PORT9	Data direction register	DDR	00h	0008 C009h	8
	Data register	DR	00h	0008 C029h	8
	Port register	PORT	Undefined	0008 C049h	8
	Input buffer control register	ICR	00h	0008 C069h	8
	Pull-up resistor control register	PCR	00h	0008 C0C9h	8
PORTA	Data direction register	DDR	00h	0008 C00Ah	8
	Data register	DR	00h	0008 C02Ah	8
	Port register	PORT	Undefined	0008 C04Ah	8
	Input buffer control register	ICR	00h	0008 C06Ah	8
	Pull-up resistor control register	PCR	00h	0008 C0CAh	8
PORTB	Data direction register	DDR	00h	0008 C00Bh	8
	Data register	DR	00h	0008 C02Bh	8
	Port register	PORT	Undefined	0008 C04Bh	8
	Input buffer control register	ICR	00h	0008 C06Bh	8
	Pull-up resistor control register	PCR	00h	0008 C0CBh	8
PORTC	Data direction register	DDR	00h	0008 C00Ch	8
	Data register	DR	00h	0008 C02Ch	8
	Port register	PORT	Undefined	0008 C04Ch	8
	Input buffer control register	ICR	00h	0008 C06Ch	8
	Open drain control register	ODR	00h	0008 C08Ch	8
	Pull-up resistor control register	PCR	00h	0008 C0CCh	8
PORTD	Data direction register	DDR	00h	0008 C00Dh	8
	Data register	DR	00h	0008 C02Dh	8
	Port register	PORT	Undefined	0008 C04Dh	8
	Input buffer control register	ICR	00h	0008 C06Dh	8
	Pull-up resistor control register	PCR	00h	0008 C0CDh	8
PORTE	Data direction register	DDR	00h	0008 C00Eh	8
	Data register	DR	00h	0008 C02Eh	8
	Port register	PORT	Undefined	0008 C04Eh	8
	Input buffer control register	ICR	00h	0008 C06Eh	8
	Pull-up resistor control register	PCR	00h	0008 C0CEh	8
PORTF	Data direction register	DDR	00h	0008 C00Fh	8
	Data register	DR	00h	0008 C02Fh	8
	Port register	PORT	Undefined	0008 C04Fh	8
	Input buffer control register	ICR	00h	0008 C06Fh	8
PORTG	Data direction register	DDR	00h	0008 C010h	8
	Data register	DR	00h	0008 C030h	8
	Port register	PORT	Undefined	0008 C050h	8
	Input buffer control register	ICR	00h	0008 C070h	8
	Pull-up resistor control register	PCR	00h	0008 C0D0h	8

Table 17.3 Registers of I/O Ports (176-Pin LFBGA) (3 / 3)

Port Symbol	Register Name	Register Symbol	Value after Reset	Address	Access Size
IOPORT	Port function control register 0	PFOCSE	00h	0008 C100h	8
	Port function control register 1	PF1CSS	00h	0008 C101h	8
	Port function control register 2	PF2CSS	00h	0008 C102h	8
	Port function control register 3	PF3BUS	00h	0008 C103h	8
	Port function control register 4	PF4BUS	00h	0008 C104h	8
	Port function control register 5	PF5BUS	00h	0008 C105h	8
	Port function control register 6	PF6BUS	00h	0008 C106h	8
	Port function control register 7	PF7DMA	00h	0008 C107h	8
	Port function control register 8	PF8IRQ	00h	0008 C108h	8
	Port function control register 9	PF9IRQ	00h	0008 C109h	8
	Port function control register A	PFAADC	00h	0008 C10Ah	8
	Port function control register B	PFBTMR	00h	0008 C10Bh	8
	Port function control register C	PFCMTU	00h	0008 C10Ch	8
	Port function control register D	PFDMTU	00h	0008 C10Dh	8
	Port function control register E	PFENET	00h	0008 C10Eh	8
	Port function control register F	PFFSCI	00h	0008 C10Fh	8
	Port function control register G	PFGSPI	00h	0008 C110h	8
	Port function control register H	PFHSPI	00h	0008 C111h	8
	Port function control register J	PFJCAN	00h	0008 C113h	8
Port function control register K	PFKUSB	00h	0008 C114h	8	
Port function control register L	PFLUSB	00h	0008 C115h	8	
Port function control register M	PFMPOE	00h	0008 C116h	8	
Port function control register N	PFNPOE	00h	0008 C117h	8	

Table 17.4 Valid Bits in Each Register (176-Pin LFBGA) (1 / 3)

Register Symbol	b7	b6	b5	b4	b3	b2	b1	b0
PORT0.DDR	√	x	√	x	√	√	√	√
PORT1.DDR	√	√	√	√	√	√	√	√
PORT2.DDR	√	√	√	√	√	√	√	√
PORT3.DDR	x	x	x	√	√	√	√	√
PORT4.DDR	√	√	√	√	√	√	√	√
PORT5.DDR	√	√	√	√	√	√	√	√
PORT6.DDR	√	√	√	√	√	√	√	√
PORT7.DDR	√	√	√	√	√	√	√	√
PORT8.DDR	x	x	√	√	√	√	√	√
PORT9.DDR	√	√	√	√	√	√	√	√
PORTA.DDR	√	√	√	√	√	√	√	√
PORTB.DDR	√	√	√	√	√	√	√	√
PORTC.DDR	√	√	√	√	√	√	√	√
PORTD.DDR	√	√	√	√	√	√	√	√
PORTE.DDR	√	√	√	√	√	√	√	√
PORTF.DDR	x	x	x	√	√	√	√	√
PORTG.DDR	√	√	√	√	√	√	√	√
PORT0.DR	√	x	√	x	√	√	√	√
PORT1.DR	√	√	√	√	√	√	√	√
PORT2.DR	√	√	√	√	√	√	√	√
PORT3.DR	x	x	x	√	√	√	√	√
PORT4.DR	√	√	√	√	√	√	√	√
PORT5.DR	√	√	√	√	x	√	√	√
PORT6.DR	√	√	√	√	√	√	√	√
PORT7.DR	√	√	√	√	√	√	√	√
PORT8.DR	x	x	√	√	√	√	√	√
PORT9.DR	√	√	√	√	√	√	√	√
PORTA.DR	√	√	√	√	√	√	√	√
PORTB.DR	√	√	√	√	√	√	√	√
PORTC.DR	√	√	√	√	√	√	√	√
PORTD.DR	√	√	√	√	√	√	√	√
PORTE.DR	√	√	√	√	√	√	√	√
PORTF.DR	x	x	x	√	√	√	√	√
PORTG.DR	√	√	√	√	√	√	√	√
PORT0.PORT	√	x	√	x	√	√	√	√
PORT1.PORT	√	√	√	√	√	√	√	√
PORT2.PORT	√	√	√	√	√	√	√	√
PORT3.PORT	x	x	√	√	√	√	√	√
PORT4.PORT	√	√	√	√	√	√	√	√
PORT5.PORT	√	√	√	√	√	√	√	√
PORT6.PORT	√	√	√	√	√	√	√	√
PORT7.PORT	√	√	√	√	√	√	√	√
PORT8.PORT	x	x	√	√	√	√	√	√
PORT9.PORT	√	√	√	√	√	√	√	√

Table 17.4 Valid Bits in Each Register (176-Pin LFBGA) (2 / 3)

Register Symbol	b7	b6	b5	b4	b3	b2	b1	b0
PORTA.PORT	√	√	√	√	√	√	√	√
PORTB.PORT	√	√	√	√	√	√	√	√
PORTC.PORT	√	√	√	√	√	√	√	√
PORTD.PORT	√	√	√	√	√	√	√	√
PORTE.PORT	√	√	√	√	√	√	√	√
PORTF.PORT	x	x	x	√	√	√	√	√
PORTG.PORT	√	√	√	√	√	√	√	√
PORT0.ICR	√	x	√	x	√	√	√	√
PORT1.ICR	√	√	√	√	√	√	√	√
PORT2.ICR	√	√	√	√	√	√	√	√
PORT3.ICR	x	x	x	√	√	√	√	√
PORT4.ICR	√	√	√	√	√	√	√	√
PORT5.ICR	√	√	√	√	√	√	√	√
PORT6.ICR	√	√	√	√	√	√	√	√
PORT7.ICR	√	√	√	√	√	√	√	√
PORT8.ICR	x	x	√	√	√	√	√	√
PORT9.ICR	√	√	√	√	√	√	√	√
PORTA.ICR	√	√	√	√	√	√	√	√
PORTB.ICR	√	√	√	√	√	√	√	√
PORTC.ICR	√	√	√	√	√	√	√	√
PORTD.ICR	√	√	√	√	√	√	√	√
PORTE.ICR	√	√	√	√	√	√	√	√
PORTF.ICR	x	x	x	√	√	√	√	√
PORTG.ICR	√	√	√	√	√	√	√	√
PORT0.ODR	√	x	√	x	√	√	√	√
PORT1.ODR	√	√	√	√	√	√	√	√
PORT2.ODR	√	√	√	√	√	√	√	√
PORT3.ODR	x	x	x	√	√	√	√	√
PORTC.ODR	√	√	√	√	√	√	√	√
PORT9.PCR	√	√	√	√	√	√	√	√
PORTA.PCR	√	√	√	√	√	√	√	√
PORTB.PCR	√	√	√	√	√	√	√	√
PORTC.PCR	√	√	√	√	√	√	√	√
PORTD.PCR	√	√	√	√	√	√	√	√
PORTE.PCR	√	√	√	√	√	√	√	√
PORTG.PCR	√	√	√	√	√	√	√	√
IOPORT.PF0CSE	√	√	√	√	√	√	√	√
IOPORT.PF1CSS	√	√	√	√	√	√	√	√
IOPORT.PF2CSS	√	√	√	√	√	√	x	√
IOPORT.PF3BUS	√	√	√	√	√	√	√	√
IOPORT.PF4BUS	√	√	√	√	√	√	√	√
IOPORT.PF5BUS	√	√	√	√	x	x	√	x
IOPORT.PF6BUS	√	√	x	√	x	x	√	√
IOPORT.PF7DMA	√	√	√	√	x	x	x	x
IOPORT.PF8IRQ	√	x	√	x	√	√	√	√

Table 17.4 Valid Bits in Each Register (176-Pin LFBGA) (3 / 3)

Register Symbol	b7	b6	b5	b4	b3	b2	b1	b0
IOPORT.PF9IRQ	√	√	√	√	√	√	√	√
IOPORT.PFAADC	x	x	x	x	x	x	x	√
IOPORT.PFBTMR	x	x	x	x	√	√	√	√
IOPORT.PFCMTU	√	√	√	√	√	√	x	x
IOPORT.PFDMTU	√	√	x	x	x	x	x	x
IOPORT.PFENET	√	x	x	√	√	√	√	√
IOPORT.PFFSCI	x	√	x	x	√	√	√	x
IOPORT.PFGSPI	√	√	√	√	√	√	√	√
IOPORT.PFHSPI	√	√	√	√	√	√	√	√
IOPORT.PFJCAN	x	x	x	x	x	x	x	√
IOPORT.PFKUSB	x	x	x	√	√	√	√	√
IOPORT.PFLUSB	x	x	x	√	√	√	√	√
IOPORT.PFMPOE	√	√	√	√	√	√	√	√
IOPORT.PFNPOE	x	x	x	x	x	x	√	√

17.1.2.1 Data Direction Register (DDR)

Addresses: PORT0.DDR 0008 C000h, PORT1.DDR 0008 C001h, PORT2.DDR 0008 C002h, PORT3.DDR 0008 C003h, PORT4.DDR 0008 C004h, PORT5.DDR 0008 C005h, PORT6.DDR 0008 C006h, PORT7.DDR 0008 C007h, PORT8.DDR 0008 C008h, PORT9.DDR 0008 C009h, PORTA.DDR 0008 C00Ah, PORTB.DDR 0008 C00Bh, PORTC.DDR 0008 C00Ch, PORTD.DDR 0008 C00Dh, PORTE.DDR 0008 C00Eh, PORTF.DDR 0008 C00Fh, PORTG.DDR 0008 C010h

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

Note: Bits 6 and 4 are reserved in PORT0.DDR.
 The lower five bits are valid and the upper three bits are reserved in PORT3.DDR.
 The lower six bits are valid and the upper two bits are reserved in PORT8.DDR.
 The lower five bits are valid and the upper three bits are reserved in PORTF.DDR.
 The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 I/O Select	0: An input pin	R/W
b1	B1	Pn1 I/O Select	1: An output pin	R/W
b2	B2	Pn2 I/O Select		R/W
b3	B3	Pn3 I/O Select		R/W
b4	B4	Pn4 I/O Select		R/W
b5	B5	Pn5 I/O Select		R/W
b6	B6	Pn6 I/O Select		R/W
b7	B7	Pn7 I/O Select		R/W

[Legend] n = 0 to 9, A to G

Each DDR is used to select the input or output direction for individual pins of the corresponding port that have been configured to function as general I/O pins.

Each bit of a PORTn.DDR (n = 0 to 9 and A to G) corresponds to a pin of PORTn, and the settings can change from bit to bit.

The PORT5.DDR.B3 bit selects P53 input or BCLK output. Setting the PORT5.DDR.B3 bit to 1 specifies output of the BCLK signal on the pin that would otherwise be P53. Operation as a general output is not selectable for this pin.

17.1.2.2 Data Register (DR)

Addresses: PORT0.DR 0008 C020h, PORT1.DR 0008 C021h, PORT2.DR 0008 C022h,
 PORT3.DR 0008 C023h, PORT4.DR 0008 C024h, PORT5.DR 0008 C025h,
 PORT6.DR 0008 C026h, PORT7.DR 0008 C027h, PORT8.DR 0008 C028h,
 PORT9.DR 0008 C029h, PORTA.DR 0008 C02Ah, PORTB.DR 0008 C02Bh,
 PORTC.DR 0008 C02Ch, PORTD.DR 0008 C02Dh, PORTE.DR 0008 C02Eh,
 PORTF.DR 0008 C02Fh, PORTG.DR 0008 C030h

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

Note: Bits 6 and 4 are reserved in PORT0.DR.

The lower five bits are valid and the upper three bits are reserved in PORT3.DR.

The lower six bits are valid and the upper two bits are reserved in PORT8.DR.

The lower five bits are valid and the upper three bits are reserved in PORTF.DR.

The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 Output Data Store	Output data are stored.	R/W
b1	B1	Pn1 Output Data Store		R/W
b2	B2	Pn2 Output Data Store		R/W
b3	B3	Pn3 Output Data Store		R/W
b4	B4	Pn4 Output Data Store		R/W
b5	B5	Pn5 Output Data Store		R/W
b6	B6	Pn6 Output Data Store		R/W
b7	B7	Pn7 Output Data Store		R/W

[Legend] n = 0 to 9, A to G

Each DR stores the output data from the individual pins of the corresponding port used as a general I/O port.

The output of the P53 pin is the BCLK signal and the value of the B3 bit in PORT5.DR does not affect the pin.

17.1.2.3 Port Register (PORT)

Addresses: PORT0.PORT 0008 C040h, PORT1.PORT 0008 C041h, PORT2.PORT 0008 C042h, PORT3.PORT 0009 C043h, PORT4.PORT 0008 C044h, PORT5.PORT 0008 C045h, PORT6.PORT 0008 C046h, PORT7.PORT 0008 C047h, PORT8.PORT 0008 C048h, PORT9.PORT 0008 C049h, PORTA.PORT 0008 C04Ah, PORTB.PORT 0008 C04Bh, PORTC.PORT 0008 C04Ch, PORTD.PORT 0008 C04Dh, PORTE.PORT 0008 C04Eh, PORTF.PORT 0008 C04Fh, PORTG.PORT 0008 C050h

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

Note: Bits 6 and 4 are reserved in PORT0.PORT.
 The lower six bits are valid and the upper two bits are reserved in PORT3.PORT.
 The lower six bits are valid and the upper two bits are reserved in PORT8.PORT.
 The lower five bits are valid and the upper three bits are reserved in PORTF.PORT.
 The reserved bits are read as 1 and cannot be modified.

Bit	Symbol	Bit Name	Description	R/W
b0	B0*	Pn0	Individual pin states of the corresponding port are reflected.	R
b1	B1*	Pn1		R
b2	B2*	Pn2		R
b3	B3*	Pn3		R
b4	B4*	Pn4		R
b5	B5*	Pn5		R
b6	B6*	Pn6		R
b7	B7*	Pn7		R

[Legend] n = 0 to 9, A to G

Note : * Before reading this register, set the corresponding bit in PORTn.ICR to 1. If this register is read with the corresponding bit in PORTn.ICR set to 0, the read value is undefined.

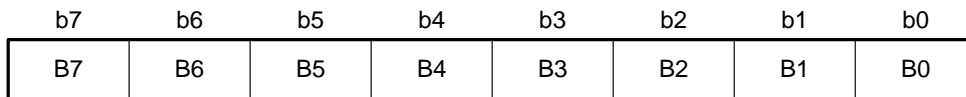
PORT reflects individual pin states of the corresponding port.

When a PORTn.PORT (n = 0 to 9 and A to G) is read, the corresponding pin states are read out to here.

The NMI pin state is read out to the P35 bit.

17.1.2.4 Input Buffer Control Register (ICR)

Addresses: PORT0.ICR 0008 C060h, PORT1.ICR 0008 C061h, PORT2.ICR 0008 C062h, PORT3.ICR 0008 C063h, PORT4.ICR 0008 C064h, PORT5.ICR 0008 C065h, PORT6.ICR 0008 C066h, PORT7.ICR 0008 C067h, PORT8.ICR 0008 C068h, PORT9.ICR 0008 C069h, PORTA.ICR 0008 C06Ah, PORTB.ICR 0008 C06Bh, PORTC.ICR 0008 C06Ch, PORTD.ICR 0008 C06Dh, PORTE.ICR 0008 C06Eh, PORTF.ICR 0008 C06Fh, PORTG.ICR 0008 C070h



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

Note: Bits 6 and 4 are reserved in PORT0.ICR.
 The lower five bits are valid and the upper three bits are reserved in PORT3.ICR.
 The lower six bits are valid and the upper two bits are reserved in PORT8.ICR.
 The lower five bits are valid and the upper three bits are reserved in PORTF.ICR.
 The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0*	Pn0 Input Buffer Control	0: The input buffer for the corresponding pin is disabled. 1: The input buffer for the corresponding pin is enabled.	R/W
b1	B1*	Pn1 Input Buffer Control		R/W
b2	B2*	Pn2 Input Buffer Control		R/W
b3	B3*	Pn3 Input Buffer Control		R/W
b4	B4*	Pn4 Input Buffer Control		R/W
b5	B5*	Pn5 Input Buffer Control		R/W
b6	B6*	Pn6 Input Buffer Control		R/W
b7	B7*	Pn7 Input Buffer Control		R/W

[Legend] n = 0 to 9, A to G

Note : * For pins being used as input pins, set the corresponding bits to 1. Set the bits corresponding to pins that are not being used for their input functions or are being used as analog I/O pins to 0.

Each ICR controls the input buffers for the individual pins of the corresponding port.

Each bit of a PORTn.ICR (n = 0 to 9 and A to G) corresponds to a pin of PORTn, and the settings can change from bit to bit.

When to be used as an input pin for the peripheral module, the input buffer for the corresponding pin should be enabled beforehand by setting the PORTn.ICR bit to 1. If this register is used as an input pin for the peripheral module while the PORTn.ICR bit is 0, the input signal to the peripheral module is fixed high. Additionally, when the external bus pins are used for the data bus, setting the PORTn.ICR bits to 1 is not required because the input buffers for the corresponding pins are automatically enabled.

Changes in the settings of a PORTn.ICR may generate edges internally, depending on the pin state. For this reason, change the settings of PORTn.ICR while the corresponding input pins are not in use. For example, in the case of IRQi (i = 0 to 15) inputs, change settings of the corresponding PORTn.ICR with interrupts disabled by clearing the IR flag in IRI (i = 64 to 79 ("i" shows an interrupt vector number)) of the interrupt control unit (ICU) to 0, and then enable the corresponding interrupts. If a change to a PORTn.ICR setting does generate an edge, negate the edge.

17.1.2.5 Open Drain Control Register (ODR)

Addresses: PORT0.ODR 0008 C080h, PORT1.ODR 0008 C081h, PORT2.ODR 0008 C082h,
PORT3.ODR 0008 C083h, PORTC.ODR 0008 C08Ch

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

Note: Bits 6 and 4 are reserved in PORT0.ODR.

The lower five bits are valid and the upper three bits are reserved in PORT3.ODR.

The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 Output Type Select	0: CMOS output pin	R/W
b1	B1	Pn1 Output Type Select	1: NMOS open-drain output pin	R/W
b2	B2	Pn2 Output Type Select		R/W
b3	B3	Pn3 Output Type Select		R/W
b4	B4	Pn4 Output Type Select		R/W
b5	B5	Pn5 Output Type Select		R/W
b6	B6	Pn6 Output Type Select		R/W
b7	B7	Pn7 Output Type Select		R/W

[Legend] n = 0 to 3, G

Each ODR is used to select an output type for the individual pins.

17.1.2.6 Pull-Up Resistor Control Register (PCR)

Addresses: PORT9.PCR 0008 C0C9h, PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTC.PCR 0008 C0CCh, PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh, PORTG.PCR 0008 C0D0h

	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	Pn0 Input Pull-Up Resistor Control	0: Input pull-up resistor is off. 1: Input pull-up resistor is on.	R/W
b1	B1	Pn1 Input Pull-Up Resistor Control		R/W
b2	B2	Pn2 Input Pull-Up Resistor Control		R/W
b3	B3	Pn3 Input Pull-Up Resistor Control		R/W
b4	B4	Pn4 Input Pull-Up Resistor Control		R/W
b5	B5	Pn5 Input Pull-Up Resistor Control		R/W
b6	B6	Pn6 Input Pull-Up Resistor Control		R/W
b7	B7	Pn7 Input Pull-Up Resistor Control		R/W

[Legend] n = 9, A to E, G

Each PCR controls enabled/disabled of input pull-up resistor for individual pins of the corresponding port.

When in input pin state, for the pins corresponding to bits where the value in PORTn.PCR is 1, input pull-up resistor is turned on. Table 17.5 summarizes the input pull-up resistor states.

Table 17.5 Input Pull-Up Resistor States (176-Pin LFBGA)

Port	Pin State	Reset or Hardware Standby Mode	In Other Operations
Port 9	Data I/O	Disabled	
	Address output	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port A	Address output	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port B	Address output	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port C	Address output	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	
	Port input	Disabled	
Port D	Data I/O	Disabled	Enabled/Disabled
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port E	Data I/O	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port G	Data I/O	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled

[Legend]

Disabled: Input pull-up resistor is always disabled.

Enabled/Disabled: Input pull-up resistor is enabled when the PORTm.PCR.Bj bit (m = 9, A to E, and G, j = 0 to 7) is set to 1, and disabled when the bit is cleared to 0.

17.1.2.7 Port Function Control Register 0 (PF0CSE)

Address: 0008 C100h

	b7	b6	b5	b4	b3	b2	b1	b0
	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CS0E	CS0 Output Enable	0: Designated as an I/O port pin.	R/W
b1	CS1E	CS1 Output Enable	1: Designated as the CSn# output pin (n = 0 to 7)	R/W
b2	CS2E	CS2 Output Enable		R/W
b3	CS3E	CS3 Output Enable		R/W
b4	CS4E	CS4 Output Enable		R/W
b5	CS5E	CS5 Output Enable		R/W
b6	CS6E	CS6 Output Enable		R/W
b7	CS7E	CS7 Output Enable		R/W

PF0CSE enables or disables CSn# output.

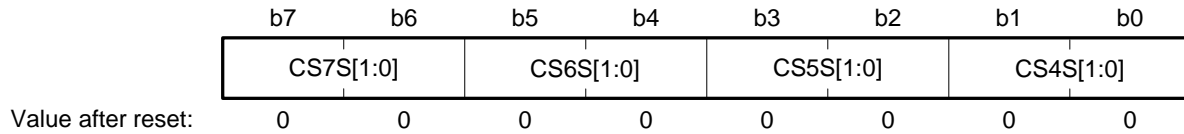
CSnE Bit (CSn Output Enable) (n = 0 to 7)

Each bit enables or disables the corresponding CSn# output.

To output a CSn signal, set the corresponding CSnE bit in PF0CSE to 1.

17.1.2.8 Port Function Control Register 1 (PF1CSS)

Address: 0008 C101h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	CS4S[1:0]	CS4# Output Pin Select	b1 b0 0 0: P64 is designated as the CS4#-A output pin. 0 1: P74 is designated as the CS4#-B output pin. 1 x: P24 is designated as the CS4#-C output pin.	R/W
b3, b2	CS5S[1:0]	CS5# Output Pin Select	b3 b2 0 0: P65 is designated as the CS5#-A output pin. 0 1: P75 is designated as the CS5#-B output pin. 1 x: P25 is designated as the CS5#-C output pin.	R/W
b5, b4	CS6S[1:0]	CS6# Output Pin Select	b5 b4 0 0: P66 is designated as the CS6#-A output pin. 0 1: P76 is designated as the CS6#-B output pin. 1 x: P26 is designated as the CS6#-C output pin.	R/W
b7, b6	CS7S[1:0]	CS7# Output Pin Select	b7 b6 0 0: P67 is designated as the CS7#-A output pin. 0 1: P77 is designated as the CS7#-B output pin. 1 x: P27 is designated as the CS7#-C output pin.	R/W

[Legend]

x: Don't care

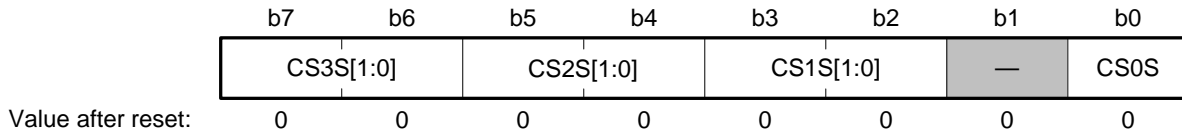
PF1CSS is used to select a pin for each CSn# output (n = 4 to 7).

CSnS[1:0] Bits (CSn# Output Pin Select) (n = 4 to 7)

These bits select a pin for each CSn# output when a CSn# output is enabled (the corresponding CSnE bit in PFOCSE = 1).

17.1.2.9 Port Function Control Register 2 (PF2CSS)

Address: 0008 C102h



Bit	Symbol	Bit Name	Description	R/W
b0	CS0S	CS0# Output Pin Select	0: P60 is designated as the CS0#-A output pin. 1: PC7 is designated as the CS0#-B output pin.	R/W
b1	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b3, b2	CS1S[1:0]	CS1# Output Pin Select	b3 b2 0 0: P61 is designated as the CS1#-A output pin. 0 1: P71 is designated as the CS1#-B output pin. 1 x: PC6 is designated as the CS1#-C output pin.	R/W
b5, b4	CS2S[1:0]	CS2# Output Pin Select	b5 b4 0 0: P62 is designated as the CS2#-A output pin. 0 1: P72 is designated as the CS2#-B output pin. 1 x: PC5 is designated as the CS2#-C output pin.	R/W
b7, b6	CS3S[1:0]	CS3# Output Pin Select	b7 b6 0 0: P63 is designated as the CS3#-A output pin. 0 1: P73 is designated as the CS3#-B output pin. 1 x: PC4 is designated as the CS3#-C output pin.	R/W

[Legend]

x: Don't care

PF2CSS is used to select a pin for each CSn# output (n = 0 to 3).

CSnS[1:0] Bits (CSn# Output Pin Select) (n = 0 to 3)

These bits select a pin for each CSn# output when a CSn# output is enabled (the corresponding CSnE bit in PFOCSE = 1).

17.1.2.10 Port Function Control Register 3 (PF3BUS)

Address: 0008 C103h

	b7	b6	b5	b4	b3	b2	b1	b0
	A23E	A22E	A21E	A20E	A19E	A18E	A17E	A16E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	A16E	Address A16 Output Enable	0: A16 output is disabled. 1: A16 output is enabled.	R/W
b1	A17E	Address A17 Output Enable	0: A17 output is disabled. 1: A17 output is enabled.	R/W
b2	A18E	Address A18 Output Enable	0: A18 output is disabled. 1: A18 output is enabled.	R/W
b3	A19E	Address A19 Output Enable	0: A19 output is disabled. 1: A19 output is enabled.	R/W
b4	A20E	Address A20 Output Enable	0: A20 output is disabled. 1: A20 output is enabled.	R/W
b5	A21E	Address A21 Output Enable	0: A21 output is disabled. 1: A21 output is enabled.	R/W
b6	A22E	Address A22 Output Enable	0: A22 output is disabled. 1: A22 output is enabled.	R/W
b7	A23E	Address A23 Output Enable	0: A23 output is disabled. 1: A23 output is enabled.	R/W

PF3BUS enables or disables address outputs.

AnE Bit (Address An Output Enable) (n = 16 to 23)

Each bit enables or disables an address output (An).

17.1.2.11 Port Function Control Register 4 (PF4BUS)

Address: 0008 C104h

b7	b6	b5	b4	b3	b2	b1	b0
A15E	A14E	A13E	A12E	A11E	A10E	ADRLE[1:0]	
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ADRLE[1: 0]	Address Lower A9 to A0 Output Enable	b1 b0 0 0: A9 to A0 output is disabled. 0 1: A9 to A4 output is disabled, A3 to A0 output is enabled 1 0: A9 to A8 output is disabled, A7 to A0 output is enabled 1 1: A9 to A0 output is enabled	R/W
b2	A10E	Address A10 Output Enable	0: A10 output is disabled. 1: A10 output is enabled.	R/W
b3	A11E	Address A11 Output Enable	0: A11 output is disabled. 1: A11 output is enabled.	R/W
b4	A12E	Address A12 Output Enable	0: A12 output is disabled. 1: A12 output is enabled.	R/W
b5	A13E	Address A13 Output Enable	0: A13 output is disabled. 1: A13 output is enabled.	R/W
b6	A14E	Address A14 Output Enable	0: A14 output is disabled. 1: A14 output is enabled.	R/W
b7	A15E	Address A15 Output Enable	0: A15 output is disabled. 1: A15 output is enabled.	R/W

PF4BUS enables or disables address outputs.

ADRLE[1: 0] Bits (Address Lower A9 to A0 Output Enable)

These bits enable or disable an address output (A9 to A0).

AnE Bit (Address An Output Enable) (n = 10 to 15)

Each bit enables or disables an address output (An).

17.1.2.12 Port Function Control Register 5 (PF5BUS)

Address: 0008 C105h

	b7	b6	b5	b4	b3	b2	b1	b0
	WR32 BC32E	WR1BC1E	DH32E	DHE	—	—	ADRHMS	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b1	ADRHMS	A23-to-A16 Output Select	A23-to-A16 Output Select 0: PC7 to PC0 are designated as the A23-A to A16-A pins (function as part of the external address bus) 1: P97 to P90 are designated as the A23-B to A16-B pins (function as part of the external address bus)	R/W
b3, b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	DHE	D15-to-D8 Enable	0: PE7 to PE0 are designated as I/O port pins. 1: PE7 to PE0 are designated as D15 to D8 pins (function as part of the external data bus)	R/W
b5	DH32E	D31-to-D16 Enable	0: PG7 to PG0 and P97 to P90 are designated as I/O port pins. 1: PG7 to PG0 and P97 to P90 are designated as D31 to D16 pins (function as part of the external data bus)	R/W
b6	WR1BC1E	WR1#/BC1# Output Enable	0: P51 is designated as an I/O port pin. 1: P51 is designated as the WR1# or BC1# pin.	R/W
b7	WR32BC32E	WR3#/BC3# and WR2#/BC2# Output Enable	0: P56 and P57 are designated as I/O port pins. 1: P56 is designated as the WR2# or BC2# pin P57 is designated as the WR3# or BC3# pin.	R/W

ADRHMS Bit (A23-to-A16 Output Select)

This bit selects the set of pins for the output of address signals A23 to A16.

DHE Bit (D15-to-D8 Enable)

This bit enables or disables the input and output of data signals D15 to D8 (valid in expansion mode with on-chip ROM disabled or enabled).

DH32E Bit (D31-to-D16 Enable)

This bit enables or disables the input and output of data signals D31 to D16 (valid in expansion mode with on-chip ROM disabled or enabled).

WR1BC1E Bit (WR1#/BC1# Output Enable)

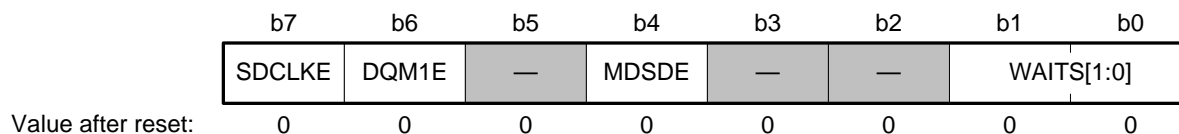
This bit enables or disables WR1#/BC1# output (valid in expansion mode with on-chip ROM disabled or enabled).

WR32BC32E Bit (WR3#/BC3# and WR2#/BC2# Output Enable)

This bit enables or disables WR3#/BC3# and WR2#/BC2# output (valid in expansion mode with on-chip ROM disabled or enabled).

17.1.2.13 Port Function Control Register 6 (PF6BUS)

Address: 0008 C106h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	WAITS[1:0]	WAIT Select	b1 b0 0 0: P57 is designated as the WAIT#-A input pin. 0 1: P55 is designated as the WAIT#-B input pin. 1 0: PC5 is designated as the WAIT#-C input pin. 1 1: P51 is designated as the WAIT#-D input pin.	R/W
b3, b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	MDSDE	SDRAM Pins Enable	See bit 6 (the DQM1E bit) in this register.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b6	DQM1E	DQM1 Output Enable	MDSDE DQM1E 0 x: Of the SDRAM pins, only the P70/SDCLK pin is available. The SDCLKE bit controls whether or not it is enabled. 1 0: All pin functions for the SDRAM (other than the P67/DQM1 pin) are enabled. 1 1: All pin functions for the SDRAM are enabled.	R/W
b7	SDCLKE	SDCLK Output Enable	0: SDCLK output is disabled. 1: SDCLK output is enabled.	R/W

WAITS Bits (WAIT Select)

These bits select a pin for a WAIT# input.

MDSDE Bit (SDRAM Pins Enable)

This bit enables or disables the output of the SDRAM pins.

While the MDSDE bit is 1, the output of the DQM1 pin is enabled or disabled independently according to the DQM1E bit setting. The output of the SDCLK pin is enabled or disabled independently according to the SDCLKE bit setting, regardless of the MDSDE bit setting.

DQM1E Bit (DQM1 Output Enable)

This bit enables or disables the output of the DQM1 pin.

When MDSDE bit is set to 1, the DQM1E bit setting is enabled. When the MDSDE bit is cleared to 0, the DQM1E bit setting is ignored.

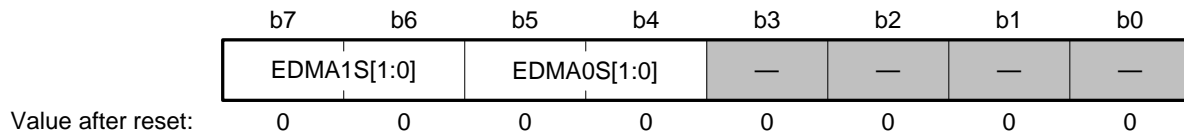
SDCLK Bit (SDCLK Output Enable)

This bit enables or disables the output of the SDCLK pin.

The SDCLK bit setting should be changed after the clock to the SDCLK has been stopped.

17.1.2.14 Port Function Control Register 7 (PF7DMA)

Address: 0008 C107h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b5, b4	EDMA0S[1: 0]	EXDMAC0 Pin Select	b5 b4 0 0: P80 is designated as the EDREQ0-A pin. P81 is designated as the EDACK0-A pin. 0 1: P22 is designated as the EDREQ0-B pin. P23 is designated as the EDACK0-B pin. 1 x: P55 is designated as the EDREQ0-C pin. P54 is designated as the EDACK0-C pin.	R/W
b7, b6	EDMA1S[1: 0]	EXDMAC1 Pin Select	b7 b6 0 0: P82 is designated as the EDREQ1-A pin. P83 is designated as the EDACK1-A pin. 0 1: P24 is designated as the EDREQ1-B pin. P25 is designated as the EDACK1-B pin. 1 x: P57 is designated as the EDREQ1-C pin. P56 is designated as the EDACK1-C pin.	R/W

[Legend] x: Don't care

EDMA_nS Bits (EXDMAC_n Pin Select) (n = 0, 1)These bits select a pin for EXDMAC_n.

17.1.2.15 Port Function Control Register 8 (PF8IRQ)

Address: 0008 C108h

	b7	b6	b5	b4	b3	b2	b1	b0
	ITS15	—	ITS13	—	ITS11	ITS10	ITS9	ITS8
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITS8	IRQ8 Pin Select	0: P00 is designated as the IRQ8-A input pin. 1: P40 is designated as the IRQ8-B input pin.	R/W
b1	ITS9	IRQ9 Pin Select	0: P01 is designated as IRQ9-A input pin. 1: P41 is designated as IRQ9-B input pin.	R/W
b2	ITS10	IRQ10 Pin Select	0: P02 is designated as the IRQ10-A input pin. 1: P42 is designated as the IRQ10-B input pin	R/W
b3	ITS11	IRQ11 Pin Select	0: P03 is designated as the IRQ11-A input pin. 1: P43 is designated as the IRQ11-B input pin.	R/W
b4	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b5	ITS13	IRQ13 Pin Select	0: P05 is designated as the IRQ13-A input pin. 1: P45 is designated as the IRQ13-B input pin.	R/W
b6	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b7	ITS15	IRQ15 Pin Select	0: P07 is designated as the IRQ15-A input pin. 1: P47 is designated as the IRQ15-B input pin.	R/W

PF8IRQ is used to select pins for IRQ8 to IRQ15 inputs.

ITS_i Bit (IRQ_i Pin Select) (i = 8 to 11, 13, 15)

Each bit selects a pin for an IRQ_i input.

17.1.2.16 Port Function Control Register 9 (PF9IRQ)

Address: 0008 C109h

	b7	b6	b5	b4	b3	b2	b1	b0
	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	ITS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITS0	IRQ0 Pin Select	0: P30 is designated as the IRQ0-A input pin. 1: P10 is designated as the IRQ0-B input pin.	R/W
b1	ITS1	IRQ1 Pin Select	0: P31 is designated as the IRQ1-A input pin. 1: P11 is designated as the IRQ1-B input pin.	R/W
b2	ITS2	IRQ2 Pin Select	0: P32 is designated as the IRQ2-A input pin. 1: P12 is designated as the IRQ2-B input pin.	R/W
b3	ITS3	IRQ3 Pin Select	0: P33 is designated as the IRQ3-A input pin. 1: P13 is designated as the IRQ3-B input pin.	R/W
b4	ITS4	IRQ4 Pin Select	0: P34 is designated as the IRQ4-A input pin. 1: P14 is designated as the IRQ4-B input pin.	R/W
b5	ITS5	IRQ5 Pin Select	0: PE5 is designated as the IRQ5-A input pin. 1: P15 is designated as the IRQ5-B input pin.	R/W
b6	ITS6	IRQ6 Pin Select	0: PE6 is designated as the IRQ6-A input pin. 1: P16 is designated as the IRQ6-B input pin.	R/W
b7	ITS7	IRQ7 Pin Select	0: PE7 is designated as the IRQ7-A input pin. 1: P17 is designated as the IRQ7-B input pin.	R/W

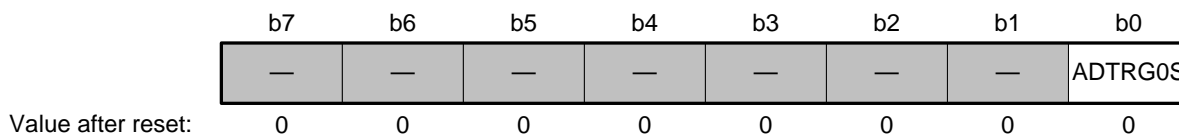
PF9IRQ is used to select pins for IRQ0 to IRQ7 inputs.

ITS_i Bit (IRQ_i Pin Select) (i = 0 to 7)

Each bit selects a pin for an IRQ_i input.

17.1.2.17 Port Function Control Register A (PFAADC)

Address: 0008 C10Ah



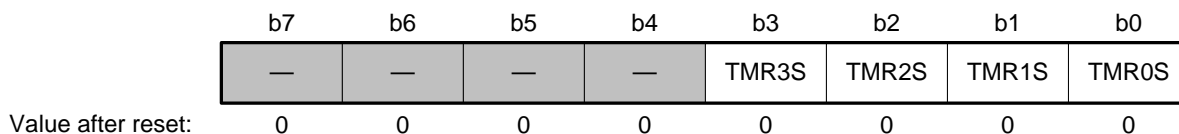
Bit	Symbol	Bit Name	Description	R/W
b0	ADTRG0S	ADTRG0# Input Select	0: P07 is designated as the ADTRG0#-A input pin. 1: P25 is designated as the ADTRG0#-B input pin.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

ADTRG0S Bit (ADTRG0# Input Select)

This bit selects a pin for an ADTRG0# input.

17.1.2.18 Port Function Control Register B (PFBTMR)

Address: 0008 C10Bh



Bit	Symbol	Bit Name	Description	R/W
b0	TMR0S	TMR0 Input Pin Select	0: P01 is designated as the TMCIO-A pin. P00 is designated as the TMRIO-A pin. 1: P21 is designated as the TMCIO-B pin. P20 is designated as the TMRIO-B pin.	R/W
b1	TMR1S	TMR1 Input Pin Select	0: P02 is designated as the TMC11-A pin. 1: P12 is designated as the TMC11-B pin.	R/W
b2	TMR2S	TMR2 Input Pin Select	0: P15 is designated as the TMC12-A pin. 1: P31 is designated as the TMC12-B pin.	R/W
b3	TMR3S	TMR3 Input Pin Select	0: P11 is designated as the TMC13-A pin. P10 is designated as the TMRI3-A pin. 1: P34 is designated as the TMC13-B pin. P30 is designated as the TMRI3-B pin.	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PFBTMR is used to select pins for TMR0 to TMR3.

TMRnS Bit (TMRn Input Pin Select) (n = 0 to 3)

Each bit selects a pin for a TMRn input.

17.1.2.19 Port Function Control Register C (PFCMTU)

Address: 0008 C10Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	TCLKS	MTUS6	MTUS5	MTUS4	MTUS3	MTUS2	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b2	MTUS2	MTU Pin Select 2	0: P16 is designated as the MTIOC3C-A pin. 1: P56 is designated as the MTIOC3C-B pin.	R/W
b3	MTUS3	MTU Pin Select 3	0: P22 is designated as the MTIOC3B-A pin. P23 is designated as the MTIOC3D-A pin. 1: P80 is designated as the MTIOC3B-B pin. P81 is designated as the MTIOC3D-B pin.	R/W
b4	MTUS4	MTU Pin Select 4	0: P24 is designated as the MTIOC4A-A pin. P25 is designated as the MTIOC4C-A pin. 1: P82 is designated as the MTIOC4A-B pin. P83 is designated as the MTIOC4C-B pin.	R/W
b5	MTUS5	MTU Pin Select 5	0: P30 is designated as the MTIOC4B-A pin. P31 is designated as the MTIOC4D-A pin. 1: P54 is designated as the MTIOC4B-B pin. P55 is designated as the MTIOC4D-B pin.	R/W
b6	MTUS6	MTU Pin Select 6	0: P12 is designated as the MTIC5U-A pin. P11 is designated as the MTIC5V-A pin. P10 is designated as the MTIC5W-A pin. 1: PD7 is designated as the MTIC5U-B pin. PD6 is designated as the MTIC5V-B pin. PD5 is designated as the MTIC5W-B pin.	R/W
b7	TCLKS	MTCLK Pin Select	0: P24 is designated as the MTCLKA-A pin. P25 is designated as the MTCLKB-A pin. P22 is designated as the MTCLKC-A pin. P23 is designated as the MTCLKD-A pin. 1: PC6 is designated as the MTCLKA-B pin. PC7 is designated as the MTCLKB-B pin. PC4 is designated as the MTCLKC-B pin. PC5 is designated as the MTCLKD-B pin.	R/W

PFCMTU is used to select pins for MTU unit 0.

MTUS_j Bit (MTU Pin Select) (j = 2 to 6)

Each bit selects a pin for an MTU input/output.

TCLKS Bit (MTCLK Pin Select)

This bit selects a pin for a MTCLK input of the MTU.

17.1.2.20 Port Function Control Register D (PFDMTU)

Address: 0008 C10Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	TCLKS	MTUS6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b6	MTUS6	MTU Pin Select 6	0: PC7 is designated as the MTIC11U-A pin. PC6 is designated as the MTIC11V-A pin. PC5 is designated as the MTIC11W-A pin. 1: PD4 is designated as the MTIC11U-B pin. PD3 is designated as the MTIC11V-B pin. PD2 is designated as the MTIC11W-B pin.	R/W
b7	TCLKS	MTCLK Pin Select	0: PC2 is designated as the MTCLKE-A pin. PC3 is designated as the MTCLKF-A pin. PC0 is designated as the MTCLKG-A pin. PC1 is designated as the MTCLKH-A pin. 1: PB4 is designated as the MTCLKE-B pin. PB5 is designated as the MTCLKF-B pin. PB2 is designated as the MTCLKG-B pin. PB3 is designated as the MTCLKH-B pin.	R/W

PFDMTU is used to select pins for MTU unit 1.

MTUS6 Bit (MTU Pin Select 6)

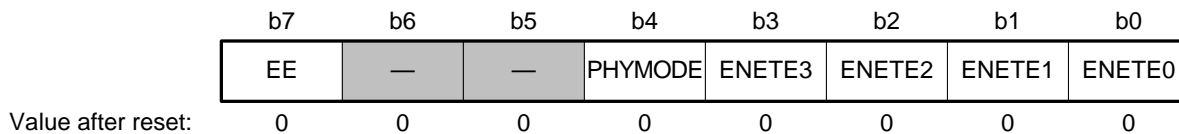
Each bit selects a pin for a MTIC11U/V/W input.

TCLKS Bit (MTCLK Pin Select)

This bit selects a pin for a MTCLK input of the MTU.

17.1.2.21 Port Function Control Register E (PFENET)

Address: 0008 C10Eh



Bit	Symbol	Bit Name	Description	R/W
b0	ENETE0	Ethernet Pin Enable 0	0: The ET_WOL pin is disabled. 1: The ET_WOL pin is enabled.	R/W
b1	ENETE1	Ethernet Pin Enable 1	0: The ET_LINKSTA pin is disabled. 1: The ET_LINKSTA pin is enabled.	R/W
b2	ENETE2	Ethernet Pin Enable 2	0: The ET_EXOUT pin is disabled. 1: The ET_EXOUT pin is enabled.	R/W
b3	ENETE3	Ethernet Pin Enable 3	0: The ET_TX_ER pin is disabled. 1: The ET_TX_ER pin is enabled.	R/W
b4	PHYMODE	Ethernet Mode Setting	0: RMI mode 1: MII mode	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	EE	Ethernet Pins Enable	0: All pin functions for the Ethernet interface are disabled. 1: All pin functions for the Ethernet interface are enabled.	R/W

PFENET is used to select I/O pins for the Ethernet interface.

ENETE0 to ENETE3 Bits (Ethernet Pin Enable)

These bits select pins for the Ethernet interface.

PHYMODE Bit (Ethernet Mode Setting)

This bit selects the PHY mode for the Ethernet interface.

Table 17.6 lists the relationship between the setting of the PHYMODE bit and the mode of the Ethernet interface.

EE Bit (Ethernet Pins Enable)

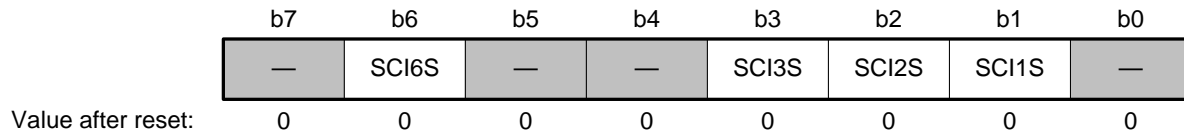
This bit enables or disables all pin functions for the Ethernet interface.

Table 17.6 Relationship between the PHYMODE Bit Setting and Ethernet Mode (176-Pin LFBGA)

PHYMODE	Ethernet Mode	Pin to be Used for the		Remark
		Ethernet Interface	Pin Allocation	
0	RMII mode	ET_MDC	P72	
		ET_MDIO	P71	
		ET_WOL	P73	Enabled when ENETE0 = 1
		ET_LINKSTA	P54	Enabled when ENETE1 = 1
		ET_EXOUT	P55	Enabled when ENETE2 = 1
		REF50CK	P76	
		RMII_TXD0	P81	
		RMII_TXD1	P82	
		RMII_TXD_EN	P80	
		RMII_RXD0	P75	
		RMII_RXD1	P74	
		RMII_RX_ER	P77	
		RMII_CRSDV	P83	
		1	MII mode	ET_MDC
ET_MDIO	P71			
ET_WOL	P73			Enabled when ENETE0 = 1
ET_LINKSTA	P54			Enabled when ENETE1 = 1
ET_EXOUT	P55			Enabled when ENETE2 = 1
ET_TX_CLK	PC4			
ET_ETXD0	P81			
ET_ETXD1	P82			
ET_ETXD2	PC5			
ET_ETXD3	PC6			
ET_TX_EN	P80			
ET_TX_ER	PC3			Enabled when ENETE3 = 1
ET_COL	PC7			
ET_CRSDV	P83			
ET_RX_CLK	P76			
ET_ERXD0	P75			
ET_ERXD1	P74			
ET_ERXD2	PC1			
ET_ERXD3	PC0			
ET_RX_DV	PC2			
ET_RX_ER	P77			

17.1.2.22 Port Function Control Register F (PFFSCI)

Address: 0008 C10Fh



Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b1	SCI1S	SCI1 Pin Select	0: P30 is designated as the RxD1-A pin. P27 is designated as the SCK1-A pin. P26 is designated as the TxD1-A pin. 1: PF2 is designated as the RxD1-B pin. PF1 is designated as the SCK1-B pin. PF0 is designated as the TxD1-B pin.	R/W
b2	SCI2S	SCI2 Pin Select	0: P12 is designated as the RxD2-A pin. P11 is designated as the SCK2-A pin. P13 is designated as the TxD2-A pin. 1: P52 is designated as the RxD2-B pin. P51 is designated as the SCK2-B pin. P50 is designated as the TxD2-B pin.	R/W
b3	SCI3S	SCI3 Pin Select	0: P16 is designated as the RxD3-A pin. P15 is designated as the SCK3-A pin. P17 is designated as the TxD3-A pin. 1: P25 is designated as the RxD3-B pin. P24 is designated as the SCK3-B pin. P23 is designated as the TxD3-B pin.	R/W
b5, b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b6	SCI6S	SCI6 Pin Select	0: P01 is designated as the RxD6-A pin. P02 is designated as the SCK6-A pin. P00 is designated as the TxD6-A pin. 1: P33 is designated as the RxD6-B pin. P34 is designated as the SCK6-B pin. P32 is designated as the TxD6-B pin.	R/W
b7	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

PFFSCI is used to select pins for SCI.

SCI_nS Bit (SCI_n Pin Select) (n = 1 to 3, 6)

Each bit selects a pin for an SCI channel-n input/output.

17.1.2.23 Port Function Control Register G (PFGSPI)

Address: 0008 C110h

	b7	b6	b5	b4	b3	b2	b1	b0
	SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	RSPIS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPIS	RSPI Pin Select	0: PC7 is designated as the MISOA-A pin. PC6 is designated as the MOSIA-A pin. PC5 is designated as the RSPCKA-A pin. PC4 is designated as the SSLA0-A pin. PC0 is designated as the SSLA1-A pin. PC1 is designated as the SSLA2-A pin. PC2 is designated as the SSLA3-A pin. 1: PA7 is designated as the MISOA-B pin. PA6 is designated as the MOSIA-B pin. PA5 is designated as the RSPCKA-B pin. PA4 is designated as the SSLA0-B pin. PA0 is designated as the SSLA1-B pin. PA1 is designated as the SSLA2-B pin. PA2 is designated as the SSLA3-B pin.	R/W
b1	RSPCKE	RSPCKA Output Enable	0: The RSPCKA pin is disabled. 1: The RSPCKA pin is enabled.	R/W
b2	MOSIE	MOSIA Output Enable	0: The MOSIA pin is disabled. 1: The MOSIA pin is enabled.	R/W
b3	MISOE	MISOA Output Enable	0: The MISOA pin is disabled. 1: The MISOA pin is enabled.	R/W
b4	SSL0E	SSLA0 Output Enable	0: The SSLA0 pin is disabled. 1: The SSLA0 pin is enabled.	R/W
b5	SSL1E	SSLA1 Output Enable	0: The SSLA1 pin is disabled. 1: The SSLA1 pin is enabled.	R/W
b6	SSL2E	SSLA2 Output Enable	0: The SSLA2 pin is disabled. 1: The SSLA2 pin is enabled.	R/W
b7	SSL3E	SSLA3 Output Enable	0: The SSLA3 pin is disabled. 1: The SSLA3 pin is enabled.	R/W

PFGSPI is used to select I/O pins for RSPI channel 0.

RSPIS Bit (RSPI Pin Select)

This bit selects a pin for an RSPI input/output.

As an enable bit is provided for each RSPI input/output pin, the input/output pin is selectable while the corresponding enable bit is 1. Otherwise, the pin cannot be selected.

RSPCKE Bit (RSPCKA Output Enable)

This bit enables or disables the output of the RSPCKA pin. Set this bit to 1 to use the RSPCKA pin.

MOSIE Bit (MOSIA Output Enable)

This bit enables or disables the output of the MOSIA pin. Set this bit to 1 to use the MOSIA pin.

MISOE Bit (MISOA Output Enable)

This bit enables or disables the output of the MISOA pin. Set this bit to 1 to use the MISOA pin.

SSL0E Bit (SSLA0 Output Enable)

This bit enables or disables the output of the SSLA0 pin. Set this bit to 1 to use the SSLA0 pin.

SSL1E Bit (SSLA1 Output Enable)

This bit enables or disables the output of the SSLA1 pin. Set this bit to 1 to use the SSLA1 pin.

SSL2E Bit (SSLA2 Output Enable)

This bit enables or disables the output of the SSLA2 pin. Set this bit to 1 to use the SSLA2 pin.

SSL3E Bit (SSLA3 Output Enable)

This bit enables or disables the output of the SSLA3 pin. Set this bit to 1 to use the SSLA3 pin.

17.1.2.24 Port Function Control Register H (PFHSPI)

Address: 0008 C111h

	b7	b6	b5	b4	b3	b2	b1	b0
	SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	RSPIS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPIS	RSPI Pin Select	0: P30 is designated as the MISOB-A pin. P26 is designated as the MOSIB-A pin. P27 is designated as the RSPCKB-A pin. P31 is designated as the SSLB0-A pin. P50 is designated as the SSLB1-A pin. P51 is designated as the SSLB2-A pin. P52 is designated as the SSLB3-A pin. 1: PE7 is designated as the MISOB-B pin. PE6 is designated as the MOSIB-B pin. PE5 is designated as the RSPCKB-B pin. PE4 is designated as the SSLB0-B pin. PE0 is designated as the SSLB1-B pin. PE1 is designated as the SSLB2-B pin. PE2 is designated as the SSLB3-B pin.	R/W
b1	RSPCKE	RSPCKB Output Enable	0: The RSPCKB pin is disabled. 1: The RSPCKB pin is enabled	R/W
b2	MOSIE	MOSIB Output Enable	0: The MOSIB pin is disabled. 1: The MOSIB pin is enabled	R/W
b3	MISOE	MISOB Output Enable	0: The MISOB pin is disabled. 1: The MISOB pin is enabled	R/W
b4	SSL0E	SSLB0 Output Enable	0: The SSLB0 pin is disabled. 1: The SSLB0 pin is enabled	R/W
b5	SSL1E	SSLB1 Output Enable	0: The SSLB1 pin is disabled. 1: The SSLB1 pin is enabled	R/W
b6	SSL2E	SSLB2 Output Enable	0: The SSLB2 pin is disabled. 1: The SSLB2 pin is enabled	R/W
b7	SSL3E	SSLB3 Output Enable	0: The SSLB3 pin is disabled. 1: The SSLB3 pin is enabled	R/W

PFHSPI is used to select I/O pins for RSPI channel 1.

RSPIS Bit (RSPI Pin Select)

This bit selects a pin for an RSPI input/output.

As an enable bit is provided for each RSPI input/output pin, the input/output pin is selectable while the corresponding enable bit is 1. Otherwise, the pin cannot be selected.

RSPCKE Bit (RSPCKB Output Enable)

This bit enables or disables the output of the RSPCKB pin. Set this bit to 1 to use the RSPCKB pin.

MOSIE Bit (MOSIB Output Enable)

This bit enables or disables the output of the MOSIB pin. Set this bit to 1 to use the MOSIB pin.

MISOE Bit (MISOB Output Enable)

This bit enables or disables the output of the MISOB pin. Set this bit to 1 to use the MISOB pin.

SSL0E Bit (SSLB0 Output Enable)

This bit enables or disables the output of the SSLB0 pin. Set this bit to 1 to use the SSLB0 pin.

SSL1E Bit (SSLB1 Output Enable)

This bit enables or disables the output of the SSLB1 pin. Set this bit to 1 to use the SSLB1 pin.

SSL2E Bit (SSLB2 Output Enable)

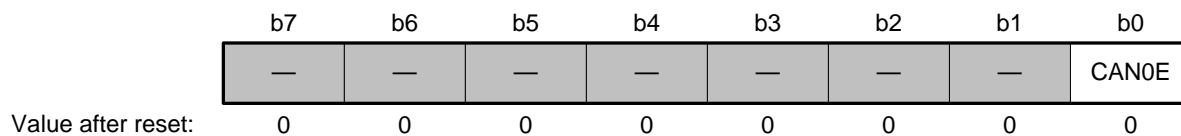
This bit enables or disables the output of the SSLB2 pin. Set this bit to 1 to use the SSLB2 pin.

SSL3E Bit (SSLB3 Output Enable)

This bit enables or disables the output of the SSLB3 pin. Set this bit to 1 to use the SSLB3 pin.

17.1.2.25 Port Function Control Register J (PFJCAN)

Address: 0008 C113h



Bit	Symbol	Bit Name	Description	R/W
b0	CAN0E	CAN0 Pins Enable	0: The CTX0 and CRX0 pins are disabled. 1: The CTX0 and CRX0 pins are enabled.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

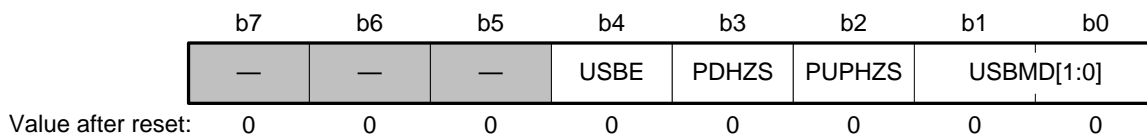
PFJCAN is used to select I/O pins for the CAN.

CANnE Bit (CANn Pins Enable) (n = 0)

This bit enables or disables the CANn pins. Set this bit to 1 to use the CANn pins.

17.1.2.26 Port Function Control Register K (PFKUSB)

Address: 0008 C114h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	USBMD[1:0]	USB Mode Setting	b1 b0 0 0: Select function mode for the USB0 pins. 0 1: Select host mode for the USB0 pins. 1 0: Select host/function mode for the USB0 pin. (as an optional function)* 1 1: Select OTG mode for the USB0 pins.	R/W
b2	PUPHZS	PUPHZ Select	0: USB0_DPUPE pin is for output of the high and low levels (external pull-up control signal). 1: USB0_DPUPE pin is for high-level output or the Hi-Z state (pull-up output is from the USB0_DP pin).	R/W
b3	PDHZS	PDHZ Select	0: USB0_DPRPD and USB0_DRPD pins are for output of the high and low levels (external pull-down control signals). 1: USB0_DPRPD and USB0_DRPD pins are for low-level output or the Hi-Z state (pull-down output is from the USB0_DP and USB0_DM pins).	R/W
b4	USBE	USB Enable	0: All pin functions for USB0 are disabled. 1: All pin functions for USB0 are enabled.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note : * Please contact a Renesas Electronics sales office for details of the optional function.

PFKUSB is used to set I/O pins for the USB0.

USBMD[1:0] Bits (USB Mode Setting)

These bits select a mode for the USB.

Table 17.7 lists the relationship between the setting of the USBMD[1:0] bits and the mode of the USB.

PUPHZS Bit (PUPHZ Select)

This bit selects the output mode (external pull-up control/pin pull-up output) for the DPUPE pin of the USB.

When the PUPHZS bit is set to 0, the control signal output mode for the external pull-up IC is selected and the high-active control signal is output from the DPUPE pin. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the low-level output state.

When the PUPHZS bit is set to 1, the output mode where the DP pin is directly pulled-up is selected. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the high-impedance state.

PDHVS Bit (PDHZ Select)

This bit selects the output mode (external pull-down control/pin pull-down output) for the DPRPD and DRPD pins of the USB.

When the PDHVS bit is set to 0, the control signal output mode for the external pull-down IC is selected and the high-active control signal is output from the DPRPD and DRPD pins. When the DP and DM pins are pulled down, the DPRPD and DRPD pins are set to the high-level output state. When the pulled-down state of the DP and DM pins is canceled, the DPRPD and DRPD pins are set to the low-level output state.

When the PDHVS bit is set to 1, the output mode where the DP and DM pins are directly pulled-down is selected. When the DP and DM pins are pulled down, the DPRPD and DRPD pins are set to the low-level output state. When the pulled-down state of the DP and DM pins is canceled, the DPRPD and DRPD pins are set to the high-impedance state.

USBE Bit (USB Enable)

This bit enables all pin functions for the USB0.

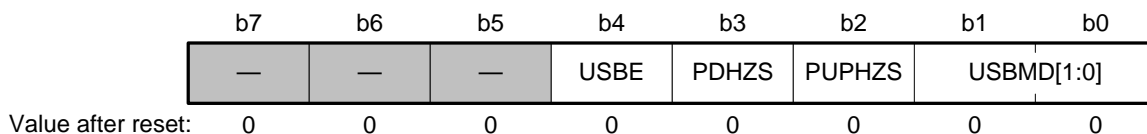
Table 17.7 Relationship between the USBMD[1:0] Bits Setting and USB Mode (USB0)

USBMD1	USBMD0	USB0 Mode	Pin to be Used for the USB	Pin Allocation	Remarks
0	0	Function mode	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_VBUS	P16	
			USB0_DPUPE-B	P14	Selection of -B side
0	1	Host mode	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_OVRCURA	P14	
			USB0_VBUSEN-B	P16	Selection of -B side
1	0	Host/function mode (as an optional function)*	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_OVRCURA	P14	
			USB0_VBUS	P16	
			USB0_DRPD	P22	
			USB0_DPUPE-A	P23	Selection of -A side
			USB0_VBUSEN-A	P24	Selection of -A side
1	1	OTG mode	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_OVRCURA	P14	
			USB0_OVRCURB	P16	
			USB0_DPRPD	P25	
			USB0_DRPD	P22	
			USB0_EXICEN	P21	
			USB0_ID	P20	
			USB0_DPUPE-A	P23	Selection of -A side
USB0_VBUSEN-A	P24	Selection of -A side			

Note : * Please contact a Renesas Electronics sales office for details of the optional function.

17.1.2.27 Port Function Control Register L (PFLUSB)

Address: 0008 C115h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	USBMD[1:0]	USB Mode Setting	b1 b0 0 0: Select function mode for the USB1 pins. 0 1: Select host mode for the USB1 pins. 1 0: Select host/function mode for the USB0 pin. (as an optional function)* 1 1: Select OTG mode for the USB1 pins.	R/W
b2	PUPHZS	PUPHZ Select	0: USB1_DPUPE pin is for output of the high and low levels (external pull-up control signal). 1: USB1_DPUPE pin is for high-level output or the Hi-Z state (pull-up output is from the USB1_DP pin).	R/W
b3	PDHZS	PDHZ Select	0: USB1_DPRPD and USB1_DRPD pins are for output of the high and low levels (external pull-down control signals). 1: USB1_DPRPD and USB1_DRPD pins are for low-level output or the Hi-Z state (pull-down output is from the USB1_DP and USB1_DM pins).	R/W
b4	USBE	USB Enable	0: All pin functions for USB1 are disabled. 1: All pin functions for USB1 are enabled.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note : * Please contact a Renesas Electronics sales office for details of the optional function.

PFLUSB is used to set I/O pins for the USB1.

USBMD[1:0] Bits (USB Mode Setting)

These bits select a mode for the USB.

Table 17.8 lists the relationship between the setting of the USBMD[1:0] bits and the mode of the USB.

PUPHZS Bit (PUPHZ Select)

This bit selects the output mode (external pull-up control/pin pull-up output) for the DPUPE pin of the USB.

When the PUPHZS bit is set to 0, the control signal output mode for the external pull-up IC is selected and the high-active control signal is output from the DPUPE pin. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the low-level output state.

When the PUPHZS bit is set to 1, the output mode where the DP pin is directly pulled-up is selected. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the high-impedance state.

PDHVS Bit (PDHZ Select)

This bit selects the output mode (external pull-down control/pin pull-down output) for the DPRPD and DRPD pins of the USB.

When the PDHVS bit is set to 0, the control signal output mode for the external pull-down IC is selected and the high-active control signal is output from the DPRPD and DRPD pins. When the DP and DM pins are pulled down, the DPRPD and DRPD pins are set to the high-level output state. When the pulled-down state of the DP and DM pins is canceled, the DPRPD and DRPD pins are set to the low-level output state.

When the PDHVS bit is set to 1, the output mode where the DP and DM pins are directly pulled-down is selected. When the DP and DM pins are pulled down, the DPRPD and DRPD pins are set to the low-level output state. When the pulled-down state of the DP and DM pins is canceled, the DPRPD and DRPD pins are set to the high-impedance state.

USBE Bit (USB Enable)

This bit enables all pin functions for the USB1.

Table 17.8 Relationship between the USBMD[1:0] Bits Setting and USB Mode (USB1)

USBMD1	USBMD0	USB1 Mode	Pin to be Used for the USB	Pin Allocation	Remark
0	0	Function mode	USB1_DP	USB1_DP	
			USB1_DM	USB1_DM	
			USB1_VBUS	P17	
			USB1_DPUPE-B	P15	Selection of -B side
0	1	Host mode	USB1_DP	USB1_DP	
			USB1_DM	USB1_DM	
			USB1_OVRCURA	P15	
			USB1_VBUSEN-B	P17	Selection of -B side
1	0	Host/function mode (as an optional function)*	USB1_DP	USB1_DP	
			USB1_DM	USB1_DM	
			USB1_OVRCURA	P15	
			USB1_VBUS	P17	
			USB1_DRPD	P30	
			USB1_DPUPE-A	P10	Selection of -A side
1	1	OTG mode	USB1_DP	USB1_DP	
			USB1_DM	USB1_DM	
			USB1_OVRCURA	P15	
			USB1_OVRCURB	P17	
			USB1_DPRPD	P31	
			USB1_DRPD	P30	
			USB1_EXICEN	P27	
			USB1_ID	P26	
USB1_DPUPE-A	P10	Selection of -A side			
			USB1_VBUSEN-A	P11	Selection of -A side

Note: * Please contact a Renesas Electronics sales office for details of the optional function.

17.1.2.28 Port Function Control Register M (PFMPOE)

Address: 0008 C116h

	b7	b6	b5	b4	b3	b2	b1	b0
	POE7E	POE6E	POE5E	POE4E	POE3E	POE2E	POE1E	POE0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	POE0E	POE0 Input Enable	0: Designated as an I/O port pin.	R/W*
b1	POE1E	POE1 Input Enable	1: Designated as the POEn# input pin (n = 0 to 7)	R/W*
b2	POE2E	POE2 Input Enable		R/W*
b3	POE3E	POE3 Input Enable		R/W*
b4	POE4E	POE4 Input Enable		R/W*
b5	POE5E	POE5 Input Enable		R/W*
b6	POE6E	POE6 Input Enable		R/W*
b7	POE7E	POE7 Input Enable		R/W*

Note : * The first round of writing after a reset is enabled; though subsequent writing is not allowed.

PFMPOE enables or disables POE input pins.

To prevent the error in the system, be sure to write to this register after a reset.

The first round of writing after a reset is only enabled.

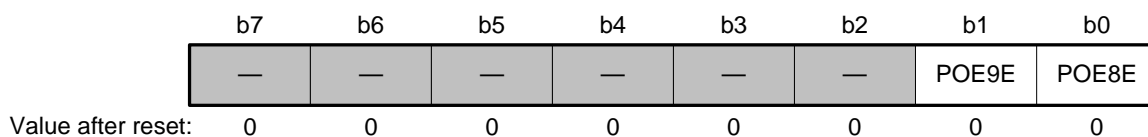
POEnE Bit (POEn Input Enable) (n = 0 to 7)

Each bit enables or disables the corresponding POEn# input.

To use POEn#, set the corresponding POEnE bit to 1.

17.1.2.29 Port Function Control Register N (PFNPOE)

Address: 0008 C117h



Bit	Symbol	Bit Name	Description	R/W
b0	POE8E	POE8 Input Enable	0: Designated as an I/O port pin.	R/W*
b1	POE9E	POE9 Input Enable	1: Designated as the POEn# input pin (n = 8, 9)	R/W*
b7 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note : * The first round of writing after a reset is enabled; though subsequent writing is not allowed.

PFNPOE enables or disables POE input pins.

To prevent the error in the system, be sure to write to this register after a reset.

The first round of writing after a reset is only enabled.

POEnE Bit (POEn Input Enable) (n = 8, 9)

Each bit enables or disables the corresponding POEn# input

To use POEn#, set the corresponding POEnE bit to 1.

17.1.3 Settings of Ports

When individual pins for peripheral modules are enabled, the settings for each port are modified.

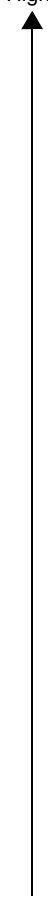
An input pin for the peripheral module is specified independently by the peripheral module. To use an input pin for the peripheral module, the corresponding bit in the input buffer control register (ICR) should be set to 1 to enable the input buffer, except for the port register read, data bus input, NMI, and POE pin inputs.

The pins that function as output pins and I/O pins should be enabled for respective peripheral modules. If a conflict occurs among the output signal enable settings for peripheral modules, that are multiplexed to the same port, the priority will be handled according to the port-multiplexed priority.

Table 17.9 lists the port-multiplexed priority for peripheral modules.

Table 17.10 lists the output enable settings for each port.

Table 17.9 Port-Multiplexed Priority for Peripheral Modules (176-Pin LFBGA)

Priority	Module Name	Output Pins	
High  Low	1	External bus (Data)	D0 to D31 (Data bus)
	2	External bus SDRAM	RD#, WR#, WR0# to WR3#, BC0# to BC3#, BCLK, SDCLK, SDCS#, RAS#, CAS#, WE#, CKE, DQM0 to DQM3, A0 to A23 (Address bus)
	3	External bus (CS)	CS0# to CS7# (Chip select)
	4	RSPI0, RSPI1	RSPCKn, MOSIn, MISON, SSLn0 to SSLn3 (n = A, B)
	5	USB0, USB1	USBm_DPUPE, USBm_VBUSEN, USBm_EXICEN, USBm_DRPD, USBm_DPRPD (m = 0, 1)
	6	CAN0	CTX0
	7	EtherNET	ET_MDC, ET_MDIO, ET_EXOUT, ET_WOL, ET_TX_EN, ET_TX_ER, ET_ETXD0 to ET_ETXD3, RMII_TXD_EN, RMII_TXD0, RMII_TXD1
	8	EXDMAC0, EXDMAC1	EDACK0, EDACK1
	9	MTU0 to MTU4, MTU6 to MTU10	MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D, MTIOC1A, MTIOC1B, MTIOC2A, MTIOC2B, MTIOC3A, MTIOC3B, MTIOC3C, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6A, MTIOC6B, MTIOC6C, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC8A, MTIOC8B, MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D, MTIOC10A, MTIOC10B, MTIOC10C, MTIOC10D
	10	TMR0 to TMR3	TMO0 to TMO3
	11	SCI0 to SCI3, SCI5 to SCI6	SCK0 to SCK3, SCK5 to SCK6, TxD0 to TxD3, TxD5 to TxD6
	12	RTC	RTCOU
	13	PPG0, PPG1	PO0 to PO15, PO16 to PO31
	14	RIIC0, RIIC1	SCL0 to SCL1, SDA0 to SDA1
	15	DA	DA0 to DA1
	16	I/O PORT	P00 to P03, P05, P07, P10 to P17, P20 to P27, P30 to P34, P50 to P52, P54 to P57, P60 to P67, P70 to P77, P80 to P85, P90 to P97, PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7, PF0 to PF4, PG0 to PG7

17.1.4 List of Output Enable Settings

Table 17.10 lists the output enable settings for each port.

For details on the applicable output signals, see descriptions of the registers for each peripheral module.

Setting the port function control register changes the functions of peripheral-module pins with names ending in A to D.

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (1 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P00	SCI6	TxD6-A	PFFSCI.SCI6S = 0	SCI6.SCR.TE = 1
	PORT0	P00		PORT0.DDR.B0 = 1
P01	PORT0	P01		PORT0.DDR.B1 = 1
P02	SCI6	SCK6-A	PFFSCI.SCI6S = 0	When SCI6.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI6.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PORT0	P02		PORT0.DDR.B2 = 1
P03	DA	DA0		DACR.DAOE0 = 1
	PORT0	P03		PORT0.DDR.B3 = 1
P05	DA	DA1		DACR.DAOE1 = 1
	PORT0	P05		PORT0.DDR.B5 = 1
P07	PORT0	P07		PORT0.DDR.B7 = 1
P10	USB1	USB1_DPUPE-A	PFLUSB.USBE = 1 PFLUSB.USBMD[1: 0] = 10/11	(The signal output state is specified by the peripheral module settings.)
	PORT1	P10		PORT1.DDR.B0 = 1
P11	USB1	USB1_VBUSEN-A	PFLUSB.USBE = 1 PFLUSB.USBMD[1: 0] = 10/11	(The signal output state is specified by the peripheral module settings.)
	SCI2	SCK2-A	PFFSCI.SCI2S = 0	When SCI2.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI2.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PORT1	P11		PORT1.DDR.B1 = 1
P12	RIIC0	SCL0		RIIC0.ICCR1.ICE = 1
	PORT1	P12		PORT1.DDR.B2 = 1
P13	TMR3	TMO3		TMO3.TCSR.OSA[1: 0] = 01/10/11 or TMO3.TCSR.OSB[1: 0] = 01/10/11
	SCI2	TxD2-A	PFFSCI.SCI2S = 0	SCI2.SCR.TE = 1
	RIIC0	SDA0		RIIC0.ICCR1.ICE = 1
	PORT1	P13		PORT1.DDR.B3 = 1
P14	USB0	USB0_DPUPE-B	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 00	(The signal output state is specified by the peripheral module settings.)
	PORT1	P14		PORT1.DDR.B4 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (2 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P15	USB1	USB1_DPUPE-B	PFLUSB.USBE = 1 PFLUSB.USBMD[1: 0] = 00	(The signal output state is specified by the peripheral module settings.)
	MTU0	MTIOC0B		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	SCI3	SCK3-A	PFFSCI.SCI3S = 0	When SCI3.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI3.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO13		PPG0.NDERH.NDER13 = 1
	PORT1	P15		PORT1.DDR.B5 = 1
P16	USB0	USB0_VBUSEN-B	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 01	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3C-A	PFCMTU.MTUS2 = 0	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	TMR2	TMO2		TMO2.TCSR.OSA[1: 0] = 01/10/11 or TMO2.TCSR.OSB[1: 0] = 01/10/11
	PPG0	PO14		PPG0.NDERH.NDER14 = 1
	PORT1	P16		PORT1.DDR.B6 = 1
P17	USB1	USB1_VBUSEN-B	PFLUSB.USBE = 1 PFLUSB.USBMD[1: 0] = 01	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3A		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	SCI3	TxD3-A	PFFSCI.SCI3S = 0	SCI3.SCR.TE = 1
	PPG0	PO15		PPG0.NDERH.NDER15 = 1
	PORT1	P17		PORT1.DDR.B7 = 1
P20	MTU1	MTIOC1A		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	SCI0	TxD0		SCI0.SCR.TE = 1
	PPG0	PO0		PPG0.NDERL.NDER0 = 1
	RIIC1	SDA1		RIIC1.ICCR1.ICE = 1
	PORT2	P20		PORT2.DDR.B0 = 1
P21	USB0	USB0_EXICEN	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 11	(The signal output state is specified by the peripheral module settings.)
	MTU1	MTIOC1B		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO1		PPG0.NDERL.NDER1 = 1
	RIIC1	SCL1		RIIC1.ICCR1.ICE = 1
	PORT2	P21		PORT2.DDR.B1 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (3 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P22	USB0	USB0_DRPD	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 10/11	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3B-A	PFCMTU.MTUS3 = 0	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	TMR0	TMO0		TMO0.TCSR.OSA[1: 0] = 01/10/11 or TMO0.TCSR.OSB[1: 0] = 01/10/11
	SCI0	SCK0		When SCI0.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI0.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO2		PPG0.NDERL.NDER2 = 1
	PORT2	P22		PORT2.DDR.B2 = 1
P23	USB0	USB0_DPUPE-A	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 10/11	(The signal output state is specified by the peripheral module settings.)
	EXDMAC0	EDACK0-B	PF7DMA.EDMA0S[1: 0] = 01	EXDMAC0.EDMOMD.DACKE = 1 (The signal output state is specified by the EXDMAC0 settings.)
	MTU3	MTIOC3D-A	PFCMTU.MTUS3 = 0	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	SCI3	TxD3-B	PFFSCI.SCI3S = 1	SCI3.SCR.TE = 1
	PPG0	PO3		PPG0.NDERL.NDER3 = 1
	PORT2	P23		PORT2.DDR.B3 = 1
P24	External bus (CS)	CS4#-C	PF0CSE.CS4E = 1 PF1CSS.CS4S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	USB0	USB0_VBUSEN-A	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 10/11	(The signal output state is specified by the peripheral module settings.)
	MTU4	MTIOC4A-A	PFCMTU.MTUS4 = 0	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	SCI3	SCK3-B	PFFSCI.SCI3S = 1	When SCI3.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI3.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO4		PPG0.NDERL.NDER4 = 1
	PORT2	P24		PORT2.DDR.B4 = 1
P25	External bus (CS)	CS5#-C	PF0CSE.CS5E = 1 PF1CSS.CS5S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	USB0	USB0_DPRPD	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 11	(The signal output state is specified by the peripheral module settings.)
	EXDMAC1	EDACK1-B	PF7DMA.EDMA1S[1: 0] = 01	EDMAC1.EDMOMD.DACKE = 1 (The signal output state is specified by the EXDMAC1 settings.)
	MTU4	MTIOC4C-A	PFCMTU.MTUS4 = 0	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO5		PPG0.NDERL.NDER5 = 1
	PORT2	P25		PORT2.DDR.B5 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (4 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P26	External bus (CS)	CS6#-C	PF0CSE.CS6E = 1 PF1CSS.CS6S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSP11	MOSIB-A	PFHSPI.MOSIE = 1 PFHSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU2	MTIOC2A		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	TMR1	TMO1		TMO1.TCSR.OSA[1: 0] = 01/10/11 or TMO1.TCSR.OSB[1: 0] = 01/10/11
	SCI1	TxD1-A	PF0SCI.SCI1S = 0	SCI1.SCR.TE = 1
	PPG0	PO6		PPG0.NDERL.NDER6 = 1
	PORT2	P26		PORT2.DDR.B6 = 1
P27	External bus (CS)	CS7#-C	PF0CSE.CS7E = 1 PF1CSS.CS7S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSP11	RSPCKB-A	PFHSPI.RSPCKE = 1 PFHSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	USB1	USB1_EXICEN	PFLUSB.USBE = 1 PFLUSB.USBMD[1: 0] = 11	(The signal output state is specified by the peripheral module settings.)
	MTU2	MTIOC2B		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	SCI1	SCK1-A	PF0SCI.SCI1S = 0	When SCI1.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI1.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO7		PPG0.NDERL.NDER7 = 1
	PORT2	P27		PORT2.DDR.B7 = 1
P30	RSP11	MISOB-A	PFHSPI.MISOE = 1 PFHSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	USB1	USB1_DRPD	PFLUSB.USBE = 1 PFLUSB.USBMD[1: 0] = 10/11	(The signal output state is specified by the peripheral module settings.)
	MTU4	MTIOC4B-A	PFCMTU.MTUS5 = 0	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO8		PPG0.NDERH.NDER8 = 1
	PORT3	P30		PORT3.DDR.B0 = 1
P31	RSP11	SSLB0-A	PFHSPI.SSL0E = 1 PFHSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	USB1	USB1_DPRPD	PFLUSB.USBE = 1 PFLUSB.USBMD[1: 0] = 11	(The signal output state is specified by the peripheral module settings.)
	MTU4	MTIOC4D-A	PFCMTU.MTUS5 = 0	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO9		PPG0.NDERH.NDER9 = 1
	PORT3	P31		PORT3.DDR.B1 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (5 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P32	CAN0	CTX0	PFJCAN.CAN0E = 1	(The signal output state is specified by the peripheral module settings.)
	MTU0	MTIOC0C		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	SCI6	TxD6-B	PFFSCI.SCI6S = 1	SCI6.SCR.TE = 1
	RTC	RTCOUT		RCR2.RTCOE = 1
	PPG0	PO10		PPG0.NDERH.NDER10 = 1
	PORT3	P32		PORT3.DDR.B2 = 1
P33	MTU0	MTIOC0D		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO11		PPG0.NDERH.NDER11 = 1
	PORT3	P33		PORT3.DDR.B3 = 1
P34	MTU0	MTIOC0A		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	SCI6	SCK6-B	PFFSCI.SCI6S = 1	When SCI6.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI6.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO12		PPG0.NDERH.NDER12 = 1
	PORT3	P34		PORT3.DDR.B4 = 1
P35	(NA)	(NA)		
P40	PORT4	P40		PORT4.DDR.B0 = 1
P41	PORT4	P41		PORT4.DDR.B1 = 1
P42	PORT4	P42		PORT4.DDR.B2 = 1
P43	PORT4	P43		PORT4.DDR.B3 = 1
P44	PORT4	P44		PORT4.DDR.B4 = 1
P45	PORT4	P45		PORT4.DDR.B5 = 1
P46	PORT4	P46		PORT4.DDR.B6 = 1
P47	PORT4	P47		PORT4.DDR.B7 = 1
P50	External bus	WR# WR0#		SYSCR0.EXBE = 1
	RSP11	SSLB1-A	PFHSPI.SSL1E = 1 PFHSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	SCI2	TxD2-B	PFFSCI.SCI2S = 1	SCI2.SCR.TE = 1
	PORT5	P50		PORT5.DDR.B0 = 1
P51	External bus	WR1# BC1#	PF5BUS.WR1BC1E = 1	SYSCR0.EXBE = 1
	RSP11	SSLB2-A	PFHSPI.SSL2E = 1 PFHSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	SCI2	SCK2-B	PFFSCI.SCI2S = 1	When SCI2.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI2.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PORT5	P51		PORT5.DDR.B1 = 1
P52	External bus	RD#	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB3-A	PFHSPI.SSL3E = 1 PFHSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	PORT5	P52		PORT5.DDR.B2 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (6 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P53	External bus	BCLK		PORT5.DDR.B3 = 1
P54	EXDMAC0	EDACK0-C	PF7DMA.EDMA0S[1: 0] = 11/10	EXDMAC0.EDMOMD.DACKE = 1 (The signal output state is specified by the EXDMAC0 settings.)
	MTU4	MTIOC4B-B	PFCMTU.MTUS5 = 1	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PORT5	P54		PORT5.DDR.B4 = 1
P55	EtherNET	ET_EXOUT	PFENET.EE = 1 PFENET.ENETE2 = 1	(The signal output state is specified by the peripheral module settings.)
	MTU4	MTIOC4D-B	PFCMTU.MTUS5 = 1	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PORT5	P55		PORT5.DDR.B5 = 1
P56	External bus	WR2# BC2#	PF5BUS.WR32BC32E = 1	SYSCR0.EXBE = 1
	EXDMAC1	EDACK1-C	PF7DMA.EDMA1S[1: 0] = 11/10	EDMAC1.EDMOMD.DACKE = 1 (The signal output state is specified by the EXDMAC1 settings.)
	MTU3	MTIOC3C-B	PFCMTU.MTUS2 = 1	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PORT5	P56		PORT5.DDR.B6 = 1
P57	External bus	WR3# BC3#	PF5BUS.WR32BC32E = 1	SYSCR0.EXBE = 1
	PORT5	P57		PORT5.DDR.B7 = 1
P60	External bus (CS)	CS0#-A	PF0CSE.CS0E = 1 PF2CSS.CS0S = 0	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P60		PORT6.DDR.B0 = 1
P61	SDRAM	SDCS#	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS1#-A	PF0CSE.CS1E = 1 PF2CSS.CS1S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P61		PORT6.DDR.B1 = 1
P62	SDRAM	RAS#	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS2#-A	PF0CSE.CS2E = 1 PF2CSS.CS2S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P62		PORT6.DDR.B2 = 1
P63	SDRAM	CAS#	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS3#-A	PF0CSE.CS3E = 1 PF2CSS.CS3S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P63		PORT6.DDR.B3 = 1
P64	SDRAM	WE#	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS4#-A	PF0CSE.CS4E = 1 PF1CSS.CS4S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P64		PORT6.DDR.B4 = 1
P65	SDRAM	CKE	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS5#-A	PF0CSE.CS5E = 1 PF1CSS.CS5S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P65		PORT6.DDR.B5 = 1
P66	SDRAM	DQM0	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS6#-A	PF0CSE.CS6E = 1 PF1CSS.CS6S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P66		PORT6.DDR.B6 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (7 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P67	SDRAM	DQM1	PF6BUS.MDSDE = 1 PF6BUS.DQM1E = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS7#-A	PF0CSE.CS7E = 1 PF1CSS.CS7S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P67		PORT6.DDR.B7 = 1
P70	SDRAM	SDCLK	PF6BUS.SDCLKE = 1	
	PORT7	P70		PORT7.DDR.B0 = 1
P71	External bus (CS)	CS1#-B	PF0CSE.CS1E = 1 PF2CSS.CS1S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	EtherNET	ET_MDIO	PFENET.EE = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORT7	P71		PORT7.DDR.B1 = 1
P72	External bus (CS)	CS2#-B	PF0CSE.CS2E = 1 PF2CSS.CS2S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	EtherNET	ET_MDC	PFENET.EE = 1	(The signal output state is specified by the peripheral module settings.)
	PORT7	P72		PORT7.DDR.B2 = 1
P73	External bus (CS)	CS3#-B	PF0CSE.CS3E = 1 PF2CSS.CS3S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	EtherNET	ET_WOL	PFENET.EE = 1 PFENET.ENETE0 = 1	(The signal output state is specified by the peripheral module settings.)
	PORT7	P73		PORT7.DDR.B3 = 1
P74	External bus (CS)	CS4#-B	PF0CSE.CS4E = 1 PF1CSS.CS4S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT7	P74		PORT7.DDR.B4 = 1
P75	External bus (CS)	CS5#-B	PF0CSE.CS5E = 1 PF1CSS.CS5S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT7	P75		PORT7.DDR.B5 = 1
P76	External bus (CS)	CS6#-B	PF0CSE.CS6E = 1 PF1CSS.CS6S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT7	P76		PORT7.DDR.B6 = 1
P77	External bus (CS)	CS7#-B	PF0CSE.CS7E = 1 PF1CSS.CS7S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT7	P77		PORT7.DDR.B7 = 1
P80	EtherNET	ET_TX_EN	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	EtherNET	RMII_TXD_EN	PFENET.EE = 1 PFENET.PHYMODE = 0	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3B-B	PFCMTU.MTUS3 = 1	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PORT8	P80		PORT8.DDR.B0 = 1
P81	EtherNET	ET_ETXD0	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	EtherNET	RMII_TXD0	PFENET.EE = 1 PFENET.PHYMODE = 0	(The signal output state is specified by the peripheral module settings.)
	EXDMAC0	EDACK0-A	PF7DMA.EDMA0S[1: 0] = 00	EXDMAC0.EDMOMD.DACKE = 1 (The signal output state is specified by the EXDMAC0 settings.)
	MTU3	MTIOC3D-B	PFCMTU.MTUS3 = 1	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PORT8	P81		PORT8.DDR.B1 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (8 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P82	EtherNET	ET_ETXD1	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	EtherNET	RMII_TXD1	PFENET.EE = 1 PFENET.PHYMODE = 0	(The signal output state is specified by the peripheral module settings.)
	MTU4	MTIOC4A-B	PFCMTU.MTUS4 = 1	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PORT8	P82		PORT8.DDR.B2 = 1
P83	EXDMAC1	EDACK1-A	PF7DMA.EDMA1S[1: 0] = 00	EDMAC1.EDMOMD.DACKE = 1 (The signal output state is specified by the EXDMAC1 settings.)
	MTU4	MTIOC4C-B	PFCMTU.MTUS4 = 1	For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PORT8	P83		PORT8.DDR.B3 = 1
P84	PORT8	P84		PORT8.DDR.B4 = 1
P85	PORT8	P85		PORT8.DDR.B5 = 1
P90	External bus (Data)	D16	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	External bus	A16-B	PF3BUS.A16E = 1 PF5BUS.ADRHMS = 1	SYSCR0.EXBE = 1
	PORT9	P90		PORT9.DDR.B0 = 1
P91	External bus (Data)	D17	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	External bus	A17-B	PF3BUS.A17E = 1 PF5BUS.ADRHMS = 1	SYSCR0.EXBE = 1
	PORT9	P91		PORT9.DDR.B1 = 1
P92	External bus (Data)	D18	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	External bus	A18-B	PF3BUS.A18E = 1 PF5BUS.ADRHMS = 1	SYSCR0.EXBE = 1
	PORT9	P92		PORT9.DDR.B2 = 1
P93	External bus (Data)	D19	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	External bus	A19-B	PF3BUS.A19E = 1 PF5BUS.ADRHMS = 1	SYSCR0.EXBE = 1
	PORT9	P93		PORT9.DDR.B3 = 1
P94	External bus (Data)	D20	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	External bus	A20-B	PF3BUS.A20E = 1 PF5BUS.ADRHMS = 1	SYSCR0.EXBE = 1
	PORT9	P94		PORT9.DDR.B4 = 1
P95	External bus (Data)	D21	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	External bus	A21-B	PF3BUS.A21E = 1 PF5BUS.ADRHMS = 1	SYSCR0.EXBE = 1
	PORT9	P95		PORT9.DDR.B5 = 1
P96	External bus (Data)	D22	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	External bus	A22-B	PF3BUS.A22E = 1 PF5BUS.ADRHMS = 1	SYSCR0.EXBE = 1
	PORT9	P96		PORT9.DDR.B6 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (9 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P97	External bus (Data)	D23	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	External bus	A23-B	PF3BUS.A23E = 1 PF5BUS.ADRHMS = 1	SYSCR0.EXBE = 1
	PORT9	P97		PORT9.DDR.B7 = 1
PA0	External bus SDRAM	A0 BC0# DQM2	PF4BUS.ADRLE[1: 0] = 01/ 10/11	SYSCR0.EXBE = 1
	RSP10	SSLA1-B	PFGSPI.SSL1E = 1 PFGSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	MTU6	MTIOC6A		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO16		PPG1.NDERL.NDER0 = 1
	PORTA	PA0		PORTA.DDR.B0 = 1
PA1	External bus SDRAM	A1 DQM3	PF4BUS.ADRLE[1: 0] = 01/ 10/11	SYSCR0.EXBE = 1
	RSP10	SSLA2-B	PFGSPI.SSL2E = 1 PFGSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	MTU6	MTIOC6B		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO17		PPG1.NDERL.NDER1 = 1
	PORTA	PA1		PORTA.DDR.B1 = 1
PA2	External bus	A2	PF4BUS.ADRLE[1: 0] = 01/ 10/11	SYSCR0.EXBE = 1
	RSP10	SSLA3-B	PFGSPI.SSL3E = 1 PFGSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	MTU6	MTIOC6C		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO18		PPG1.NDERL.NDER2 = 1
	PORTA	PA2		PORTA.DDR.B2 = 1
PA3	External bus	A3	PF4BUS.ADRLE[1: 0] = 01/ 10/11	SYSCR0.EXBE = 1
	MTU6	MTIOC6D		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO19		PPG1.NDERL.NDER3 = 1
	PORTA	PA3		PORTA.DDR.B3 = 1
PA4	External bus	A4	PF4BUS.ADRLE[1: 0] = 10/ 11	SYSCR0.EXBE = 1
	RSP10	SSLA0-B	PFGSPI.SSL0E = 1 PFGSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU7	MTIOC7A		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO20		PPG1.NDERL.NDER4 = 1
	PORTA	PA4		PORTA.DDR.B4 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (10 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PA5	External bus	A5	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPI0	RSPCKA-B	PFGSPI.RSPCKE = 1 PFGSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU7	MTIOC7B		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO21		PPG1.NDERL.NDER5 = 1
	PORTA	PA5		PORTA.DDR.B5 = 1
PA6	External bus	A6	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPI0	MOSIA-B	PFGSPI.MOSIE = 1 PFGSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU8	MTIOC8A		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO22		PPG1.NDERL.NDER6 = 1
	PORTA	PA6		PORTA.DDR.B6 = 1
PA7	External bus	A7	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPI0	MISOA-B	PFGSPI.MISOE = 1 PFGSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU8	MTIOC8B		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO23		PPG1.NDERL.NDER7 = 1
	PORTA	PA7		PORTA.DDR.B7 = 1
PB0	External bus	A8	PF4BUS.ADRLE[1: 0] = 11	SYSCR0.EXBE = 1
	MTU9	MTIOC9A		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO24		PPG1.NDERH.NDER8 = 1
	PORTB	PB0		PORTB.DDR.B0 = 1
PB1	External bus	A9	PF4BUS.ADRLE[1: 0] = 11	SYSCR0.EXBE = 1
	MTU9	MTIOC9C		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO25		PPG1.NDERH.NDER9 = 1
	PORTB	PB1		PORTB.DDR.B1 = 1
PB2	External bus	A10	PF4BUS.A10E = 1	SYSCR0.EXBE = 1
	MTU9	MTIOC9B		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO26		PPG1.NDERH.NDER10 = 1
	PORTB	PB2		PORTB.DDR.B2 = 1
PB3	External bus	A11	PF4BUS.A11E = 1	SYSCR0.EXBE = 1
	MTU9	MTIOC9D		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO27		PPG1.NDERH.NDER11 = 1
	PORTB	PB3		PORTB.DDR.B3 = 1
PB4	External bus	A12	PF4BUS.A12E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10A		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO28		PPG1.NDERH.NDER12 = 1
	PORTB	PB4		PORTB.DDR.B4 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (11 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PB5	External bus	A13	PF4BUS.A13E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10C		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO29		PPG1.NDERH.NDER13 = 1
	PORTB	PB5		PORTB.DDR.B5 = 1
PB6	External bus	A14	PF4BUS.A14E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10B		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO30		PPG1.NDERH.NDER14 = 1
	PORTB	PB6		PORTB.DDR.B6 = 1
PB7	External bus	A15	PF4BUS.A15E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10D		For the MTU settings, see Table 17.11, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO31		PPG1.NDERH.NDER15 = 1
	PORTB	PB7		PORTB.DDR.B7 = 1
PC0	External bus	A16-A	PF3BUS.A16E = 1 PF5BUS.ADRHMS = 0	SYSCR0.EXBE = 1
	RSPI0	SSLA1-A	PFGSPI.SSL1E = 1 PFGSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	PORTC	PC0		PORTC.DDR.B0 = 1
PC1	External bus	A17-A	PF3BUS.A17E = 1 PF5BUS.ADRHMS = 0	SYSCR0.EXBE = 1
	RSPI0	SSLA2-A	PFGSPI.SSL2E = 1 PFGSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	SCI5	SCK5		When SCI5.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1:0] = 01 or SMR.GM = 1 When SCI5.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1:0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PORTC	PC1		PORTC.DDR.B1 = 1
PC2	External bus	A18-A	PF3BUS.A18E = 1 PF5BUS.ADRHMS = 0	SYSCR0.EXBE = 1
	RSPI0	SSLA3-A	PFGSPI.SSL3E = 1 PFGSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	PORTC	PC2		PORTC.DDR.B2 = 1
PC3	External bus	A19-A	PF3BUS.A19E = 1 PF5BUS.ADRHMS = 0	SYSCR0.EXBE = 1
	EtherNET	ET_TX_ER	PFENET.EE = 1 PFENET.ENETE3 = 1	(The signal output state is specified by the peripheral module settings.)
	SCI5	TxD5		SCI5.SCR.TE = 1
	PORTC	PC3		PORTC.DDR.B3 = 1
PC4	External bus	A20-A	PF3BUS.A20E = 1 PF5BUS.ADRHMS = 0	SYSCR0.EXBE = 1
	External bus (CS)	CS3#-C	PF0CSE.CS3E = 1 PF2CSS.CS3S[1:0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSPI0	SSLA0-A	PFGSPI.SSL0E = 1 PFGSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTC	PC4		PORTC.DDR.B4 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (12 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PC5	External bus	A21-A	PF3BUS.A21E = 1 PF5BUS.ADRHMS = 0	SYSCR0.EXBE = 1
	External bus (CS)	CS2#-C	PF0CSE.CS2E = 1 PF2CSS.CS2S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSPI0	RSPCKA-A	PFGSPI.RSPCKE = 1 PFGSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	EtherNET	ET_ETXD2	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	PORTC	PC5		PORTC.DDR.B5 = 1
PC6	External bus	A22-A	PF3BUS.A22E = 1 PF5BUS.ADRHMS = 0	SYSCR0.EXBE = 1
	External bus (CS)	CS1#-C	PF0CSE.CS1E = 1 PF2CSS.CS1S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSPI0	MOSIA-A	PFGSPI.MOSIE = 1 PFGSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	EtherNET	ET_ETXD3	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	PORTC	PC6		PORTC.DDR.B6 = 1
PC7	External bus	A23-A	PF3BUS.A23E = 1 PF5BUS.ADRHMS = 0	SYSCR0.EXBE = 1
	External bus (CS)	CS0#-B	PF0CSE.CS0E = 1 PF2CSS.CS0S = 1	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSPI0	MISOA-A	PFGSPI.MISOE = 1 PFGSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTC	PC7		PORTC.DDR.B7 = 1
PD0	External bus (Data)	D0		SYSCR0.EXBE = 1
	PORTD	PD0		PORTD.DDR.B0 = 1
PD1	External bus (Data)	D1		SYSCR0.EXBE = 1
	PORTD	PD1		PORTD.DDR.B1 = 1
PD2	External bus (Data)	D2		SYSCR0.EXBE = 1
	PORTD	PD2		PORTD.DDR.B2 = 1
PD3	External bus (Data)	D3		SYSCR0.EXBE = 1
	PORTD	PD3		PORTD.DDR.B3 = 1
PD4	External bus (Data)	D4		SYSCR0.EXBE = 1
	PORTD	PD4		PORTD.DDR.B4 = 1
PD5	External bus (Data)	D5		SYSCR0.EXBE = 1
	PORTD	PD5		PORTD.DDR.B5 = 1
PD6	External bus (Data)	D6		SYSCR0.EXBE = 1
	PORTD	PD6		PORTD.DDR.B6 = 1
PD7	External bus (Data)	D7		SYSCR0.EXBE = 1
	PORTD	PD7		PORTD.DDR.B7 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (13 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PE0	External bus (Data)	D8	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB1-B	PFHSPI.SSL1E = 1 PFHSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	PORTE	PE0		PORTE.DDR.B0 = 1
PE1	External bus (Data)	D9	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB2-B	PFHSPI.SSL2E = 1 PFHSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	PORTE	PE1		PORTE.DDR.B1 = 1
PE2	External bus (Data)	D10	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB3-B	PFHSPI.SSL3E = 1 PFHSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	PORTE	PE2		PORTE.DDR.B2 = 1
PE3	External bus (Data)	D11	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	PORTE	PE3		PORTE.DDR.B3 = 1
PE4	External bus (Data)	D12	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB0-B	PFHSPI.SSL0E = 1 PFHSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTE	PE4		PORTE.DDR.B4 = 1
PE5	External bus (Data)	D13	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	RSPCKB-B	PFHSPI.RSPCKE = 1 PFHSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTE	PE5		PORTE.DDR.B5 = 1
PE6	External bus (Data)	D14	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	MOSIB-B	PFHSPI.MOSIE = 1 PFHSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTE	PE6		PORTE.DDR.B6 = 1
PE7	External bus (Data)	D15	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	MISOB-B	PFHSPI.MISOE = 1 PFHSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTE	PE7		PORTE.DDR.B7 = 1
PF0	SCI1	TxD1-B	PFFSCI.SCI1S = 1	SCI1.SCR.TE = 1
	PORTF	PF0		PORTF.DDR.B0 = 1
PF1	SCI1	SCK1-B	PFFSCI.SCI1S = 1	When SCI1.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1:0] = 01 or SMR.GM = 1 When SCI1.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1:0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PORTF	PF1		PORTF.DDR.B1 = 1
PF2	PORTF	PF2		PORTF.DDR.B2 = 1
PF3	PORTF	PF3		PORTF.DDR.B3 = 1
PF4	PORTF	PF4		PORTF.DDR.B4 = 1

Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA) (14 / 14)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PG0	External bus (Data)	D24	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	PORTG	PG0		PORTG.DDR.B0 = 1
PG1	External bus (Data)	D25	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	PORTG	PG1		PORTG.DDR.B1 = 1
PG2	External bus (Data)	D26	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	PORTG	PG2		PORTG.DDR.B2 = 1
PG3	External bus (Data)	D27	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	PORTG	PG3		PORTG.DDR.B3 = 1
PG4	External bus (Data)	D28	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	PORTG	PG4		PORTG.DDR.B4 = 1
PG5	External bus (Data)	D29	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	PORTG	PG5		PORTG.DDR.B5 = 1
PG6	External bus (Data)	D30	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	PORTG	PG6		PORTG.DDR.B6 = 1
PG7	External bus (Data)	D31	PF5BUS.DH32E = 1	SYSCR0.EXBE = 1
	PORTG	PG7		PORTG.DDR.B7 = 1

Table 17.11 Settings to Enable Output on the Various MTU Pins (1 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins	
MTU0	MTIOC0A	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TIORH.IOA[3] = 0 MTU0.TIORH.IOA[1:0] = 01/10/11	
		PWM mode 1	MTU0.TMDR.MD[3:0] = 0010	MTU0.TIORH.IOA[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	Except MTU0.TCR.CCLR[2:0] = 001 MTU0.TIORH.IOA[1:0] = 01/10/11	
	MTIOC0B	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TIORH.IOB[3] = 0 MTU0.TIORH.IOB[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	Except MTU0.TCR.CCLR[2:0] = 010 MTU0.TIORH.IOB[1:0] = 01/10/11	
	MTIOC0C	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TMDR.BFA = 0 MTU0.TIORL.IOC[3] = 0 MTU0.TIORL.IOC[1:0] = 01/10/11	
		PWM mode 1	MTU0.TMDR.MD[3:0] = 0010	MTU0.TMDR.BFA = 0 MTU0.TMDR.BFB = 0 MTU0.TIORL.IOC[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	MTU0.TMDR.BFA = 0 Except MTU0.TCR.CCLR[2:0] = 101 MTU0.TIORL.IOC[1:0] = 01/10/11	
	MTIOC0D	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TMDR.BFB = 0 MTU0.TIORL.IOD[3] = 0 MTU0.TIORL.IOD[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	MTU0.TMDR.BFB = 0 Except MTU0.TCR.CCLR[2:0] = 110 MTU0.TIORL.IOD[1:0] = 01/10/11	
	MTU1	MTIOC1A	Normal operation	MTU1.TMDR.MD[3:0] = 0000	MTU1.TIOR.IOA[3] = 0 MTU1.TIOR.IOA[1:0] = 01/10/11
			PWM mode 1	MTU1.TMDR.MD[3:0] = 0010	MTU1.TIOR.IOA[1:0] = 01/10/11
PWM mode 2			MTU1.TMDR.MD[3:0] = 0011	Except MTU1.TCR.CCLR[1:0] = 01 MTU1.TIOR.IOA[1:0] = 01/10/11	
Phase count mode 1			MTU1.TMDR.MD[3:0] = 0100	MTU1.TIOR.IOA[3] = 0 MTU1.TIOR.IOA[1:0] = 01/10/11	
Phase count mode 2			MTU1.TMDR.MD[3:0] = 0101		
Phase count mode 3			MTU1.TMDR.MD[3:0] = 0110		
Phase count mode 4			MTU1.TMDR.MD[3:0] = 0111		
MTIOC1B		Normal operation	MTU1.TMDR.MD[3:0] = 0000	MTU1.TIOR.IOB[3] = 0 MTU1.TIOR.IOB[1:0] = 01/10/11	
		PWM mode 2	MTU1.TMDR.MD[3:0] = 0011	Except MTU1.TCR.CCLR[1:0] = 10 MTU1.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 1	MTU1.TMDR.MD[3:0] = 0100	MTU1.TIOR.IOB[3] = 0 MTU1.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 2	MTU1.TMDR.MD[3:0] = 0101		
		Phase count mode 3	MTU1.TMDR.MD[3:0] = 0110		
		Phase count mode 4	MTU1.TMDR.MD[3:0] = 0111		
MTU2		MTIOC2A	Normal operation	MTU2.TMDR.MD[3:0] = 0000	MTU2.TIOR.IOA[3] = 0 MTU2.TIOR.IOA[1:0] = 01/10/11
			PWM mode 1	MTU2.TMDR.MD[3:0] = 0010	MTU2.TIOR.IOA[1:0] = 01/10/11
	PWM mode 2		MTU2.TMDR.MD[3:0] = 0011	Except MTU2.TCR.CCLR[1:0] = 01 MTU2.TIOR.IOA[1:0] = 01/10/11	
	Phase count mode 1		MTU2.TMDR.MD[3:0] = 0100	MTU2.TIOR.IOA[3] = 0 MTU2.TIOR.IOA[1:0] = 01/10/11	
	Phase count mode 2		MTU2.TMDR.MD[3:0] = 0101		
	Phase count mode 3		MTU2.TMDR.MD[3:0] = 0110		
	Phase count mode 4		MTU2.TMDR.MD[3:0] = 0111		

Table 17.11 Settings to Enable Output on the Various MTU Pins (2 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins	
MTU2	MTIOC2B	Normal operation	MTU2.TMDR.MD[3:0] = 0000	MTU2.TIOR.IOB[3] = 0 MTU2.TIOR.IOB[1:0] = 01/10/11	
		PWM mode 2	MTU2.TMDR.MD[3:0] = 0011	Except MTU2.TCR.CCLR[1:0] = 10 MTU2.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 1	MTU2.TMDR.MD[3:0] = 0100	MTU2.TIOR.IOB[3] = 0	
		Phase count mode 2	MTU2.TMDR.MD[3:0] = 0101	MTU2.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 3	MTU2.TMDR.MD[3:0] = 0110		
		Phase count mode 4	MTU2.TMDR.MD[3:0] = 0111		
MTU3	MTIOC3A	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTU3.TIORH.IOA[3] = 0 MTU3.TIORH.IOA[1:0] = 01/10/11	
		PWM mode 1	MTU3.TMDR.MD[3:0] = 0010	MTU3.TIORH.IOA[1:0] = 01/10/11	
		Reset-synchronized PWM mode	MTU3.TMDR.MD[3:0] = 1000	MTUA.TOCCR1.PSYE = 1	
		Complementary PWM mode 1	MTU3.TMDR.MD[3:0] = 1101		
		Complementary PWM mode 2	MTU3.TMDR.MD[3:0] = 1110		
		Complementary PWM mode 3	MTU3.TMDR.MD[3:0] = 1111		
	MTIOC3B	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTUA.TOER.OE3B = 1 MTU3.TIORH.IOB[3] = 0 MTU3.TIORH.IOB[1:0] = 01/10/11	
		Reset-synchronized PWM mode	MTU3.TMDR.MD[3:0] = 1000	MTUA.TOER.OE3B = 1	
		Complementary PWM mode 1	MTU3.TMDR.MD[3:0] = 1101		
		Complementary PWM mode 2	MTU3.TMDR.MD[3:0] = 1110		
		Complementary PWM mode 3	MTU3.TMDR.MD[3:0] = 1111		
		MTIOC3C	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTU3.TMDR.BFA = 0 MTU3.TIORL.IOC[3] = 0 MTU3.TIORL.IOC[1:0] = 01/10/11
	PWM mode 1		MTU3.TMDR.MD[3:0] = 0010	MTU3.TMDR.BFA = 0 MTU3.TMDR.BFB = 0 MTU3.TIORL.IOC[1:0] = 01/10/11	
	MTIOC3D	Normal operation	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTUA.TOER.OE3D = 1 MTU3.TMDR.BFB = 0 MTU3.TIORL.IOD[3] = 0 MTU3.TIORL.IOD[1:0] = 01/10/11
			Reset-synchronized PWM mode	MTU3.TMDR.MD[3:0] = 1000	MTUA.TOER.OE3D = 1
Complementary PWM mode 1			MTU3.TMDR.MD[3:0] = 1101		
Complementary PWM mode 2			MTU3.TMDR.MD[3:0] = 1110		
Complementary PWM mode 3			MTU3.TMDR.MD[3:0] = 1111		
MTU4			MTIOC4A	Normal operation	MTU4.TMDR.MD[3:0] = 0000
	PWM mode 1	MTU4.TMDR.MD[3:0] = 0010		MTUA.TOER.OE4A = 1 MTU4.TIORH.IOA[1:0] = 01/10/11	
	– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000		MTUA.TOER.OE4A = 1	
	– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101			
	– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110			
	– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111			
	Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000		MTUA.TOER.OE4A = 1	
	Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000			
	Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000			
	Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000			

Table 17.11 Settings to Enable Output on the Various MTU Pins (3 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins
MTU4	MTIOC4B	Normal operation	MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4B = 1 MTU4.TIORH.IOB[3] = 0 MTU4.TIORH.IOB[1:0] = 01/10/11
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000	MTUA.TOER.OE4B = 1
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4B = 1
		Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000	
		MTIOC4C	MTIOC4C	Normal operation
PWM mode 1	MTU4.TMDR.MD[3:0] = 0010			MTUA.TOER.OE4C = 1 MTU4.TMDR.BFA = 0 MTU4.TMDR.BFB = 0 MTU4.TIORL.IOC[1:0] = 01/10/11
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000			MTUA.TOER.OE4C = 1 MTU4.TMDR.BFA = 0
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101			
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110			
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111			
Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000			MTUA.TOER.OE4C = 1 MTU4.TMDR.BFA = 0
Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000			
Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000			
Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000			
MTIOC4D	MTIOC4D	Normal operation	MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4D = 1 MTU4.TMDR.BFB = 0 MTU4.TIORL.IOD[3] = 0 MTU4.TIORL.IOD[1:0] = 01/10/11
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000	MTUA.TOER.OE4D = 1
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4D = 1
		Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000	

Table 17.11 Settings to Enable Output on the Various MTU Pins (4 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins		
MTU6	MTIOC6A	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TIORH.IOA[3] = 0 MTU6.TIORH.IOA[1:0] = 01/10/11		
		PWM mode 1	MTU6.TMDR.MD[3:0] = 0010	MTU6.TIORH.IOA[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	Except MTU6.TCR.CCLR[2:0] = 001 MTU6.TIORH.IOA[1:0] = 01/10/11		
	MTIOC6B	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TIORH.IOB[3] = 0 MTU6.TIORH.IOB[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	Except MTU6.TCR.CCLR[2:0] = 010 MTU6.TIORH.IOB[1:0] = 01/10/11		
	MTIOC6C	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TMDR.BFA = 0 MTU6.TIORL.IOC[3] = 0 MTU6.TIORL.IOC[1:0] = 01/10/11		
		PWM mode 1	MTU6.TMDR.MD[3:0] = 0010	MTU6.TMDR.BFA = 0 MTU6.TMDR.BFB = 0 MTU6.TIORL.IOC[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	MTU6.TMDR.BFA = 0 Except MTU6.TCR.CCLR[2:0] = 101 MTU6.TIORL.IOC[1:0] = 01/10/11		
	MTIOC6D	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TMDR.BFB = 0 MTU6.TIORL.IOD[3] = 0 MTU6.TIORL.IOD[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	MTU6.TMDR.BFB = 0 Except MTU6.TCR.CCLR[2:0] = 110 MTU6.TIORL.IOD[1:0] = 01/10/11		
	MTU7	MTIOC7A	Normal operation	MTU7.TMDR.MD[3:0] = 0000	MTU7.TIOR.IOA[3] = 0 MTU7.TIOR.IOA[1:0] = 01/10/11	
			PWM mode 1	MTU7.TMDR.MD[3:0] = 0010	MTU7.TIOR.IOA[1:0] = 01/10/11	
PWM mode 2			MTU7.TMDR.MD[3:0] = 0011	Except MTU7.TCR.CCLR[1:0] = 01 MTU7.TIOR.IOA[1:0] = 01/10/11		
Phase count mode			MTU7.TMDR.MD[3:0] = 0100	MTU7.TIOR.IOA[3] = 0 MTU7.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 2			MTU7.TMDR.MD[3:0] = 0101			
Phase count mode 3			MTU7.TMDR.MD[3:0] = 0110			
Phase count mode 4			MTU7.TMDR.MD[3:0] = 0111			
MTIOC7B		Normal operation	MTU7.TMDR.MD[3:0] = 0000	MTU7.TIOR.IOB[3] = 0 MTU7.TIOR.IOB[1:0] = 01/10/11		
		PWM mode 2	MTU7.TMDR.MD[3:0] = 0011	Except MTU7.TCR.CCLR[1:0] = 10 MTU7.TIOR.IOB[1:0] = 01/10/11		
		Phase count mode 1	MTU7.TMDR.MD[3:0] = 0100	MTU7.TIOR.IOB[3] = 0 MTU7.TIOR.IOB[1:0] = 01/10/11		
		Phase count mode 2	MTU7.TMDR.MD[3:0] = 0101			
		Phase count mode 3	MTU7.TMDR.MD[3:0] = 0110			
		Phase count mode 4	MTU7.TMDR.MD[3:0] = 0111			
		MTU8	MTIOC8A	Normal operation	MTU8.TMDR.MD[3:0] = 0000	MTU8.TIOR.IOA[3] = 0 MTU8.TIOR.IOA[1:0] = 01/10/11
				PWM mode 1	MTU8.TMDR.MD[3:0] = 0010	MTU8.TIOR.IOA[1:0] = 01/10/11
PWM mode 2	MTU8.TMDR.MD[3:0] = 0011			Except MTU8.TCR.CCLR[1:0] = 01 MTU8.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 1	MTU8.TMDR.MD[3:0] = 0100			MTU8.TIOR.IOA[3] = 0 MTU8.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 2	MTU8.TMDR.MD[3:0] = 0101					
Phase count mode 3	MTU8.TMDR.MD[3:0] = 0110					
Phase count mode 4	MTU8.TMDR.MD[3:0] = 0111					

Table 17.11 Settings to Enable Output on the Various MTU Pins (5 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins
MTU8	MTIOC8B	Normal operation	MTU8.TMDR.MD[3:0] = 0000	MTU8.TIOR.IOB[3] = 0 MTU8.TIOR.IOB[1:0] = 01/10/11
		PWM mode 2	MTU8.TMDR.MD[3:0] = 0011	Except MTU8.TCR.CCLR[1:0] = 10 MTU8.TIOR.IOB[1:0] = 01/10/11
		Phase count mode 1	MTU8.TMDR.MD[3:0] = 0100	MTU8.TIOR.IOB[3] = 0
		Phase count mode 2	MTU8.TMDR.MD[3:0] = 0101	MTU8.TIOR.IOB[1:0] = 01/10/11
		Phase count mode 3	MTU8.TMDR.MD[3:0] = 0110	
		Phase count mode 4	MTU8.TMDR.MD[3:0] = 0111	
MTU9	MTIOC9A	Normal operation	MTU9.TMDR.MD[3:0] = 0000	MTU9.TIORH.IOA[3] = 0 MTU9.TIORH.IOA[1:0] = 01/10/11
		PWM mode 1	MTU9.TMDR.MD[3:0] = 0010	MTU9.TIORH.IOA[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU9.TMDR.MD[3:0] = 1000	MTUB.TOER1.PSYE = 1
		Complementary PWM mode 1	MTU9.TMDR.MD[3:0] = 1101	
		Complementary PWM mode 2	MTU9.TMDR.MD[3:0] = 1110	
		Complementary PWM mode 3	MTU9.TMDR.MD[3:0] = 1111	
	MTIOC9B	Normal operation	MTU9.TMDR.MD[3:0] = 0000	MTUB.TOER.OE3B = 1 MTU9.TIORH.IOB[3] = 0 MTU9.TIORH.IOB[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU9.TMDR.MD[3:0] = 1000	MTUB.TOER.OE3B = 1
		Complementary PWM mode 1	MTU9.TMDR.MD[3:0] = 1101	
		Complementary PWM mode 2	MTU9.TMDR.MD[3:0] = 1110	
		Complementary PWM mode 3	MTU9.TMDR.MD[3:0] = 1111	
		MTIOC9C	Normal operation	MTU9.TMDR.MD[3:0] = 0000
	PWM mode 1		MTU9.TMDR.MD[3:0] = 0010	MTU9.TMDR.BFA = 0 MTU9.TMDR.BFB = 0 MTU9.TIORL.IOC[1:0] = 01/10/11
	MTIOC9D	Normal operation	MTU9.TMDR.MD[3:0] = 0000	MTUB.TOER.OE3D = 1 MTU9.TMDR.BFB = 0 MTU9.TIORL.IOD[3] = 0 MTU9.TIORL.IOD[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU9.TMDR.MD[3:0] = 1000	MTUB.TOER.OE3D = 1
Complementary PWM mode 1		MTU9.TMDR.MD[3:0] = 1101		
Complementary PWM mode 2		MTU9.TMDR.MD[3:0] = 1110		
Complementary PWM mode 3		MTU9.TMDR.MD[3:0] = 1111		
MTU10		MTIOC10A	Normal operation	MTU10.TMDR.MD[3:0] = 0000
	PWM mode 1		MTU10.TMDR.MD[3:0] = 0010	MTUB.TOER.OE4A = 1 MTU10.TIORH.IOA[1:0] = 01/10/11
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1000	MTUB.TOER.OE4A = 1
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1101	
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1110	
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1111	
	Reset-synchronized PWM mode (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4A = 1
	Complementary PWM mode 1 (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000	
	Complementary PWM mode 2 (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000	
	Complementary PWM mode 3 (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000	

Table 17.11 Settings to Enable Output on the Various MTU Pins (6 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins
MTU10	MTIOC10B	Normal operation	MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4B = 1 MTU10.TIORH.IOB[3] = 0 MTU10.TIORH.IOB[1:0] = 01/10/11
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1000	MTUB.TOER.OE4B = 1
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4B = 1
		Complementary PWM mode 1 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000	
		MTIOC10C	MTIOC10C	Normal operation
PWM mode 1	MTU10.TMDR.MD[3:0] = 0010			MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0 MTU10.TMDR.BFB = 0 MTU10.TIORL.IOC[1:0] = 01/10/11
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1000			MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1101			
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1110			
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1111			
Reset-synchronized PWM mode (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000			MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0
Complementary PWM mode 1 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000			
Complementary PWM mode 2 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000			
Complementary PWM mode 3 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000			
MTIOC10D	MTIOC10D	Normal operation	MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4D = 1 MTU10.TMDR.BFB = 0 MTU10.TIORL.IOD[3] = 0 MTU10.TIORL.IOD[1:0] = 01/10/11
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1000	MTUB.TOER.OE4D = 1
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4D = 1
		Complementary PWM mode 1 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000	

17.1.5 Treatment of Unused Pins

The treatment of unused pins is listed in Table 17.12.

Table 17.12 Treatment of Unused Pins (176-Pin LFBGA)

Pin Name	Treatment
EMLE	Connect this pin to Vss via a pull-down resistor.
MD1, MD0	(Always used as a mode pin)
MDE	(Always used as mode pins)
RES#	Connect this pin to Vcc via a pull-up resistor.
USB0_DP	Leave these pins open.
USB0_DM	
USB1_DP	
USB1_DM	
BSCANP	Connect this pin to Vss via a pull-down resistor.
P35/NMI	Connect this pin to Vcc via a pull-up resistor.
EXTAL	(Always used as a clock pin)
XTAL	Leave this pin open.
XCIN	Connect this pin to Vcc via a pull-up resistor or to Vss via a pull-down resistor.
XCOU	Leave this pin open.
WDTOVF#	Leave this pin open.
Ports 0 to 9, and A to G	<ul style="list-style-type: none"> • Connect these pins to Vcc via a pull-up resistor or to Vss via a pull-down resistor • These pins can be left while PORTn.ICR is in the initial state (the input buffer disabled)*.
VREFH	Connect this pin to AVcc
VREFL	Connect this pin to AVSS
CNVSS	Connect this pin to VSS via pull-down resistor.

Note : * Do not change the initial value of PORTn.ICR. Changing the initial value may generate shoot-through current.

17.2 I/O Ports [for 145-Pin TFLGA/144-Pin LQFP]

The RX62N/RX621 Group (145-pin TFLGA/144-pin LQFP) has 15 I/O ports (ports 0 to 9 and A to E), which handle 105 I/O pins.

17.2.1 Overview

Table 17.13 gives the specifications of the I/O ports and Table 17.14 lists I/O ports and pin functions.

Table 17.13 Specifications of I/O Ports (145-Pin TFLGA/144-Pin LQFP)

Item	Description
I/O pins	103
Input pins	2
Number of ports	15 (0 to 9, A to E)
Built-in input pull-up resistor	Ports 9, A, B, C, D, E
Open drain outputs	Ports 0, 1, 2, 3 (P30 to P34), C
5-V tolerance	Port 0 (P00, P01, P02, P07), port 1 (P12, P13, P16, P17), port 2 (P20, P21), port 3 (P33)
Schmitt trigger input pins	All port inputs, CAN inputs, USB inputs, IRQ inputs, MTU inputs, POE inputs, TMR inputs, RIIC inputs, SCI inputs, and A/D trigger inputs
Others	<ul style="list-style-type: none"> • Each pin is capable of driving a capacitive load of 30 pF in the case of a TTL load. • When configured as an output, a pin is capable of driving a Darlington transistor.

Table 17.14 Port Functions (145-Pin TFLGA/144-Pin LQFP) (1 / 5)

Port	Description	Function			CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability	
		Bit	I/O	Input					Output
Port 0	General I/O port pins, interrupt inputs, TMR inputs, SCI I/O signals, A/D converter inputs, and D/A converter outputs	0	P00	TMRI0-A/IRQ8-A	TxD6-A	—	All input functions	—	√
		1	P01	TMCI0-A/RxD6-A/IRQ9-A			All input functions		
		2	P02/SCK6-A	TMCI1-A/IRQ10-A			All input functions		
		3	P03	IRQ11-A	DA0		All input functions		
		5	P05	IRQ13-A	DA1		All input functions		
		7	P07	ADTRG0#-A/IRQ15-A			All input functions		
Port 1	General I/O port pins, USB I/O signals, MTU I/O signals, TMR I/O signals, interrupt inputs, SCI I/O signals, RIIC I/O signals, PPG I/O signals, and A/D converter inputs	2	P12/SCL0	TMCI1-B/RxD2-A/IRQ2-B		—	All input functions	—	√
		3	P13/SDA0	ADTRG1#/IRQ3-B	TMO3/TxD2-A		All input functions		√
		4	P14	USB0_OVRCURA/TMRI2/IRQ4-B	USB0_DPUPE-B		All input functions		√
		5	P15/MTIOC0B/SCK3-A	TMCI2-A/IRQ5-B	PO13		All input functions		√
		6	P16/MTIOC3C-A	USB0_VBUS/RxD3-A/IRQ6-B	TMO2/PO14/USB0_VBUSEN-B		All input functions		√
		7	P17/MTIOC3A	IRQ7-B	TxD3-A/PO15		All input functions		√

Table 17.14 Port Functions (145-Pin TFLGA/144-Pin LQFP) (2 / 5)

Port	Description	Bit	Function			CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
			I/O	Input	Output				
Port 2	General I/O port pins, bus control I/O signals, EXDMAC I/O signals, USB I/O signals, RSPI I/O signals, MTU I/O signals, PPG outputs, TMR I/O signals, SCI I/O signals, RIIC I/O signals, A/D converter inputs, and on-chip emulator I/O signals	0	P20/MTIOC1A/SDA1	USB0_ID/TMRI0-B	PO0/TxD0	—	All input functions	—	√
		1	P21/MTIOC1B/SCL1	TMCI0-B/RxD0	USB0_EXICEN/PO1	—	All input functions	—	—
		2	P22/MTIOC3B-A/SCK0	EDREQ0-B/MTCLKC-A	USB0_DRPD/PO2/TMO0	EDREQ0-B	P22, MTIOC3B-A/SCK0, MTCLKC-A	—	—
		3	P23/MTIOC3D-A	MTCLKD-A	EDACK0-B/USB0_DPUPE-A/TxD3-B/PO3	—	All input functions	—	—
		4	P24/MTIOC4A-A/SCK3-B	EDREQ1-B/MTCLKA-A/TMRI1	CS4#-C/USB0_VBUSEN-A/PO4	EDREQ1-B	P24, MTIOC4A-A, SCK3-B, MTCLKA-A, TMRI1	—	—
		5	P25/MTIOC4C-A	MTCLKB-A/ADTRG0#-B/RxD3-B	CS5#-C/EDACK1-B/USB0_DPRPD/PO5	—	All input functions	—	—
		6	P26/MOSIB-A/MTIOC2A	—	CS6#-C/PO6/TMO1/TxD1/TDO	MOSIB-A	P26, MTIOC2A	—	—
Port 3	General I/O port pins, CAN I/O signals, RSPI I/O signals, MTU I/O signals, TMR inputs, SCI I/O signals, interrupt inputs, PPG outputs, RTC outputs and on-chip emulator I/O signals	0	P30/MISOB-A/MTIOC4B-A	TMRI3/RxD1/IRQ0/TDI	PO8	MISOB-A	P30, MTIOC4B-A/TMRI3, RxD1, IRQ0, TDI	—	√
		1	P31/SSLB0-A/MTIOC4D-A	TMCI2-B/IRQ1/TMS	PO9	SSLB0-A	P31, MTIOC4D-A/TMCI2-B, IRQ1, TMS	—	√
		2	P32/MTIOC0C	IRQ2-A	CTX0/TxD6-B/PO10/RTCOUT	—	All input functions	—	√
		3	P33/MTIOC0D	CRX0/RxD6-B/IRQ3-A	PO11	—	All input functions	—	√
		4	P34/MTIOC0A/SCK6-B	TMCI3/IRQ4/TRST#	PO12	—	All input functions	—	√
		5	—	P35/NMI	—	—	All input functions	—	—
Port 4	General I/O port pins, interrupt inputs, and A/D converter inputs	0	P40	AN0/IRQ8-B	—	—	P40, IRQ8-B	—	—
		1	P41	AN1/IRQ9-B	—	—	P41, IRQ9-B	—	—
		2	P42	AN2/IRQ10-B	—	—	P42, IRQ10-B	—	—
		3	P43	AN3/IRQ11-B	—	—	P43, IRQ11-B	—	—
		4	P44	AN4/IRQ12	—	—	P44, IRQ12	—	—
		5	P45	AN5/IRQ13-B	—	—	P45, IRQ13-B	—	—
		6	P46	AN6/IRQ14	—	—	P46, IRQ14	—	—
Port 5	General I/O port pins, external bus clock outputs, bus control I/O signals, USB outputs, EXDMAC I/O signals, RSPI outputs, Ether I/O signals, MTU I/O signals, SCI I/O signals, and trace outputs	0	P50	—	WR0#/WR#/SSLB1-A/TxD2-B	—	All input functions	—	—
		1	P51/SCK2	WAIT#-D	WR1#/BC1#/SSLB2-A	WAIT#-D	P51/SCK2	—	—
		2	P52	RxD2-B	RD#/SSLB3-A	—	All input functions	—	—
		3	P53	P53	BCLK	—	All input functions	—	—
		4	P54/MTIOC4B-B	ET_LINKSTA	TRDATA2/EDACK0-C	ET_LI/NKSTA	P54, MTIOC4B-B	—	—
		5	P55/MTIOC4D-B	WAIT#-B/EDREQ0-C	TRDATA3/ET_EXOUT	WAIT#-B, EDREQ0-C	P55, MTIOC4D-B	—	—
6	P56/MTIOC3C-B	—	EDACK1-C	—	All input functions	—	—		

Table 17.14 Port Functions (145-Pin TFLGA/144-Pin LQFP) (3 / 5)

Port	Description	Function				CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
		Bit	I/O	Input	Output				
Port 6	General I/O port pins, SDRAM outputs, and bus control outputs	0	P60		CS0#-A	—	All input functions	—	—
		1	P61		CS1#-A/SDCS#		All input functions		
		2	P62		CS2#-A/RAS#		All input functions		
		3	P63		CS3#-A/CAS#		All input functions		
		4	P64		CS4#-A/WE#		All input functions		
		5	P65		CS5#-A/CKE		All input functions		
		6	P66		CS6#-A/DQM0		All input functions		
		7	P67		CS7#-A/DQM1		All input functions		
Port 7	General I/O port pins, SDRAM outputs, bus control outputs, and Ether I/O signals	0	P70		SDCLK	—	All input functions	—	—
		1	P71/ET_MDIO		CS1#-B	ET_M DIO	P71		
		2	P72		CS2#-B/ET_MDC	—	All input functions		
		3	P73		CS3#-B/ET_WOL		All input functions		
		4	P74	ET_ERXD1/ RMII_RXD1	CS4#-B	ET_ER XD1/ RMII_ RXD1	P74		
		5	P75	ET_ERXD0/ RMII_RXD0	CS5#-B	ET_ER XD0/ RMII_ RXD0	P75		
		6	P76	ET_RX_CLK/ REF50CK	CS6#-B	ET_RX _CLK/ REF50 CK	P76		
		7	P77	ET_RX_ER/ RMII_RX_ER	CS7#-B	ET_RX _ER/ RMII_ RX_ER	P77		
Port 8	General I/O port pins, trace outputs, EXDMAC I/O signals, MTU I/O signals, and Ether I/O signals	0	P80/MTIOC3B-B	EDREQ0-A	TRDATA0/ ET_TX_EN/ RMII_TXD_EN	EDRE Q0-A	P80, MTIOC3B-B	—	—
		1	P81/MTIOC3D-B		TRDATA1/ EDACK0-A/ ET_ETXD0/ RMII_TXD0	—	All input functions		
		2	P82/MTIOC4A-B	EDREQ1-A	TRSYNC/ ET_ETXD1 RMII_TXD1	EDRE Q1-A	P82, MTIOC4A-B		
		3	P83/MTIOC4C-B	ET_CRCS/ RMII_CRCS_DV	TRCLK/ EDACK1-A	ET_CR S/ RMII_ CRCS_ DV	P83, MTIOC4C-B		
Port 9	General I/O port pins, and address outputs	0	P90		A16-B	—	All input functions	√	—
		1	P91		A17-B	—	All input functions	√	
		2	P92		A18-B	—	All input functions	√	
		3	P93		A19-B	—	All input functions	√	

Table 17.14 Port Functions (145-Pin TFLGA/144-Pin LQFP) (4 / 5)

Port	Description	Bit	Function			CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
			I/O	Input	Output				
Port A	General I/O port pins, address outputs, bus control I/O signals, RSPI I/O signals, MTU I/O signals, and PPG outputs	0	PA0/MTIOC6A		A0/BC0# / SSLA1-B/PO16	—	All input functions	√	—
		1	PA1/MTIOC6B		A1/ SSLA2-B/PO17		All input functions		
		2	PA2/MTIOC6C		A2/SSLA3-B/ PO18		All input functions		
		3	PA3/MTIOC6D		A3/PO19		All input functions		
		4	PA4/SSLA0-B/ MTIOC7A		A4/PO20	SSLA0 -B	PA4, MTIOC7A		
		5	PA5/ RSPCKA-B/ MTIOC7B		A5/PO21	RSPC KA-B	PA5, MTIOC7B		
		6	PA6/MOSIA-B/ MTIOC8A		A6/PO22	MOSIA -B	PA6, MTIOC8A		
		7	PA7/MISOA-B/ MTIOC8B		A7/PO23	MISOA -B	PA7, MTIOC8B		
Port B	General I/O port pins, address outputs, MTU I/O signals, and PPG outputs	0	PB0/MTIOC9A		A8/PO24	—	All input functions	√	—
		1	PB1/MTIOC9C		A9/PO25		All input functions		
		2	PB2/MTIOC9B	MTCLKG-B	A10/PO26		All input functions		
		3	PB3/MTIOC9D	MTCLKH-B	A11/PO27		All input functions		
		4	PB4/ MTIOC10A	MTCLKE-B	A12/PO28		All input functions		
		5	PB5/ MTIOC10C	MTCLKF-B	A13/PO29		All input functions		
		6	PB6/ MTIOC10B		A14/PO30		All input functions		
		7	PB7/ MTIOC10D		A15/PO31		All input functions		
Port C	General I/O port pins, address outputs, bus control outputs, RSPI I/O signals, Ether I/O signals, MTU inputs, and SCI I/O signals	0	PC0	ET_ERXD3/ MTCLKG-A	A16-A/SSLA1-A	ET_ER XD3	PC0, MTCLKG-A	√	√
		1	PC1/SCK5	ET_ERXD2/ MTCLKH-A	A17-A/SSLA2-A	ET_ER XD2	PC1, SCK5, MTCLKH-A		
		2	PC2	ET_RX_DV/ MTCLKE-A/RxD5	A18-A/SSLA3-A	ET_RX _DV	PC2, MTCLKE-A, RxD5		
		3	PC3	ET_TX_ER/ MTCLKF-A	A19-A/TxD5	ET_TX _ER	PC3, MTCLKF-A		
		4	PC4/SSLA0-A	MTCLKC-B/ ET_TX_CLK	A20-A/CS3#-C	SSLA0 -A, ET_TX _CLK	PC4, MTCLKC-B		
		5	PC5/ RSPCKA-A	WAIT#-C/ MTIC11W-A/ MTCLKD-B	A21-A/CS2#-C/ ET_ETXD2	RSPC KA-A, WAIT# -C	PC5, MTIC11W-A, MTCLKD-B		
		6	PC6/MOSIA-A	MTIC11V-A/ MTCLKA-B	A22-A /CS1#-C/ ET_ETXD3	MOSIA -A	PC6, MTIC11V-A MTCLKA-B		
		7	PC7/MISOA-A	ET_COL/ MTIC11U-A/ MTCLKB-B	A23-A/CS0#-B	MISOA -A, ET_CO L	PC7, MTIC11U-A MTCLKB-B		

Table 17.14 Port Functions (145-Pin TFLGA/144-Pin LQFP) (5 / 5)

Port	Description	Function				CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
		Bit	I/O	Input	Output				
Port D	General I/O port pins, bidirectional data-bus lines, MTU inputs, and POE inputs	0	PD0/D0	POE7#		D0	PD0, POE7#	√	—
		1	PD1/D1	POE6#		D1	PD1, POE6#		
		2	PD2/D2	MTIC11W-B/ POE5#		D2	PD2, MTIC11W-B, POE5#		
		3	PD3/D3	MTIC11V-B/ POE4#		D3	PD3, MTIC11V-B, POE4#		
		4	PD4/D4	MTIC11U-B/ POE3#		D4	PD4, MTIC11U-B, POE3#		
		5	PD5/D5	MTIC5W/ POE2#		D5	PD5, MTIC5W, POE2#		
		6	PD6/D6	MTIC5V/ POE1#		D6	PD6, MTIC5V, POE1#		
		7	PD7/D7	MTIC5U/ POE0#		D7	PD7, MTIC5U, POE0#		
Port E	General I/O port pins, bidirectional data-bus lines, RSPI I/O signals, interrupt inputs, and POE inputs	0	PE0/D8		SSLB1-B	D8	PE0	√	—
		1	PE1/D9		SSLB2-B	D9	PE1		
		2	PE2/D10	POE9#	SSLB3-B	D10	PE2, POE9#		
		3	PE3/D11	POE8#		D11	PE3, POE8#		
		4	PE4/D12/ SSLB0-B			D12, SSLB0-B	PE4		
		5	PE5/D13/ RSPCKB-B	IRQ5-A		D13, RSPC KB-B	PE5, IRQ5-A		
		6	PE6/D14/ MOSIB-B	IRQ6-A		D14, MOSIB-B	PE6, IRQ6-A		
		7	PE7/D15/ MISOB-B	IRQ7-A		D15, MISOB-B	PE7, IRQ7-A		

17.2.2 Register Descriptions

Table 17.15 lists registers of I/O ports, and Table 17.16 lists valid bits in each register.

Table 17.15 Registers of I/O Ports (145-Pin TFLGA/144-Pin LQFP) (1 / 3)

Port Symbol	Register Name	Register Symbol	Value after Reset	Address	Access Size
PORT0	Data direction register	DDR	00h	0008 C000h	8
	Data register	DR	00h	0008 C020h	8
	Port register	PORT	Undefined	0008 C040h	8
	Input buffer control register	ICR	00h	0008 C060h	8
	Open drain control register	ODR	00h	0008 C080h	8
PORT1	Data direction register	DDR	00h	0008 C001h	8
	Data register	DR	00h	0008 C021h	8
	Port register	PORT	Undefined	0008 C041h	8
	Input buffer control register	ICR	00h	0008 C061h	8
	Open drain control register	ODR	00h	0008 C081h	8
PORT2	Data direction register	DDR	00h	0008 C002h	8
	Data register	DR	00h	0008 C022h	8
	Port register	PORT	Undefined	0008 C042h	8
	Input buffer control register	ICR	00h	0008 C062h	8
	Open drain control register	ODR	00h	0008 C082h	8
PORT3	Data direction register	DDR	00h	0008 C003h	8
	Data register	DR	00h	0008 C023h	8
	Port register	PORT	Undefined	0008 C043h	8
	Input buffer control register	ICR	00h	0008 C063h	8
	Open drain control register	ODR	00h	0008 C083h	8
PORT4	Data direction register	DDR	00h	0008 C004h	8
	Data register	DR	00h	0008 C024h	8
	Port register	PORT	Undefined	0008 C044h	8
	Input buffer control register	ICR	00h	0008 C064h	8
PORT5	Data direction register	DDR	00h	0008 C005h	8
	Data register	DR	00h	0008 C025h	8
	Port register	PORT	Undefined	0008 C045h	8
	Input buffer control register	ICR	00h	0008 C065h	8
PORT6	Data direction register	DDR	00h	0008 C006h	8
	Data register	DR	00h	0008 C026h	8
	Port register	PORT	Undefined	0008 C046h	8
	Input buffer control register	ICR	00h	0008 C066h	8
PORT7	Data direction register	DDR	00h	0008 C007h	8
	Data register	DR	00h	0008 C027h	8
	Port register	PORT	Undefined	0008 C047h	8
	Input buffer control register	ICR	00h	0008 C067h	8
PORT8	Data direction register	DDR	00h	0008 C008h	8
	Data register	DR	00h	0008 C028h	8
	Port register	PORT	Undefined	0008 C048h	8
	Input buffer control register	ICR	00h	0008 C068h	8

Table 17.15 Registers of I/O Ports (145-Pin TFLGA/144-Pin LQFP) (2 / 3)

Port Symbol	Register Name	Register Symbol	Value after Reset	Address	Access Size
PORT9	Data direction register	DDR	00h	0008 C009h	8
	Data register	DR	00h	0008 C029h	8
	Port register	PORT	Undefined	0008 C049h	8
	Input buffer control register	ICR	00h	0008 C069h	8
	Pull-up resistor control register	PCR	00h	0008 C0C9h	8
PORTA	Data direction register	DDR	00h	0008 C00Ah	8
	Data register	DR	00h	0008 C02Ah	8
	Port register	PORT	Undefined	0008 C04Ah	8
	Input buffer control register	ICR	00h	0008 C06Ah	8
	Pull-up resistor control register	PCR	00h	0008 C0CAh	8
PORTB	Data direction register	DDR	00h	0008 C00Bh	8
	Data register	DR	00h	0008 C02Bh	8
	Port register	PORT	Undefined	0008 C04Bh	8
	Input buffer control register	ICR	00h	0008 C06Bh	8
	Pull-up resistor control register	PCR	00h	0008 C0CBh	8
PORTC	Data direction register	DDR	00h	0008 C00Ch	8
	Data register	DR	00h	0008 C02Ch	8
	Port register	PORT	Undefined	0008 C04Ch	8
	Input buffer control register	ICR	00h	0008 C06Ch	8
	Open drain control register	ODR	00h	0008 C08Ch	8
	Pull-up resistor control register	PCR	00h	0008 C0CCh	8
PORTD	Data direction register	DDR	00h	0008 C00Dh	8
	Data register	DR	00h	0008 C02Dh	8
	Port register	PORT	Undefined	0008 C04Dh	8
	Input buffer control register	ICR	00h	0008 C06Dh	8
	Pull-up resistor control register	PCR	00h	0008 C0CDh	8
PORTE	Data direction register	DDR	00h	0008 C00Eh	8
	Data register	DR	00h	0008 C02Eh	8
	Port register	PORT	Undefined	0008 C04Eh	8
	Input buffer control register	ICR	00h	0008 C06Eh	8
	Pull-up resistor control register	PCR	00h	0008 C0CEh	8

Table 17.15 Registers of I/O Ports (145-Pin TFLGA/144-Pin LQFP) (3 / 3)

Port Symbol	Register Name	Register Symbol	Value after Reset	Address	Access Size
IOPORT	Port function control register 0	PF0CSE	00h	0008 C100h	8
	Port function control register 1	PF1CSS	00h	0008 C101h	8
	Port function control register 2	PF2CSS	00h	0008 C102h	8
	Port function control register 3	PF3BUS	00h	0008 C103h	8
	Port function control register 4	PF4BUS	00h	0008 C104h	8
	Port function control register 5	PF5BUS	00h	0008 C105h	8
	Port function control register 6	PF6BUS	00h	0008 C106h	8
	Port function control register 7	PF7DMA	00h	0008 C107h	8
	Port function control register 8	PF8IRQ	00h	0008 C108h	8
	Port function control register 9	PF9IRQ	00h	0008 C109h	8
	Port function control register A	PFAADC	00h	0008 C10Ah	8
	Port function control register B	PFBTMR	00h	0008 C10Bh	8
	Port function control register C	PFCMTU	00h	0008 C10Ch	8
	Port function control register D	PFDMTU	00h	0008 C10Dh	8
	Port function control register E	PFENET	00h	0008 C10Eh	8
	Port function control register F	PFSCI	00h	0008 C10Fh	8
Port function control register G	PFGSPI	00h	0008 C110h	8	
Port function control register H	PFHSPI	00h	0008 C111h	8	
Port function control register J	PFJCAN	00h	0008 C113h	8	
Port function control register K	PFKUSB	00h	0008 C114h	8	
Port function control register M	PFMPOE	00h	0008 C116h	8	
Port function control register N	PFNPOE	00h	0008 C117h	8	

Table 17.16 Valid Bits in Each Register (145-Pin TFLGA/144-Pin LQFP) (1 / 3)

Register Symbol	b7	b6	b5	b4	b3	b2	b1	b0
PORT0.DDR	√	x	√	x	√	√	√	√
PORT1.DDR	√	√	√	√	√	√	x	x
PORT2.DDR	√	√	√	√	√	√	√	√
PORT3.DDR	x	x	x	√	√	√	√	√
PORT4.DDR	√	√	√	√	√	√	√	√
PORT5.DDR	x	√	√	√	√	√	√	√
PORT6.DDR	√	√	√	√	√	√	√	√
PORT7.DDR	√	√	√	√	√	√	√	√
PORT8.DDR	x	x	x	x	√	√	√	√
PORT9.DDR	x	x	x	x	√	√	√	√
PORTA.DDR	√	√	√	√	√	√	√	√
PORTB.DDR	√	√	√	√	√	√	√	√
PORTC.DDR	√	√	√	√	√	√	√	√
PORTD.DDR	√	√	√	√	√	√	√	√
PORTE.DDR	√	√	√	√	√	√	√	√
PORT0.DR	√	x	√	x	√	√	√	√
PORT1.DR	√	√	√	√	√	√	x	x
PORT2.DR	√	√	√	√	√	√	√	√
PORT3.DR	x	x	x	√	√	√	√	√
PORT4.DR	√	√	√	√	√	√	√	√
PORT5.DR	x	√	√	√	x	√	√	√
PORT6.DR	√	√	√	√	√	√	√	√
PORT7.DR	√	√	√	√	√	√	√	√
PORT8.DR	x	x	x	x	√	√	√	√
PORT9.DR	x	x	x	x	√	√	√	√
PORTA.DR	√	√	√	√	√	√	√	√
PORTB.DR	√	√	√	√	√	√	√	√
PORTC.DR	√	√	√	√	√	√	√	√
PORTD.DR	√	√	√	√	√	√	√	√
PORTE.DR	√	√	√	√	√	√	√	√
PORTORT0.PORT	√	x	√	x	√	√	√	√
PORT1.PORT	√	√	√	√	√	√	x	x
PORT2.PORT	√	√	√	√	√	√	√	√
PORT3.PORT	x	x	√	√	√	√	√	√
PORT4.PORT	√	√	√	√	√	√	√	√
PORT5.PORT	x	√	√	√	√	√	√	√
PORT6.PORT	√	√	√	√	√	√	√	√
PORT7.PORT	√	√	√	√	√	√	√	√
PORT8.PORT	x	x	x	x	√	√	√	√
PORT9.PORT	x	x	x	x	√	√	√	√
PORTA.PORT	√	√	√	√	√	√	√	√
PORTB.PORT	√	√	√	√	√	√	√	√
PORTC.PORT	√	√	√	√	√	√	√	√
PORTD.PORT	√	√	√	√	√	√	√	√

Table 17.16 Valid Bits in Each Register (145-Pin TFLGA/144-Pin LQFP) (2 / 3)

Register Symbol	b7	b6	b5	b4	b3	b2	b1	b0
PORTE.PORT	√	√	√	√	√	√	√	√
PORT0.ICR	√	x	√	x	√	√	√	√
PORT1.ICR	√	√	√	√	√	√	x	x
PORT2.ICR	√	√	√	√	√	√	√	√
PORT3.ICR	x	x	x	√	√	√	√	√
PORT4.ICR	√	√	√	√	√	√	√	√
PORT5.ICR	x	√	√	√	√	√	√	√
PORT6.ICR	√	√	√	√	√	√	√	√
PORT7.ICR	√	√	√	√	√	√	√	√
PORT8.ICR	x	x	x	x	√	√	√	√
PORT9.ICR	x	x	x	x	√	√	√	√
PORTA.ICR	√	√	√	√	√	√	√	√
PORTB.ICR	√	√	√	√	√	√	√	√
PORTC.ICR	√	√	√	√	√	√	√	√
PORTD.ICR	√	√	√	√	√	√	√	√
PORTE.ICR	√	√	√	√	√	√	√	√
PORT0.ODR	√	x	√	x	√	√	√	√
PORT1.ODR	√	√	√	√	√	√	x	x
PORT2.ODR	√	√	√	√	√	√	√	√
PORT3.ODR	x	x	x	√	√	√	√	√
PORTC.ODR	√	√	√	√	√	√	√	√
PORT9.PCR	x	x	x	x	√	√	√	√
PORTA.PCR	√	√	√	√	√	√	√	√
PORTB.PCR	√	√	√	√	√	√	√	√
PORTC.PCR	√	√	√	√	√	√	√	√
PORTD.PCR	√	√	√	√	√	√	√	√
PORTE.PCR	√	√	√	√	√	√	√	√
IOPORT.PF0CSE	√	√	√	√	√	√	√	√
IOPORT.PF1CSS	√	√	√	√	√	√	√	√
IOPORT.PF2CSS	√	√	√	√	√	√	x	√
IOPORT.PF3BUS	√	√	√	√	√	√	√	√
IOPORT.PF4BUS	√	√	√	√	√	√	√	√
IOPORT.PF5BUS	x	√	x	√	x	x	√	x
IOPORT.PF6BUS	√	√	x	√	x	x	√	√
IOPORT.PF7DMA	√	√	√	√	x	x	x	x
IOPORT.PF8IRQ	√	x	√	x	√	√	√	√
IOPORT.PF9IRQ	√	√	√	√	√	√	x	x
IOPORT.PFAADC	x	x	x	x	x	x	x	√
IOPORT.PFBTMR	x	x	x	x	x	√	√	√
IOPORT.PFCMTU	√	x	√	√	√	√	x	x
IOPORT.PFDMTU	√	√	x	x	x	x	x	x
IOPORT.PFENET	√	x	x	√	√	√	√	√
IOPORT.PFFSCI	x	√	x	x	√	√	x	x
IOPORT.PFGSPI	√	√	√	√	√	√	√	√
IOPORT.PFHSPI	√	√	√	√	√	√	√	√

Table 17.16 Valid Bits in Each Register (145-Pin TFLGA/144-Pin LQFP) (3 / 3)

Register Symbol	b7	b6	b5	b4	b3	b2	b1	b0
IOPORT.PFJCAN	x	x	x	x	x	x	x	√
IOPORT.PFKUSB	x	x	x	√	√	√	√	√
IOPORT.PFMPOE	√	√	√	√	√	√	√	√
IOPORT.PFNPOE	x	x	x	x	x	x	√	√

17.2.2.1 Data Direction Register (DDR)

Addresses: PORT0.DDR 0008 C000h, PORT1.DDR 0008 C001h, PORT2.DDR 0008 C002h,
 PORT3.DDR 0008 C003h, PORT4.DDR 0008 C004h, PORT5.DDR 0008 C005h,
 PORT6.DDR 0008 C006h, PORT7.DDR 0008 C007h, PORT8.DDR 0008 C008h,
 PORT9.DDR 0008 C009h, PORTA.DDR 0008 C00Ah, PORTB.DDR 0008 C00Bh,
 PORTC.DDR 0008 C00Ch, PORTD.DDR 0008 C00Dh, PORTE.DDR 0008 C00Eh

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

Note: Bits 6 and 4 are reserved in PORT0.DDR.
 Bits 1 and 0 are reserved in PORT1.DDR.
 The lower five bits are valid and the upper three bits are reserved in PORT3.DDR.
 The lower four bits are valid and the upper four are reserved in PORT8.DDR.
 The lower four bits are valid and the upper four are reserved in PORT9.DDR.
 The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 I/O Select	0: An input pin	R/W
b1	B1	Pn1 I/O Select	1: An output pin	R/W
b2	B2	Pn2 I/O Select		R/W
b3	B3	Pn3 I/O Select		R/W
b4	B4	Pn4 I/O Select		R/W
b5	B5	Pn5 I/O Select		R/W
b6	B6	Pn6 I/O Select		R/W
b7	B7	Pn7 I/O Select		R/W

[Legend] n = 0 to 9, A to E

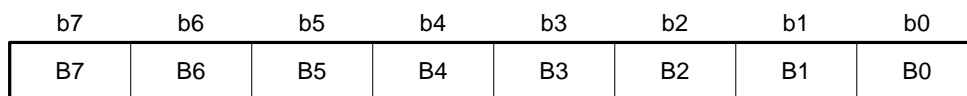
Each DDR is used to select the input or output direction for individual pins of the corresponding port that have been configured to function as general I/O pins.

Each bit of a PORTn.DDR (n = 0 to 9, A to E) corresponds to a pin of PORTn, and the settings can change from bit to bit.

The PORT5.DDR.B3 bit selects P53 input or BCLK output. Setting the PORT5.DDR.B3 bit to 1 specifies output of the BCLK signal on the pin that would otherwise be P53. Operation as a general output is not selectable for this pin.

17.2.2.2 Data Register (DR)

Addresses: PORT0.DR 0008 C020h, PORT1.DR 0008 C021h, PORT2.DR 0008 C022h,
 PORT3.DR 0008 C023h, PORT4.DR 0008 C024h, PORT5.DR 0008 C025h,
 PORT6.DR 0008 C026h, PORT7.DR 0008 C027h, PORT8.DR 0008 C028h,
 PORT9.DR 0008 C029h, PORTA.DR 0008 C02Ah, PORTB.DR 0008 C02Bh,
 PORTC.DR 0008 C02Ch, PORTD.DR 0008 C02Dh, PORTE.DR 0008 C02Eh



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

Note: Bits 6 and 4 are reserved in PORT0.DR.
 Bits 1 and 0 are reserved in PORT1.DR.
 The lower five bits are valid and the upper three bits are reserved in PORT3.DR.
 Bits 7 and 3 are reserved in PORT5.DR.
 The lower four bits are valid and the upper four bits are reserved in PORT8.DR.
 The lower four bits are valid and the upper four bits are reserved in PORT9.DR.
 The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 Output Data Store	Output data are stored.	R/W
b1	B1	Pn1 Output Data Store		R/W
b2	B2	Pn2 Output Data Store		R/W
b3	B3	Pn3 Output Data Store		R/W
b4	B4	Pn4 Output Data Store		R/W
b5	B5	Pn5 Output Data Store		R/W
b6	B6	Pn6 Output Data Store		R/W
b7	B7	Pn7 Output Data Store		R/W

[Legend] n = 0 to 9, A to E

Each DR stores the output data from the individual pins of the corresponding port used as a general I/O port.
 The output of the P53 pin is the BCLK signal and the value of the B3 bit in PORT5.DR does not affect the pin.

17.2.2.3 Port Register (PORT)

Addresses: PORT0.PORT 0008 C040h, PORT1.PORT 0008 C041h, PORT2.PORT 0008 C042h, PORT3.PORT 0009 C043h, PORT4.PORT 0008 C044h, PORT5.PORT 0008 C045h, PORT6.PORT 0008 C046h, PORT7.PORT 0008 C047h, PORT8.PORT 0008 C048h, PORT9.PORT 0008 C049h, PORTA.PORT 0008 C04Ah, PORTB.PORT 0008 C04Bh, PORTC.PORT 0008 C04Ch, PORTD.PORT 0008 C04Dh, PORTE.PORT 0008 C04Eh

b7	b6	b5	b4	b3	b2	b1	b0
B7	B6	B5	B4	B3	B2	B1	B0

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

Note: Bits 6 and 4 are reserved in PORT0.PORT.
 Bits 1 and 0 are reserved in PORT1.PORT.
 The lower six bits are valid and the upper two bits are reserved in PORT3.PORT.
 The lower seven bits are valid and the upper one bit is reserved in PORT5.PORT.
 The lower four bits are valid and the upper four bits are reserved in PORT8.PORT.
 The lower four bits are valid and the upper four bits are reserved in PORT9.PORT.
 The reserved bits are read as 1 and cannot be modified.

Bit	Symbol	Bit Name	Description	R/W
b0	B0*	Pn0	Individual pin states of the corresponding port are reflected.	R
b1	B1*	Pn1		R
b2	B2*	Pn2		R
b3	B3*	Pn3		R
b4	B4*	Pn4		R
b5	B5*	Pn5		R
b6	B6*	Pn6		R
b7	B7*	Pn7		R

[Legend] n = 0 to 9, A to E

Note : * Before reading this register, set the corresponding bit in PORTn.ICR to 1. If this register is read with the corresponding bit in PORTn.ICR set to 0, the read value is undefined.

PORT reflects individual pin states of the corresponding port.

When a PORTn.PORT (n = 0 to 9, A to E) is read, the corresponding pin states are read out to here.

The NMI pin state is read out to the P35 bit.

17.2.2.4 Input Buffer Control Register (ICR)

Addresses: PORT0.ICR 0008 C060h, PORT1.ICR 0008 C061h, PORT2.ICR 0008 C062h,
PORT3.ICR 0008 C063h, PORT4.ICR 0008 C064h, PORT5.ICR 0008 C065h,
PORT6.ICR 0008 C066h, PORT7.ICR 0008 C067h, PORT8.ICR 0008 C068h,
PORT9.ICR 0008 C069h, PORTA.ICR 0008 C06Ah, PORTB.ICR 0008 C06Bh,
PORTC.ICR 0008 C06Ch, PORTD.ICR 0008 C06Dh, PORTE.ICR 0008 C06Eh

b7	b6	b5	b4	b3	b2	b1	b0
B7	B6	B5	B4	B3	B2	B1	B0

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

Note: Bits 6 and 4 are reserved in PORT0.ICR.
Bits 1 and 0 are reserved in PORT1.ICR.
The lower five bits are valid and the upper three bits are reserved in PORT3.ICR.
The lower seven bits are valid and the upper one bit is reserved in PORT5.ICR.
The lower four bits are valid and the upper four bits are reserved in PORT8.ICR.
The lower four bits are valid and the upper four bits are reserved in PORT9.ICR.
The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0*	Pn0 Input Buffer Control	0: The input buffer for the corresponding pin is disabled. 1: The input buffer for the corresponding pin is enabled.	R/W
b1	B1*	Pn1 Input Buffer Control		R/W
b2	B2*	Pn2 Input Buffer Control		R/W
b3	B3*	Pn3 Input Buffer Control		R/W
b4	B4*	Pn4 Input Buffer Control		R/W
b5	B5*	Pn5 Input Buffer Control		R/W
b6	B6*	Pn6 Input Buffer Control		R/W
b7	B7*	Pn7 Input Buffer Control		R/W

[Legend] n = 0 to 9, A to E

Note : * For pins being used as input pins, set the corresponding bits to 1. Set the bits corresponding to pins that are not being used for their input functions or are being used as analog I/O pins to 0.

Each ICR controls the input buffers for the individual pins of the corresponding port.

Each bit of a PORTn.ICR (n = 0 to 9, A to E) corresponds to a pin of PORTn, and the settings can change from bit to bit.

When to be used as an input pin for the peripheral module, the input buffer for the corresponding pin should be enabled beforehand by setting the PORTn.ICR bit to 1. If this register is used as an input pin for the peripheral module while the PORTn.ICR bit is 0, the input signal to the peripheral module is fixed high.

Changes in the settings of a PORTn.ICR may generate edges internally, depending on the pin state. For this reason, change the settings of PORTn.ICR while the corresponding input pins are not in use. For example, in the case of IRQ_i (i = 0 to 15) inputs, change settings of the corresponding PORTn.ICR with interrupts disabled by clearing the IR flag in IRI (i = 64 to 79 ("i" shows an interrupt vector number)) of the interrupt control unit (ICU) to 0, and then enable the corresponding interrupts. If a change to a PORTn.ICR setting does generate an edge, negate the edge.

17.2.2.5 Open Drain Control Register (ODR)

Addresses: PORT0.ODR 0008 C080h, PORT1.ODR 0008 C081h, PORT2.ODR 0008 C082h,
PORT3.ODR 0008 C083h, PORTC.ODR 0008 C08Ch

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

Note: Bits 6 and 4 are reserved in PORT0.ODR.
Bits 1 and 0 are reserved in PORT1.ODR.
The lower five bits are valid and the upper three bits are reserved in PORT3.ODR.
The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 Output Type Select	0: CMOS output pin	R/W
b1	B1	Pn1 Output Type Select	1: NMOS open-drain output pin	R/W
b2	B2	Pn2 Output Type Select		R/W
b3	B3	Pn3 Output Type Select		R/W
b4	B4	Pn4 Output Type Select		R/W
b5	B5	Pn5 Output Type Select		R/W
b6	B6	Pn6 Output Type Select		R/W
b7	B7	Pn7 Output Type Select		R/W

[Legend] n = 0 to 3, C

Each ODR is used to select an output type for the individual pins.

17.2.2.6 Pull-Up Resistor Control Register (PCR)

Addresses: PORT9.PCR 0008 C0C9h, PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTC.PCR 0008 C0CCh, PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh

	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

Note: The lower four bits are valid and the upper four bits are reserved in PORT9.PORT. The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 Input Pull-Up Resistor Control	0: Input pull-up resistor is off.	R/W
b1	B1	Pn1 Input Pull-Up Resistor Control	1: Input pull-up resistor is on.	R/W
b2	B2	Pn2 Input Pull-Up Resistor Control		R/W
b3	B3	Pn3 Input Pull-Up Resistor Control		R/W
b4	B4	Pn4 Input Pull-Up Resistor Control		R/W
b5	B5	Pn5 Input Pull-Up Resistor Control		R/W
b6	B6	Pn6 Input Pull-Up Resistor Control		R/W
b7	B7	Pn7 Input Pull-Up Resistor Control		R/W

[Legend] n = 0 to 9, A to E

Each PCR controls enabled/disabled of input pull-up resistor for individual pins of the corresponding port.

When in input pin state, for the pins corresponding to bits where the value in PORTn.PCR is 1, input pull-up resistor is turned on. Table 17.17 summarizes the input pull-up resistor states.

Table 17.17 Input Pull-Up Resistor States (145-Pin TFLGA/144-Pin LQFP)

Port	Pin State	Reset or Hardware Standby Mode	In Other Operations
Port 9	Data I/O	Disabled	
	Address output	Disabled	
	Port output	Disabled	Enabled/Disabled
Port A	Port input	Disabled	
	Address output	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	Enabled/Disabled
Port B	Address output	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port C	Address output	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port D	Data I/O	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port E	Data I/O	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled

[Legend]

Disabled: Input pull-up resistor is always disabled.

Enabled/Disabled: Input pull-up resistor is enabled when the PORTm.PCR.Bj bit (m = 9, and A to E, j = 0 to 7) is set to 1, and disabled when the bit is cleared to 0.

17.2.2.7 Port Function Control Register 0 (PF0CSE)

Address: 0008 C100h

b7	b6	b5	b4	b3	b2	b1	b0
CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E

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

Bit	Symbol	Bit Name	Description	R/W
b0	CS0E	CS0 Output Enable	0: Designated as an I/O port pin.	R/W
b1	CS1E	CS1 Output Enable	1: Designated as the CSn# output pin (n = 0 to 7)	R/W
b2	CS2E	CS2 Output Enable		R/W
b3	CS3E	CS3 Output Enable		R/W
b4	CS4E	CS4 Output Enable		R/W
b5	CS5E	CS5 Output Enable		R/W
b6	CS6E	CS6 Output Enable		R/W
b7	CS7E	CS7 Output Enable		R/W

PF0CSE enables or disables CSn# output.

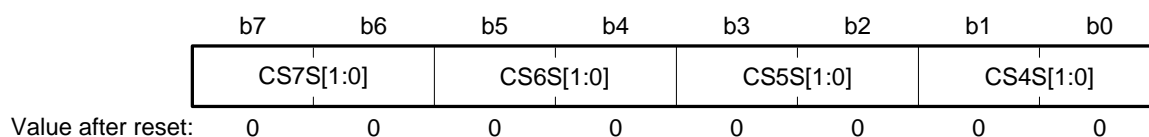
CSnE Bit (CSn Output Enable) (n = 0 to 7)

Each bit enables or disables the corresponding CSn# output.

To output a CSn signal, set the corresponding CSnE bit in PF0CSE to 1.

17.2.2.8 Port Function Control Register 1 (PF1CSS)

Address: 0008 C101h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	CS4S[1: 0]	CS4# Output Pin Select	b1 b0 0 0: P64 is designated as the CS4#-A output pin. 0 1: P74 is designated as the CS4#-B output pin. 1 x: P24 is designated as the CS4#-C output pin.	R/W
b3, b2	CS5S[1: 0]	CS5# Output Pin Select	b3 b2 0 0: P65 is designated as the CS5#-A output pin. 0 1: P75 is designated as the CS5#-B output pin. 1 x: P25 is designated as the CS5#-C output pin.	R/W
b5, b4	CS6S[1: 0]	CS6# Output Pin Select	b5 b4 0 0: P66 is designated as the CS6#-A output pin. 0 1: P76 is designated as the CS6#-B output pin. 1 x: P26 is designated as the CS6#-C output pin.	R/W
b7, b6	CS7S[1: 0]	CS7# Output Pin Select	b7 b6 0 0: P67 is designated as the CS7#-A output pin. 0 1: P77 is designated as the CS7#-B output pin. 1 x: P27 is designated as the CS7#-C output pin.	R/W

[Legend]

x: Don't care

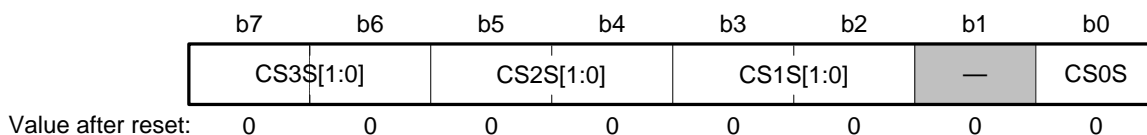
PF1CSS is used to select a pin for each CSn# output (n = 4 to 7).

CSnS[1:0] Bits (CSn# Output Pin Select) (n = 4 to 7)

These bits select a pin for each CSn# output when a CSn# output is enabled (the corresponding CSnE bit in PFOCSE = 1).

17.2.2.9 Port Function Control Register 2 (PF2CSS)

Address: 0008 C102h



Bit	Symbol	Bit Name	Description	R/W
b0	CS0S	CS0# Output Pin Select	0: P60 is designated as the CS0#-A output pin. 1: PC7 is designated as the CS0#-B output pin.	R/W
b1	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b3, b2	CS1S[1:0]	CS1# Output Pin Select	b3 b2 0 0: P61 is designated as the CS1#-A output pin. 0 1: P71 is designated as the CS1#-B output pin. 1 x: PC6 is designated as the CS1#-C output pin.	R/W
b5, b4	CS2S[1:0]	CS2# Output Pin Select	b5 b4 0 0: P62 is designated as the CS2#-A output pin. 0 1: P72 is designated as the CS2#-B output pin. 1 x: PC5 is designated as the CS2#-C output pin.	R/W
b7, b6	CS3S[1:0]	CS3# Output Pin Select	b7 b6 0 0: P63 is designated as the CS3#-A output pin. 0 1: P73 is designated as the CS3#-B output pin. 1 x: PC4 is designated as the CS3#-C output pin.	R/W

[Legend]

x: Don't care

PF2CSS is used to select a pin for each CSn# output (n = 0 to 3).

CSnS[1:0] Bits (CSn# Output Pin Select) (n = 0 to 3)

These bits select a pin for each CSn# output when a CSn# output is enabled (the corresponding CSnE bit in PFOCSE = 1).

17.2.2.10 Port Function Control Register 3 (PF3BUS)

Address: 0008 C103h

	b7	b6	b5	b4	b3	b2	b1	b0
	A23E	A22E	A21E	A20E	A19E	A18E	A17E	A16E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	A16E	Address A16 Output Enable	0: A16 output is disabled. 1: A16 output is enabled.	R/W
b1	A17E	Address A17 Output Enable	0: A17 output is disabled. 1: A17 output is enabled.	R/W
b2	A18E	Address A18 Output Enable	0: A18 output is disabled. 1: A18 output is enabled.	R/W
b3	A19E	Address A19 Output Enable	0: A19 output is disabled. 1: A19 output is enabled.	R/W
b4	A20E	Address A20 Output Enable	0: A20 output is disabled. 1: A20 output is enabled.	R/W
b5	A21E	Address A21 Output Enable	0: A21 output is disabled. 1: A21 output is enabled.	R/W
b6	A22E	Address A22 Output Enable	0: A22 output is disabled. 1: A22 output is enabled.	R/W
b7	A23E	Address A23 Output Enable	0: A23 output is disabled. 1: A23 output is enabled.	R/W

PF3BUS enables or disables address outputs.

AnE Bit (Address An Output Enable) (n = 16 to 23)

Each bit enables or disables an address output (An).

17.2.2.11 Port Function Control Register 4 (PF4BUS)

Address: 0008 C104h

b7	b6	b5	b4	b3	b2	b1	b0
A15E	A14E	A13E	A12E	A11E	A10E	ADRLE[1:0]	
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ADRLE[1: 0]	Address Lower A9 to A0 Output Enable	b1 b0 0 0: A9 to A0 output is disabled. 0 1: A9 to A4 output is disabled, A3 to A0 output is enabled 1 0: A9 to A8 output is disabled, A7 to A0 output is enabled 1 1: A9 to A0 output is enabled	R/W
b2	A10E	Address A10 Output Enable	0: A10 output is disabled. 1: A10 output is enabled.	R/W
b3	A11E	Address A11 Output Enable	0: A11 output is disabled. 1: A11 output is enabled.	R/W
b4	A12E	Address A12 Output Enable	0: A12 output is disabled. 1: A12 output is enabled.	R/W
b5	A13E	Address A13 Output Enable	0: A13 output is disabled. 1: A13 output is enabled.	R/W
b6	A14E	Address A14 Output Enable	0: A14 output is disabled. 1: A14 output is enabled.	R/W
b7	A15E	Address A15 Output Enable	0: A15 output is disabled. 1: A15 output is enabled.	R/W

PF4BUS enables or disables address outputs.

ADRLE[1: 0] Bits (Address Lower A9 to A0 Output Enable)

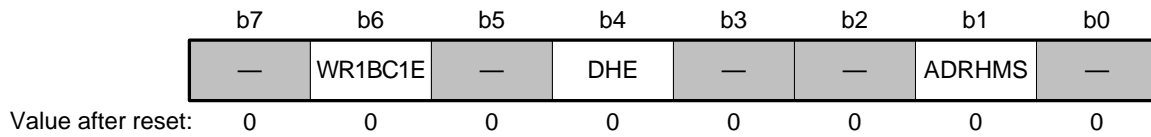
These bits enable or disable an address output (A9 to A0).

AnE Bit (Address An Output Enable) (n = 10 to 15)

Each bit enables or disables an address output (An).

17.2.2.12 Port Function Control Register 5 (PF5BUS)

Address: 0008 C105h



Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b1	ADRHMS	A19-to-A16 Output Select	A19-to-A16 Output Select 0: PC3 to PC0 are designated as the A19-A to A16-A pins (function as part of the external address bus) 1: P93 to P90 are designated as the A19-B to A16-B pins (function as part of the external address bus)	R/W
b3, b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	DHE	D15-to-D8 Enable	0: PE7 to PE0 are designated as I/O port pins. 1: PE7 to PE0 are designated as D15 to D8 pins (function as part of the external data bus)	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b6	WR1BC1E	WR1#/BC1# Output Enable	0: P51 is designated as an I/O port pin. 1: P51 is designated as the WR1# or BC1# pin.	R/W
b7	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

ADRHMS Bit (A19-to-A16 Output Select)

This bit selects the set of pins for the output of address signals A19 to A16.

DHE Bit (D15-to-D8 Enable)

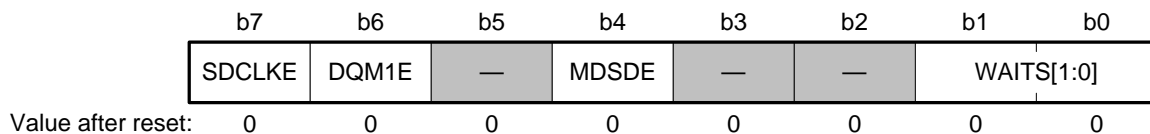
This bit enables or disables the input and output of data signals D15 to D8.

WR1BC1E Bit (WR1#/BC1# Output Enable)

This bit enables or disables WR1#/BC1# output.

17.2.2.13 Port Function Control Register 6 (PF6BUS)

Address: 0008 C106h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	WAITS[1:0]	WAIT Select	b1 b0 0 x: P55 is designated as the WAIT#-B input pin. 1 0: PC5 is designated as the WAIT#-C input pin. 1 1: P51 is designated as the WAIT#-D input pin.	R/W
b3, b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	MDSDE	SDRAM Pins Enable	See bit 6 (the DQM1E bit) in this register.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b6	DQM1E	DQM1 Output Enable	MDSDE DQM1E 0 x: Of the SDRAM pins, only the P70/SDCLK pin is available. The SDCLKE bit controls whether or not it is enabled. 1 0: All pin functions for the SDRAM (other than the P67/DQM1 pin) are enabled. 1 1: All pin functions for the SDRAM are enabled.	R/W
b7	SDCLKE	SDCLK Output Enable	0: SDCLK output is disabled. 1: SDCLK output is enabled.	R/W

[Legend] x: Don't care

WAITS Bits (WAIT Select)

These bits select a pin for a WAIT# input.

MDSDE Bit (SDRAM Pins Enable)

This bit enables or disables the output of the SDRAM pins.

While the MDSDE bit is 1, the output of the DQM1 pin is enabled or disabled independently according to the DQM1E bit setting. The output of the SDCLK pin is enabled or disabled independently according to the SDCLKE bit setting, regardless of the MDSDE bit setting.

DQM1E Bit (DQM1 Output Enable)

This bit enables or disables the output of the DQM1 pin.

When MDSDE bit is set to 1, the DQM1E bit setting is enabled. When the MDSDE bit is cleared to 0, the DQM1E bit setting is ignored.

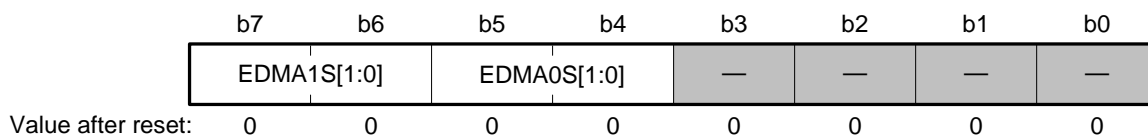
SDCLK Bit (SDCLK Output Enable)

This bit enables or disables the output of the SDCLK pin.

The SDCLK bit setting should be changed after the clock to the SDCLK has been stopped.

17.2.2.14 Port Function Control Register 7 (PF7DMA)

Address: 0008 C107h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b5, b4	EDMA0S[1: 0]	EXDMAC0 Pin Select	b5 b4 0 0: P80 is designated as the EDREQ0-A pin. P81 is designated as the EDACK0-A pin. 0 1: P22 is designated as the EDREQ0-B pin. P23 is designated as the EDACK0-B pin. 1 x: P55 is designated as the EDREQ0-C pin. P54 is designated as the EDACK0-C pin.	R/W
b7, b6	EDMA1S[1: 0]	EXDMAC1 Pin Select	b7 b6 0 0: P82 is designated as the EDREQ1-A pin. P83 is designated as the EDACK1-A pin. 0 1: P24 is designated as the EDREQ1-B pin. P25 is designated as the EDACK1-B pin. 1 x: P57 is designated as the EDREQ1-C pin. P56 is designated as the EDACK1-C pin.	R/W

[Legend] x: Don't care

EDMA_nS Bits (EXDMAC_n Pin Select) (n = 0, 1)

These bits select a pin for EXDMAC_n.

17.2.2.15 Port Function Control Register 8 (PF8IRQ)

Address: 0008 C108h

	b7	b6	b5	b4	b3	b2	b1	b0
	ITS15	—	ITS13	—	ITS11	ITS10	ITS9	ITS8
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITS8	IRQ8 Pin Select	0: P00 is designated as the IRQ8-A input pin. 1: P40 is designated as the IRQ8-B input pin.	R/W
b1	ITS9	IRQ9 Pin Select	0: P01 is designated as IRQ9-A input pin. 1: P41 is designated as IRQ9-B input pin.	R/W
b2	ITS10	IRQ10 Pin Select	0: P02 is designated as the IRQ10-A input pin. 1: P42 is designated as the IRQ10-B input pin	R/W
b3	ITS11	IRQ11 Pin Select	0: P03 is designated as the IRQ11-A input pin. 1: P43 is designated as the IRQ11-B input pin.	R/W
b4	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b5	ITS13	IRQ13 Pin Select	0: P05 is designated as the IRQ13-A input pin. 1: P45 is designated as the IRQ13-B input pin.	R/W
b6	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b7	ITS15	IRQ15 Pin Select	0: P07 is designated as the IRQ15-A input pin. 1: P47 is designated as the IRQ15-B input pin.	R/W

PF8IRQ is used to select pins for IRQ8 to IRQ11, IRQ13, and IRQ15 inputs.

ITS_i Bit (IRQ_i Pin Select) (i = 8 to 11, 13, and 15)

Each bit selects a pin for an IRQ_i input.

17.2.2.16 Port Function Control Register 9 (PF9IRQ)

Address: 0008 C109h

	b7	b6	b5	b4	b3	b2	b1	b0
	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1,.b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b2	ITS2	IRQ2 Pin Select	0: P32 is designated as the IRQ2-A input pin. 1: P12 is designated as the IRQ2-B input pin.	R/W
b3	ITS3	IRQ3 Pin Select	0: P33 is designated as the IRQ3-A input pin. 1: P13 is designated as the IRQ3-B input pin.	R/W
b4	ITS4	IRQ4 Pin Select	0: P34 is designated as the IRQ4-A input pin. 1: P14 is designated as the IRQ4-B input pin.	R/W
b5	ITS5	IRQ5 Pin Select	0: PE5 is designated as the IRQ5-A input pin. 1: P15 is designated as the IRQ5-B input pin.	R/W
b6	ITS6	IRQ6 Pin Select	0: PE6 is designated as the IRQ6-A input pin. 1: P16 is designated as the IRQ6-B input pin.	R/W
b7	ITS7	IRQ7 Pin Select	0: PE7 is designated as the IRQ7-A input pin. 1: P17 is designated as the IRQ7-B input pin.	R/W

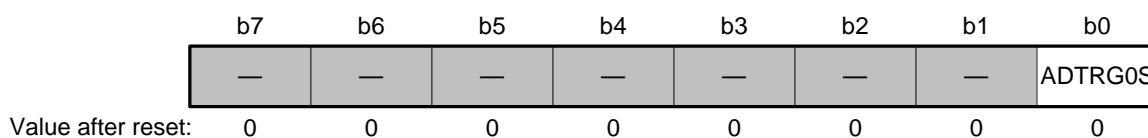
PF9IRQ is used to select pins for IRQ2 to IRQ7 inputs.

ITS_i Bit (IRQ_i Pin Select) (i = 2 to 7)

Each bit selects a pin for an IRQ_i input.

17.2.2.17 Port Function Control Register A (PFAADC)

Address: 0008 C10Ah



Bit	Symbol	Bit Name	Description	R/W
b0	ADTRG0S	ADTRG0# Input Select	0: P07 is designated as the ADTRG0#-A input pin. 1: P25 is designated as the ADTRG0#-B input pin.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

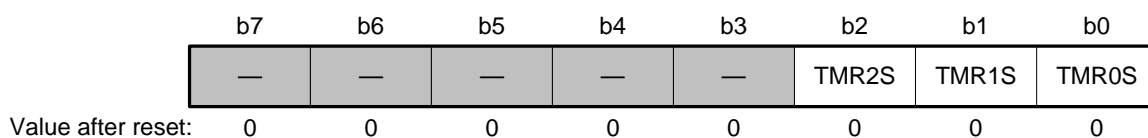
PFAADC is used to select a pin for ADTRG0# input.

ADTRG0S Bit (ADTRG0# Input Select)

This bit selects a pin for an ADTRG0# input.

17.2.2.18 Port Function Control Register B (PFBTMR)

Address: 0008 C10Bh



Bit	Symbol	Bit Name	Description	R/W
b0	TMR0S	TMR0 Input Pin Select	0: P01 is designated as the TMCI0-A pin. P00 is designated as the TMRIO-A pin. 1: P21 is designated as the TMCI0-B pin. P20 is designated as the TMRIO-B pin.	R/W
b1	TMR1S	TMR1 Input Pin Select	0: P02 is designated as the TMCI1-A pin. 1: P12 is designated as the TMCI1-B pin.	R/W
b2	TMR2S	TMR2 Input Pin Select	0: P15 is designated as the TMCI2-A pin. 1: P31 is designated as the TMCI2-B pin.	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

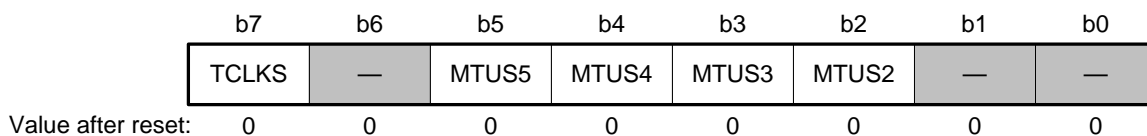
PFBTMR is used to select pins for TMR0 to TMR2.

TMRnS Bit (TMRn Input Pin Select) (n = 0 to 2)

Each bit selects a pin for a TMRn input.

17.2.2.19 Port Function Control Register C (PFCMTU)

Address: 0008 C10Ch



Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b2	MTUS2	MTU Pin Select 2	0: P16 is designated as the MTIOC3C-A pin. 1: P56 is designated as the MTIOC3C-B pin.	R/W
b3	MTUS3	MTU Pin Select 3	0: P22 is designated as the MTIOC3B-A pin. P23 is designated as the MTIOC3D-A pin. 1: P80 is designated as the MTIOC3B-B pin. P81 is designated as the MTIOC3D-B pin.	R/W
b4	MTUS4	MTU Pin Select 4	0: P24 is designated as the MTIOC4A-A pin. P25 is designated as the MTIOC4C-A pin. 1: P82 is designated as the MTIOC4A-B pin. P83 is designated as the MTIOC4C-B pin.	R/W
b5	MTUS5	MTU Pin Select 5	0: P30 is designated as the MTIOC4B-A pin. P31 is designated as the MTIOC4D-A pin. 1: P54 is designated as the MTIOC4B-B pin. P55 is designated as the MTIOC4D-B pin.	R/W
b6	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b7	TCLKS	MTCLK Pin Select	0: P24 is designated as the MTCLKA-A pin. P25 is designated as the MTCLKB-A pin. P22 is designated as the MTCLKC-A pin. P23 is designated as the MTCLKD-A pin. 1: PC6 is designated as the MTCLKA-B pin. PC7 is designated as the MTCLKB-B pin. PC4 is designated as the MTCLKC-B pin. PC5 is designated as the MTCLKD-B pin.	R/W

PFCMTU is used to select pins for MTU unit 0.

MTUSj Bit (MTU Pin Select) (j = 2 to 5)

Each bit selects a pin for an MTU input/output.

TCLKS Bit (MTCLK Pin Select)

This bit selects a pin for a MTCLK input of the MTU.

17.2.2.20 Port Function Control Register D (PFDMTU)

Address: 0008 C10Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	TCLKS	MTUS6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b6	MTUS6	MTU Pin Select 6	0: PC7 is designated as the MTIC11U-A pin. PC6 is designated as the MTIC11V-A pin. PC5 is designated as the MTIC11W-A pin. 1: PD4 is designated as the MTIC11U-B pin. PD3 is designated as the MTIC11V-B pin. PD2 is designated as the MTIC11W-B pin.	R/W
b7	TCLKS	MTCLK Pin Select	0: PC2 is designated as the MTCLKE-A pin. PC3 is designated as the MTCLKF-A pin. PC0 is designated as the MTCLKG-A pin. PC1 is designated as the MTCLKH-A pin. 1: PB4 is designated as the MTCLKE-B pin. PB5 is designated as the MTCLKF-B pin. PB2 is designated as the MTCLKG-B pin. PB3 is designated as the MTCLKH-B pin.	R/W

PFDMTU is used to select pins for MTU unit 1.

MTUS6 Bit (MTU Pin Select 6)

Each bit selects a pin for a MTIC11U/V/W input.

TCLKS Bit (MTCLK Pin Select)

This bit selects a pin for a MTCLK input of the MTU.

17.2.2.21 Port Function Control Register E (PFENET)

Address: 0008 C10Eh

b7	b6	b5	b4	b3	b2	b1	b0
EE	—	—	PHYMODE	ENETE3	ENETE2	ENETE1	ENETE0
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ENETE0	Ethernet Pin Enable 0	0: The ET_WOL pin is disabled. 1: The ET_WOL pin is enabled.	R/W
b1	ENETE1	Ethernet Pin Enable 1	0: The ET_LINKSTA pin is disabled. 1: The ET_LINKSTA pin is enabled.	R/W
b2	ENETE2	Ethernet Pin Enable 2	0: The ET_EXOUT pin is disabled. 1: The ET_EXOUT pin is enabled.	R/W
b3	ENETE3	Ethernet Pin Enable 3	0: The ET_TX_ER pin is disabled. 1: The ET_TX_ER pin is enabled.	R/W
b4	PHYMODE	Ethernet Mode Setting	0: RMI mode 1: MII mode	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	EE	Ethernet Pins Enable	0: All pin functions for the Ethernet interface are disabled. 1: All pin functions for the Ethernet interface are enabled.	R/W

PFENET is used to select I/O pins for the Ethernet interface.

ENETE0 to ENETE3 Bits (Ethernet Pin Enable)

These bits select pins for the Ethernet interface.

PHYMODE Bit (Ethernet Mode Setting)

This bit selects the PHY mode for the Ethernet interface.

Table 17.18 lists the relationship between the setting of the PHYMODE bit and the mode of the Ethernet interface.

EE Bit (Ethernet Pins Enable)

This bit enables or disables all pin functions for the Ethernet interface.

Table 17.18 Relationship between the PHYMODE Bit Setting and Ethernet Mode (145-Pin TFLGA/144-Pin LQFP)

PHYMODE	Ethernet Mode	Pin to be Used for the Ethernet Interface	Pin Allocation	Remark
0	RMII mode	ET_MDC	P72	
		ET_MDIO	P71	
		ET_WOL	P73	Enabled when ENETE0 = 1
		ET_LINKSTA	P54	Enabled when ENETE1 = 1
		ET_EXOUT	P55	Enabled when ENETE2 = 1
		REF50CK	P76	
		RMII_TXD0	P81	
		RMII_TXD1	P82	
		RMII_TXD_EN	P80	
		RMII_RXD0	P75	
		RMII_RXD1	P74	
		RMII_RX_ER	P77	
		RMII_CRSDV	P83	
		1	MII mode	ET_MDC
ET_MDIO	P71			
ET_WOL	P73			Enabled when ENETE0 = 1
ET_LINKSTA	P54			Enabled when ENETE1 = 1
ET_EXOUT	P55			Enabled when ENETE2 = 1
ET_TX_CLK	PC4			
ET_ETXD0	P81			
ET_ETXD1	P82			
ET_ETXD2	PC5			
ET_ETXD3	PC6			
ET_TX_EN	P80			
ET_TX_ER	PC3			Enabled when ENETE3 = 1
ET_COL	PC7			
ET_CRSDV	P83			
ET_RX_CLK	P76			
ET_ERXD0	P75			
ET_ERXD1	P74			
ET_ERXD2	PC1			
ET_ERXD3	PC0			
ET_RX_DV	PC2			
ET_RX_ER	P77			

17.2.2.22 Port Function Control Register F (PFFSCI)

Address: 0008 C10Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SCI6S	—	—	SCI3S	SCI2S	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b2	SCI2S	SCI2 Pin Select	0: P12 is designated as the RxD2-A pin. P51 is designated as the SCK2 pin. P13 is designated as the TxD2-A pin. 1: P52 is designated as the RxD2-B pin. P51 is designated as the SCK2 pin. P50 is designated as the TxD2-B pin.	R/W
b3	SCI3S	SCI3 Pin Select	0: P16 is designated as the RxD3-A pin. P15 is designated as the SCK3-A pin. P17 is designated as the TxD3-A pin. 1: P25 is designated as the RxD3-B pin. P24 is designated as the SCK3-B pin. P23 is designated as the TxD3-B pin.	R/W
b5, b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b6	SCI6S	SCI6 Pin Select	0: P01 is designated as the RxD6-A pin. P02 is designated as the SCK6-A pin. P00 is designated as the TxD6-A pin. 1: P33 is designated as the RxD6-B pin. P34 is designated as the SCK6-B pin. P32 is designated as the TxD6-B pin.	R/W
b7	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

PFFSCI is used to select pins for SCI.

SCInS Bit (SCIn Pin Select) (n = 2, 3, 6)

Each bit selects a pin for an SCI channel-n input/output.

17.2.2.23 Port Function Control Register G (PFGSPI)

Address: 0008 C110h

	b7	b6	b5	b4	b3	b2	b1	b0
	SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	RSPIS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPIS	RSPI Pin Select	0: PC7 is designated as the MISOA-A pin. PC6 is designated as the MOSIA-A pin. PC5 is designated as the RSPCKA-A pin. PC4 is designated as the SSLA0-A pin. PC0 is designated as the SSLA1-A pin. PC1 is designated as the SSLA2-A pin. PC2 is designated as the SSLA3-A pin. 1: PA7 is designated as the MISOA-B pin. PA6 is designated as the MOSIA-B pin. PA5 is designated as the RSPCKA-B pin. PA4 is designated as the SSLA0-B pin. PA0 is designated as the SSLA1-B pin. PA1 is designated as the SSLA2-B pin. PA2 is designated as the SSLA3-B pin.	R/W
b1	RSPCKE	RSPCKA Output Enable	0: The RSPCKA pin is disabled. 1: The RSPCKA pin is enabled.	R/W
b2	MOSIE	MOSIA Output Enable	0: The MOSIA pin is disabled. 1: The MOSIA pin is enabled.	R/W
b3	MISOE	MISOA Output Enable	0: The MISOA pin is disabled. 1: The MISOA pin is enabled.	R/W
b4	SSL0E	SSLA0 Output Enable	0: The SSLA0 pin is disabled. 1: The SSLA0 pin is enabled.	R/W
b5	SSL1E	SSLA1 Output Enable	0: The SSLA1 pin is disabled. 1: The SSLA1 pin is enabled.	R/W
b6	SSL2E	SSLA2 Output Enable	0: The SSLA2 pin is disabled. 1: The SSLA2 pin is enabled.	R/W
b7	SSL3E	SSLA3 Output Enable	0: The SSLA3 pin is disabled. 1: The SSLA3 pin is enabled.	R/W

PFGSPI is used to select I/O pins for RSPI channel 0.

RSPIS Bit (RSPI Pin Select)

This bit selects a pin for an RSPI input/output.

As an enable bit is provided for each RSPI input/output pin, the input/output pin is selectable while the corresponding enable bit is 1. Otherwise, the pin cannot be selected.

RSPCKE Bit (RSPCKA Output Enable)

This bit enables or disables the output of the RSPCKA pin. Set this bit to 1 to use the RSPCKA pin.

MOSIE Bit (MOSIA Output Enable)

This bit enables or disables the output of the MOSIA pin. Set this bit to 1 to use the MOSIA pin.

MISOE Bit (MISOA Output Enable)

This bit enables or disables the output of the MISOA pin. Set this bit to 1 to use the MISOA pin.

SSL0E Bit (SSLA0 Output Enable)

This bit enables or disables the output of the SSLA0 pin. Set this bit to 1 to use the SSLA0 pin.

SSL1E Bit (SSLA1 Output Enable)

This bit enables or disables the output of the SSLA1 pin. Set this bit to 1 to use the SSLA1 pin.

SSL2E Bit (SSLA2 Output Enable)

This bit enables or disables the output of the SSLA2 pin. Set this bit to 1 to use the SSLA2 pin.

SSL3E Bit (SSLA3 Output Enable)

This bit enables or disables the output of the SSLA3 pin. Set this bit to 1 to use the SSLA3 pin.

17.2.2.24 Port Function Control Register H (PFHSPI)

Address: 0008 C111h

b7	b6	b5	b4	b3	b2	b1	b0
SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	RSPIS
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPIS	RSPI Pin Select	0: P30 is designated as the MISOB-A pin. P26 is designated as the MOSIB-A pin. P27 is designated as the RSPCKB-A pin. P31 is designated as the SSLB0-A pin. P50 is designated as the SSLB1-A pin. P51 is designated as the SSLB2-A pin. P52 is designated as the SSLB3-A pin. 1: PE7 is designated as the MISOB-B pin. PE6 is designated as the MOSIB-B pin. PE5 is designated as the RSPCKB-B pin. PE4 is designated as the SSLB0-B pin. PE0 is designated as the SSLB1-B pin. PE1 is designated as the SSLB2-B pin. PE2 is designated as the SSLB3-B pin.	R/W
b1	RSPCKE	RSPCKB Output Enable	0: The RSPCKB pin is disabled. 1: The RSPCKB pin is enabled	R/W
b2	MOSIE	MOSIB Output Enable	0: The MOSIB pin is disabled. 1: The MOSIB pin is enabled	R/W
b3	MISOE	MISOB Output Enable	0: The MISOB pin is disabled. 1: The MISOB pin is enabled	R/W
b4	SSL0E	SSLB0 Output Enable	0: The SSLB0 pin is disabled. 1: The SSLB0 pin is enabled	R/W
b5	SSL1E	SSLB1 Output Enable	0: The SSLB1 pin is disabled. 1: The SSLB1 pin is enabled	R/W
b6	SSL2E	SSLB2 Output Enable	0: The SSLB2 pin is disabled. 1: The SSLB2 pin is enabled	R/W
b7	SSL3E	SSLB3 Output Enable	0: The SSLB3 pin is disabled. 1: The SSLB3 pin is enabled	R/W

PFHSPI is used to select I/O pins for RSPI channel 1.

RSPIS Bit (RSPI Pin Select)

This bit selects a pin for an RSPI input/output.

As an enable bit is provided for each RSPI input/output pin, the input/output pin is selectable while the corresponding enable bit is 1. Otherwise, the pin cannot be selected.

RSPCKE Bit (RSPCKB Output Enable)

This bit enables or disables the output of the RSPCKB pin. Set this bit to 1 to use the RSPCKB pin.

MOSIE Bit (MOSIB Output Enable)

This bit enables or disables the output of the MOSIB pin. Set this bit to 1 to use the MOSIB pin.

MISOE Bit (MISOB Output Enable)

This bit enables or disables the output of the MISOB pin. Set this bit to 1 to use the MISOB pin.

SSL0E Bit (SSLB0 Output Enable)

This bit enables or disables the output of the SSLB0 pin. Set this bit to 1 to use the SSLB0 pin.

SSL1E Bit (SSLB1 Output Enable)

This bit enables or disables the output of the SSLB1 pin. Set this bit to 1 to use the SSLB1 pin.

SSL2E Bit (SSLB2 Output Enable)

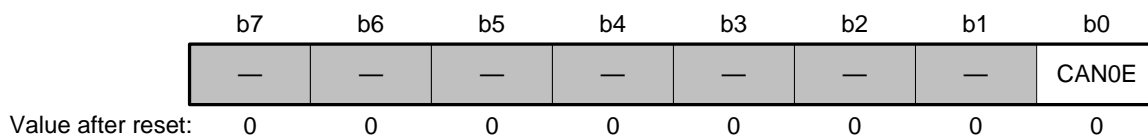
This bit enables or disables the output of the SSLB2 pin. Set this bit to 1 to use the SSLB2 pin.

SSL3E Bit (SSLB3 Output Enable)

This bit enables or disables the output of the SSLB3 pin. Set this bit to 1 to use the SSLB3 pin.

17.2.2.25 Port Function Control Register J (PFJCAN)

Address: 0008 C113h



Bit	Symbol	Bit Name	Description	R/W
b0	CAN0E	CAN0 Pins Enable	0: The CTX0 and CRX0 pins are disabled. 1: The CTX0 and CRX0 pins are enabled.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

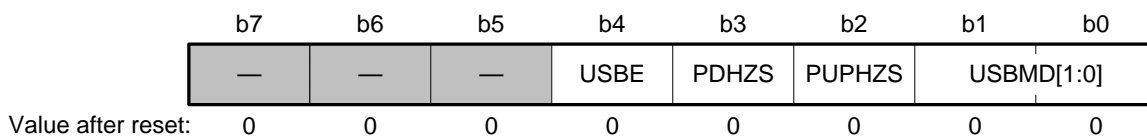
PFJCAN is used to select I/O pins for the CAN.

CANnE Bit (CANn Pins Enable) (n = 0)

This bit enables or disables the CANn pins. Set this bit to 1 to use the CANn pins.

17.2.2.26 Port Function Control Register K (PFKUSB)

Address: 0008 C114h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	USBMD[1:0]	USB Mode Setting	b1 b0 0 0: Select function mode for the USB0 pins. 0 1: Select host mode for the USB0 pins. 1 0: Select host/function mode for the USB0 pin. (as an optional function)* 1 1: Select OTG mode for the USB0 pins.	R/W
b2	PUPHZS	PUPHZ Select	0: USB0_DPUPE pin is for output of the high and low levels (external pull-up control signal). 1: USB0_DPUPE pin is for high-level output or the Hi-Z state (pull-up output is from the USB0_DP pin).	R/W
b3	PDHZS	PDHZ Select	0: USB0_DPRPD and USB0_DRPD pins are for output of the high and low levels (external pull-down control signals). 1: USB0_DPRPD and USB0_DRPD pins are for low-level output or the Hi-Z state (pull-down output is from the USB0_DP and USB0_DM pins).	R/W
b4	USBE	USB Enable	0: All pin functions for USB0 are disabled. 1: All pin functions for USB0 are enabled.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note : * Please contact a Renesas Electronics sales office for details of the optional function.

PFKUSB is used to set I/O pins for the USB0.

USBMD[1:0] Bits (USB Mode Setting)

These bits select a mode for the USB.

Table 17.19 lists the relationship between the setting of the USBMD[1:0] bits and the mode of the USB.

PUPHZS Bit (PUPHZ Select)

This bit selects the output mode (external pull-up control/pin pull-up output) for the DPUPE pin of the USB.

When the PUPHZS bit is set to 0, the control signal output mode for the external pull-up IC is selected and the high-active control signal is output from the DPUPE pin. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the low-level output state.

When the PUPHZS bit is set to 1, the output mode where the DP pin is directly pulled-up is selected. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the high-impedance state.

PDHVS Bit (PDHZ Select)

This bit selects the output mode (external pull-down control/pin pull-down output) for the DPRPD and DRPD pins of the USB.

When the PDHVS bit is set to 0, the control signal output mode for the external pull-down IC is selected and the high-active control signal is output from the DPRPD and DRPD pins. When the DP and DM pins are pulled down, the DPRPD and DRPD pins are set to the high-level output state. When the pulled-down state of the DP and DM pins is canceled, the DPRPD and DRPD pins are set to the low-level output state.

When the PDHVS bit is set to 1, the output mode where the DP and DM pins are directly pulled-down is selected. When the DP and DM pins are pulled down, the DPRPD and DRPD pins are set to the low-level output state. When the pulled-down state of the DP and DM pins is canceled, the DPRPD and DRPD pins are set to the high-impedance state.

USBE Bit (USB Enable)

This bit enables all pin functions for the USB0.

Table 17.19 Relationship between the USBMD[1:0] Bits Setting and USB Mode (USB0)

USBMD1	USBMD0	USB0 Mode	Pin to be Used for the USB	Pin Allocation	Remarks
0	0	Function mode	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_VBUS	P16	
			USB0_DPUPE-B	P14	Selection of -B side
0	1	Host mode	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_OVRCURA	P14	
			USB0_VBUSEN-B	P16	Selection of -B side
1	0	Host/function mode (as an optional function)*	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_OVRCURA	P14	
			USB0_VBUS	P16	
			USB0_DRPD	P22	
			USB0_DPUPE-A	P23	Selection of -A side
			USB0_VBUSEN-A	P24	Selection of -A side
1	1	OTG mode	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_OVRCURA	P14	
			USB0_OVRCURB	P16	
			USB0_DPRPD	P25	
			USB0_DRPD	P22	
			USB0_EXICEN	P21	
			USB0_ID	P20	
			USB0_DPUPE-A	P23	Selection of -A side
			USB0_VBUSEN-A	P24	Selection of -A side

Note : * Please contact a Renesas Electronics sales office for details of the optional function.

17.2.2.27 Port Function Control Register M (PFMPOE)

Address: 0008 C116h

	b7	b6	b5	b4	b3	b2	b1	b0
	POE7E	POE6E	POE5E	POE4E	POE3E	POE2E	POE1E	POE0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	POE0E	POE0 Input Enable	0: Designated as an I/O port pin.	R/W*
b1	POE1E	POE1 Input Enable	1: Designated as the POEn# input pin (n = 0 to 7)	R/W*
b2	POE2E	POE2 Input Enable		R/W*
b3	POE3E	POE3 Input Enable		R/W*
b4	POE4E	POE4 Input Enable		R/W*
b5	POE5E	POE5 Input Enable		R/W*
b6	POE6E	POE6 Input Enable		R/W*
b7	POE7E	POE7 Input Enable		R/W*

Note : * The first round of writing after a reset is enabled; though subsequent writing is not allowed.

PFMPOE enables or disables POE input pins.

To prevent the error in the system, be sure to write to this register after a reset.

The first round of writing after a reset is only enabled.

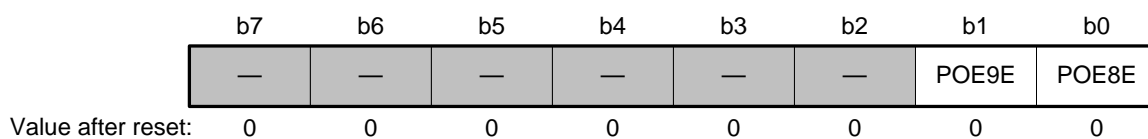
POEnE Bit (POEn Input Enable) (n = 0 to 7)

Each bit enables or disables the corresponding POEn# input.

To use POEn#, set the corresponding POEnE bit to 1.

17.2.2.28 Port Function Control Register N (PFNPOE)

Address: 0008 C117h



Bit	Symbol	Bit Name	Description	R/W
b0	POE8E	POE8 Input Enable	0: Designated as an I/O port pin.	R/W*
b1	POE9E	POE9 Input Enable	1: Designated as the POEn# input pin (n = 8, 9)	R/W*
b7 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note : * The first round of writing after a reset is enabled; though subsequent writing is not allowed.

PFNPOE enables or disables POE input pins.

To prevent the error in the system, be sure to write to this register after a reset.

The first round of writing after a reset is only enabled.

POEnE Bit (POEn Input Enable) (n = 8, 9)

Each bit enables or disables the corresponding POEn# input

To use POEn#, set the corresponding POEnE bit to 1.

17.2.3 Settings of Ports

When individual pins for peripheral modules are enabled, the settings for each port are modified.

An input pin for the peripheral module is specified independently by the peripheral module. To use an input pin for the peripheral module, the corresponding bit in the input buffer control register (ICR) should be set to 1 to enable the input buffer, except for the port register read, data bus input, NMI, and POE pin inputs.

The pins that function as output pins and I/O pins should be enabled for respective peripheral modules. If a conflict occurs among the output signal enable settings for peripheral modules, that are multiplexed to the same port, the priority will be handled according to the port-multiplexed priority.

Table 17.20 lists the port-multiplexed priority for peripheral modules.

Table 17.20 Port-Multiplexed Priority for Peripheral Modules (145-Pin TFLGA/144-Pin LQFP)

Priority	Module Name	Output Pins
High	1 External bus (Data)	D0 to D15 (Data bus)
	2 External bus SDRAM	RD#, WR#, WR0# to WR1#, BC0# to BC1#, BCLK, SDCLK, SDCS#, RAS#, CAS#, WE#, CKE, DQM0 to DQM1, A0 to A23 (Address bus)
	3 External bus (CS)	CS0# to CS7# (Chip select)
	4 RSPIO, RSP11	RSPCKn, MOSIn, MISON, SSLn0 to SSLn3 (n = A, B)
	5 USB0	USB0_DPUPE, USB0_VBUSEN, USB0_EXICEN, USB0_DRPD, USB0_DPRPD
	6 CAN0	CTX0
	7 EtherNET	ET_MDC, ET_MDIO, ET_EXOUT, ET_WOL, ET_TX_EN, ET_TX_ER, ET_ETXD0 to ET_ETXD3, RMII_TXD_EN, RMII_TXD0, RMII_TXD1
	8 EXDMAC0, EXDMAC1	EDACK0, EDACK1
	9 MTU0 to MTU4, MTU6 to MTU10	MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D, MTIOC1A, MTIOC1B, MTIOC2A, MTIOC2B, MTIOC3A, MTIOC3B, MTIOC3C, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6A, MTIOC6B, MTIOC6C, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC8A, MTIOC8B, MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D, MTIOC10A, MTIOC10B, MTIOC10C, MTIOC10D
	10 TMR0 to TMR3	TMO0 to TMO3
	11 SCI0 to SCI3, SCI5 to SCI6	SCK0 to SCK3, SCK5 to SCK6, TxD0 to TxD3, TxD5 to TxD6
	12 RTC	RTCOUT
	13 PPG0, PPG1	PO0 to PO15, PO16 to PO31
	14 RIIC0, RIIC1	SCL0 to SCL1, SDA0 to SDA1
	15 DA	DA0 to DA1
Low	16 I/O PORT	P00 to P03, P05, P07, P12 to P17, P20 to P27, P30 to P34, P50 to P52, P54 to P56, P60 to P67, P70 to P77, P80 to P83, P90 to P93, PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7

17.2.4 List of Output Enable Settings

Table 17.21 lists the output enable settings for each port.

For details on the applicable output signals, see descriptions of the registers for each peripheral module.

Setting the port function control register changes the functions of peripheral-module pins with names ending in A to D.

Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP) (1 / 12)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P00	SCI6	TxD6-A	PFFSCI.SCI6S = 0	SCI6.SCR.TE = 1
	PORT0	P00		PORT0.DDR.B0 = 1
P01	PORT0	P01		PORT0.DDR.B1 = 1
P02	SCI6	SCK6-A	PFFSCI.SCI6S = 0	When SCI6.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI6.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PORT0	P02		PORT0.DDR.B2 = 1
P03	DA	DA0		DACR.DAOE0 = 1
	PORT0	P03		PORT0.DDR.B3 = 1
P05	DA	DA1		DACR.DAOE1 = 1
	PORT0	P05		PORT0.DDR.B5 = 1
P07	PORT0	P07		PORT0.DDR.B7 = 1
P12	RIIC0	SCL0		RIIC0.ICCR1.ICE = 1
	PORT1	P12		PORT1.DDR.B2 = 1
P13	TMR3	TMO3		TMO3.TCSR.OSA[1: 0] = 01/10/11 or TMO3.TCSR.OSB[1: 0] = 01/10/11
	SCI2	TxD2-A	PFFSCI.SCI2S = 0	SCI2.SCR.TE = 1
	RIIC0	SDA0		RIIC0.ICCR1.ICE = 1
	PORT1	P13		PORT1.DDR.B3 = 1
P14	USB0	USB0_DPUPE-B	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 00	(The signal output state is specified by the peripheral module settings.)
	PORT1	P14		PORT1.DDR.B4 = 1
P15	MTU0	MTIOC0B		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	SCI3	SCK3-A	PFFSCI.SCI3S = 0	When SCI3.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI3.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO13		PPG0.NDERH.NDER13 = 1
	PORT1	P15		PORT1.DDR.B5 = 1

Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP) (2 / 12)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P16	USB0	USB0_VBUSEN-B	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 01	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3C-A	PFCMTU.MTUS2 = 0	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	TMR2	TMO2		TMO2.TCSR.OSA[1: 0] = 01/10/11 or TMO2.TCSR.OSB[1: 0] = 01/10/11
	PPG0	PO14		PPG0.NDERH.NDER14 = 1
	PORT1	P16		PORT1.DDR.B6 = 1
P17	MTU3	MTIOC3A		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	SCI3	TxD3-A	PFFSCI.SCI3S = 0	SCI3.SCR.TE = 1
	PPG0	PO15		PPG0.NDERH.NDER15 = 1
	PORT1	P17		PORT1.DDR.B7 = 1
P20	MTU1	MTIOC1A		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	SCI0	TxD0		SCI0.SCR.TE = 1
	PPG0	PO0		PPG0.NDERL.NDER0 = 1
	RIIC1	SDA1		RIIC1.ICCR1.ICE = 1
	PORT2	P20		PORT2.DDR.B0 = 1
P21	USB0	USB0_EXICEN	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 11	(The signal output state is specified by the peripheral module settings.)
	MTU1	MTIOC1B		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO1		PPG0.NDERL.NDER1 = 1
	RIIC1	SCL1		RIIC1.ICCR1.ICE = 1
	PORT2	P21		PORT2.DDR.B1 = 1
P22	USB0	USB0_DRPD	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 10/11	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3B-A	PFCMTU.MTUS3 = 0	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	TMR0	TMO0		TMO0.TCSR.OSA[1: 0] = 01/10/11 or TMO0.TCSR.OSB[1: 0] = 01/10/11
	SCI0	SCK0		When SCI0.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI0.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO2		PPG0.NDERL.NDER2 = 1
PORT2	P22		PORT2.DDR.B2 = 1	

Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP) (3 / 12)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P23	USB0	USB0_DPUPE-A	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 10/11	(The signal output state is specified by the peripheral module settings.)
	EXDMAC0	EDACK0-B	PF7DMA.EDMA0S[1: 0] = 01	EXDMAC0.EDMOMD.DACKE = 1 (The signal output state is specified by the EXDMAC0 settings.)
	MTU3	MTIOC3D-A	PFCMTU.MTUS3 = 0	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	SCI3	TxD3-B	PFFSCI.SCI3S = 1	SCI3.SCR.TE = 1
	PPG0	PO3		PPG0.NDERL.NDER3 = 1
	PORT2	P23		PORT2.DDR.B3 = 1
P24	External bus (CS)	CS4#-C	PF0CSE.CS4E = 1 PF1CSS.CS4S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	USB0	USB0_VBUSEN-A	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 10/11	(The signal output state is specified by the peripheral module settings.)
	MTU4	MTIOC4A-A	PFCMTU.MTUS4 = 0	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	SCI3	SCK3-B	PFFSCI.SCI3S = 1	When SCI3.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI3.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO4		PPG0.NDERL.NDER4 = 1
	PORT2	P24		PORT2.DDR.B4 = 1
P25	External bus (CS)	CS5#-C	PF0CSE.CS5E = 1 PF1CSS.CS5S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	USB0	USB0_DPRPD	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 11	(The signal output state is specified by the peripheral module settings.)
	EXDMAC1	EDACK1-B	PF7DMA.EDMA1S[1: 0] = 01	EDMAC1.EDMOMD.DACKE = 1 (The signal output state is specified by the EXDMAC1 settings.)
	MTU4	MTIOC4C-A	PFCMTU.MTUS4 = 0	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO5		PPG0.NDERL.NDER5 = 1
	PORT2	P25		PORT2.DDR.B5 = 1
P26	External bus (CS)	CS6#-C	PF0CSE.CS6E = 1 PF1CSS.CS6S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSP11	MOSIB-A	PFHSPI.MOSIE = 1 PFHSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU2	MTIOC2A		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	TMR1	TMO1		TMO1.TCSR.OSA[1: 0] = 01/10/11 or TMO1.TCSR.OSB[1: 0] = 01/10/11
	SCI1	TxD1		SCI1.SCR.TE = 1
	PPG0	PO6		PPG0.NDERL.NDER6 = 1
	PORT2	P26		PORT2.DDR.B6 = 1

Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP) (4 / 12)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P27	External bus (CS)	CS7#-C	PF0CSE.CS7E = 1 PF1CSS.CS7S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSP11	RSPCKB-A	PFHSPI.RSPCKE = 1 PFHSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU2	MTIOC2B		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	SCI1	SCK1		When SCI1.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI1.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO7		PPG0.NDERL.NDER7 = 1
	PORT2	P27		PORT2.DDR.B7 = 1
P30	RSP11	MISOB-A	PFHSPI.MISOE = 1 PFHSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU4	MTIOC4B-A	PFCMTU.MTUS5 = 0	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO8		PPG0.NDERH.NDER8 = 1
	PORT3	P30		PORT3.DDR.B0 = 1
P31	RSP11	SSLB0-A	PFHSPI.SSL0E = 1 PFHSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU4	MTIOC4D-A	PFCMTU.MTUS5 = 0	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO9		PPG0.NDERH.NDER9 = 1
	PORT3	P31		PORT3.DDR.B1 = 1
P32	CAN0	CTX0	PFJCAN.CAN0E = 1	(The signal output state is specified by the peripheral module settings.)
	MTU0	MTIOC0C		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	SCI6	TxD6-B	PFFSCI.SCI6S = 1	SCI6.SCR.TE = 1
	RTC	RTCOUT		RCR2.RTCOE = 1
	PPG0	PO10		PPG0.NDERH.NDER10 = 1
	PORT3	P32		PORT3.DDR.B2 = 1
P33	MTU0	MTIOC0D		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO11		PPG0.NDERH.NDER11 = 1
	PORT3	P33		PORT3.DDR.B3 = 1
P34	MTU0	MTIOC0A		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	SCI6	SCK6-B	PFFSCI.SCI6S = 1	When SCI6.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI6.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO12		PPG0.NDERH.NDER12 = 1
	PORT3	P34		PORT3.DDR.B4 = 1
P35	(NA)	(NA)		
P40	PORT4	P40		PORT4.DDR.B0 = 1
P41	PORT4	P41		PORT4.DDR.B1 = 1

Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP) (5 / 12)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P42	PORT4	P42		PORT4.DDR.B2 = 1
P43	PORT4	P43		PORT4.DDR.B3 = 1
P44	PORT4	P44		PORT4.DDR.B4 = 1
P45	PORT4	P45		PORT4.DDR.B5 = 1
P46	PORT4	P46		PORT4.DDR.B6 = 1
P47	PORT4	P47		PORT4.DDR.B7 = 1
P50	External bus	WR# WR0#		SYSCR0.EXBE = 1
	RSPI1	SSLB1-A	PFHSPI.SSL1E = 1 PFHSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	SCI2	TxD2-B	PFFSCI.SCI2S = 1	SCI2.SCR.TE = 1
	PORT5	P50		PORT5.DDR.B0 = 1
P51	External bus	WR1# BC1#	PF5BUS.WR1BC1E = 1	SYSCR0.EXBE = 1
	RSPI1	SSLB2-A	PFHSPI.SSL2E = 1 PFHSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	SCI2	SCK2	PFFSCI.SCI2S = 1	When SCI2.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI2.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PORT5	P51		PORT5.DDR.B1 = 1
P52	External bus	RD#	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	RSPI1	SSLB3-A	PFHSPI.SSL3E = 1 PFHSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	PORT5	P52		PORT5.DDR.B2 = 1
P53	External bus	BCLK		PORT5.DDR.B3 = 1
P54	EXDMAC0	EDACK0-C	PF7DMA.EDMA0S[1: 0] = 11/10	EXDMAC0.EDMOMD.DACKE = 1 (The signal output state is specified by the EXDMAC0 settings.)
	MTU4	MTIOC4B-B	PFCMTU.MTUS5 = 1	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PORT5	P54		PORT5.DDR.B4 = 1
P55	EtherNET	ET_EXOUT	PFENET.EE = 1 PFENET.ENETE2 = 1	(The signal output state is specified by the peripheral module settings.)
	MTU4	MTIOC4D-B	PFCMTU.MTUS5 = 1	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PORT5	P55		PORT5.DDR.B5 = 1
P56	EXDMAC1	EDACK1-C	PF7DMA.EDMA1S[1: 0] = 11/10	EDMAC1.EDMOMD.DACKE = 1 (The signal output state is specified by the EXDMAC1 settings.)
	MTU3	MTIOC3C-B	PFCMTU.MTUS2 = 1	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PORT5	P56		PORT5.DDR.B6 = 1
P60	External bus (CS)	CS0#-A	PF0CSE.CS0E = 1 PF2CSS.CS0S = 0	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P60		PORT6.DDR.B0 = 1

Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP) (6 / 12)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P61	SDRAM	SDCS#	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS1#-A	PF0CSE.CS1E = 1 PF2CSS.CS1S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P61		PORT6.DDR.B1 = 1
P62	SDRAM	RAS#	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS2#-A	PF0CSE.CS2E = 1 PF2CSS.CS2S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P62		PORT6.DDR.B2 = 1
P63	SDRAM	CAS#	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS3#-A	PF0CSE.CS3E = 1 PF2CSS.CS3S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P63		PORT6.DDR.B3 = 1
P64	SDRAM	WE#	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS4#-A	PF0CSE.CS4E = 1 PF1CSS.CS4S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P64		PORT6.DDR.B4 = 1
P65	SDRAM	CKE	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS5#-A	PF0CSE.CS5E = 1 PF1CSS.CS5S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P65		PORT6.DDR.B5 = 1
P66	SDRAM	DQM0	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS6#-A	PF0CSE.CS6E = 1 PF1CSS.CS6S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P66		PORT6.DDR.B6 = 1
P67	SDRAM	DQM1	PF6BUS.MDSDE = 1 PF6BUS.DQM1E = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS7#-A	PF0CSE.CS7E = 1 PF1CSS.CS7S[1: 0] = 00	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT6	P67		PORT6.DDR.B7 = 1
P70	SDRAM	SDCLK	PF6BUS.SDCLKE = 1	
	PORT7	P70		PORT7.DDR.B0 = 1
P71	External bus (CS)	CS1#-B	PF0CSE.CS1E = 1 PF2CSS.CS1S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	EtherNET	ET_MDIO	PFENET.EE = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORT7	P71		PORT7.DDR.B1 = 1
P72	External bus (CS)	CS2#-B	PF0CSE.CS2E = 1 PF2CSS.CS2S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	EtherNET	ET_MDC	PFENET.EE = 1	(The signal output state is specified by the peripheral module settings.)
	PORT7	P72		PORT7.DDR.B2 = 1

Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP) (7 / 12)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P73	External bus (CS)	CS3#-B	PF0CSE.CS3E = 1 PF2CSS.CS3S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	EtherNET	ET_WOL	PFENET.EE = 1 PFENET.ENETE0 = 1	(The signal output state is specified by the peripheral module settings.)
	PORT7	P73		PORT7.DDR.B3 = 1
P74	External bus (CS)	CS4#-B	PF0CSE.CS4E = 1 PF1CSS.CS4S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT7	P74		PORT7.DDR.B4 = 1
P75	External bus (CS)	CS5#-B	PF0CSE.CS5E = 1 PF1CSS.CS5S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT7	P75		PORT7.DDR.B5 = 1
P76	External bus (CS)	CS6#-B	PF0CSE.CS6E = 1 PF1CSS.CS6S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT7	P76		PORT7.DDR.B6 = 1
P77	External bus (CS)	CS7#-B	PF0CSE.CS7E = 1 PF1CSS.CS7S[1: 0] = 01	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	PORT7	P77		PORT7.DDR.B7 = 1
P80	EtherNET	ET_TX_EN	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	EtherNET	RMII_TXD_EN	PFENET.EE = 1 PFENET.PHYMODE = 0	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3B-B	PFCMTU.MTUS3 = 1	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PORT8	P80		PORT8.DDR.B0 = 1
P81	EtherNET	ET_ETXD0	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	EtherNET	RMII_TXD0	PFENET.EE = 1 PFENET.PHYMODE = 0	(The signal output state is specified by the peripheral module settings.)
	EXDMAC0	EDACK0-A	PF7DMA.EDMA0S[1: 0] = 00	EXDMAC0.EDMOMD.DACKE = 1 (The signal output state is specified by the EXDMAC0 settings.)
	MTU3	MTIOC3D-B	PFCMTU.MTUS3 = 1	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PORT8	P81		PORT8.DDR.B1 = 1
P82	EtherNET	ET_ETXD1	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	EtherNET	RMII_TXD1	PFENET.EE = 1 PFENET.PHYMODE = 0	(The signal output state is specified by the peripheral module settings.)
	MTU4	MTIOC4A-B	PFCMTU.MTUS4 = 1	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PORT8	P82		PORT8.DDR.B2 = 1
P83	EXDMAC1	EDACK1-A	PF7DMA.EDMA1S[1: 0] = 00	EDMAC1.EDMOMD.DACKE = 1 (The signal output state is specified by the EXDMAC1 settings.)
	MTU4	MTIOC4C-B	PFCMTU.MTUS4 = 1	For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PORT8	P83		PORT8.DDR.B3 = 1

Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP) (8 / 12)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P90	External bus	A16-B	PF3BUS.A16E = 1 PF5BUS.ADRHMS = 1	SYSCR0.EXBE = 1
	PORT9	P90		PORT9.DDR.B0 = 1
P91	External bus	A17-B	PF3BUS.A17E = 1 PF5BUS.ADRHMS = 1	SYSCR0.EXBE = 1
	PORT9	P91		PORT9.DDR.B1 = 1
P92	External bus	A18-B	PF3BUS.A18E = 1 PF5BUS.ADRHMS = 1	SYSCR0.EXBE = 1
	PORT9	P92		PORT9.DDR.B2 = 1
P93	External bus	A19-B	PF3BUS.A19E = 1 PF5BUS.ADRHMS = 1	SYSCR0.EXBE = 1
	PORT9	P93		PORT9.DDR.B3 = 1
PA0	External bus	A0 BC0#	PF4BUS.ADRLE[1: 0] = 01/10/11	SYSCR0.EXBE = 1
	RSPI0	SSLA1-B	PFGSPI.SSL1E = 1 PFGSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	MTU6	MTIOC6A		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO16		PPG1.NDERL.NDER0 = 1
	PORTA	PA0		PORTA.DDR.B0 = 1
PA1	External bus	A1	PF4BUS.ADRLE[1: 0] = 01/10/11	SYSCR0.EXBE = 1
	RSPI0	SSLA2-B	PFGSPI.SSL2E = 1 PFGSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	MTU6	MTIOC6B		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO17		PPG1.NDERL.NDER1 = 1
	PORTA	PA1		PORTA.DDR.B1 = 1
PA2	External bus	A2	PF4BUS.ADRLE[1: 0] = 01/10/11	SYSCR0.EXBE = 1
	RSPI0	SSLA3-B	PFGSPI.SSL3E = 1 PFGSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	MTU6	MTIOC6C		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO18		PPG1.NDERL.NDER2 = 1
	PORTA	PA2		PORTA.DDR.B2 = 1
PA3	External bus	A3	PF4BUS.ADRLE[1: 0] = 01/10/11	SYSCR0.EXBE = 1
	MTU6	MTIOC6D		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO19		PPG1.NDERL.NDER3 = 1
	PORTA	PA3		PORTA.DDR.B3 = 1
PA4	External bus	A4	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPI0	SSLA0-B	PFGSPI.SSL0E = 1 PFGSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU7	MTIOC7A		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO20		PPG1.NDERL.NDER4 = 1
	PORTA	PA4		PORTA.DDR.B4 = 1

Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP) (9 / 12)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PA5	External bus	A5	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPi0	RSPCKA-B	PFGSPI.RSPCKE = 1 PFGSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU7	MTIOC7B		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO21		PPG1.NDERL.NDER5 = 1
	PORTA	PA5		PORTA.DDR.B5 = 1
PA6	External bus	A6	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPi0	MOSIA-B	PFGSPI.MOSIE = 1 PFGSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU8	MTIOC8A		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO22		PPG1.NDERL.NDER6 = 1
	PORTA	PA6		PORTA.DDR.B6 = 1
PA7	External bus	A7	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPi0	MISOA-B	PFGSPI.MISOE = 1 PFGSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU8	MTIOC8B		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO23		PPG1.NDERL.NDER7 = 1
	PORTA	PA7		PORTA.DDR.B7 = 1
PB0	External bus	A8	PF4BUS.ADRLE[1: 0] = 11	SYSCR0.EXBE = 1
	MTU9	MTIOC9A		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO24		PPG1.NDERH.NDER8 = 1
	PORTB	PB0		PORTB.DDR.B0 = 1
PB1	External bus	A9	PF4BUS.ADRLE[1: 0] = 11	SYSCR0.EXBE = 1
	MTU9	MTIOC9C		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO25		PPG1.NDERH.NDER9 = 1
	PORTB	PB1		PORTB.DDR.B1 = 1
PB2	External bus	A10	PF4BUS.A10E = 1	SYSCR0.EXBE = 1
	MTU9	MTIOC9B		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO26		PPG1.NDERH.NDER10 = 1
	PORTB	PB2		PORTB.DDR.B2 = 1
PB3	External bus	A11	PF4BUS.A11E = 1	SYSCR0.EXBE = 1
	MTU9	MTIOC9D		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO27		PPG1.NDERH.NDER11 = 1
	PORTB	PB3		PORTB.DDR.B3 = 1

Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP) (10 / 12)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PB4	External bus	A12	PF4BUS.A12E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10A		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO28		PPG1.NDERH.NDER12 = 1
	PORTB	PB4		PORTB.DDR.B4 = 1
PB5	External bus	A13	PF4BUS.A13E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10C		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO29		PPG1.NDERH.NDER13 = 1
	PORTB	PB5		PORTB.DDR.B5 = 1
PB6	External bus	A14	PF4BUS.A14E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10B		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO30		PPG1.NDERH.NDER14 = 1
	PORTB	PB6		PORTB.DDR.B6 = 1
PB7	External bus	A15	PF4BUS.A15E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10D		For the MTU settings, see Table 17.22, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO31		PPG1.NDERH.NDER15 = 1
	PORTB	PB7		PORTB.DDR.B7 = 1
PC0	External bus	A16-A	PF3BUS.A16E = 1 PF5BUS.ADRHMS = 0	SYSCR0.EXBE = 1
	RSPI0	SSLA1-A	PFGSPI.SSL1E = 1 PFGSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	PORTC	PC0		PORTC.DDR.B0 = 1
PC1	External bus	A17-A	PF3BUS.A17E = 1 PF5BUS.ADRHMS = 0	SYSCR0.EXBE = 1
	RSPI0	SSLA2-A	PFGSPI.SSL2E = 1 PFGSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	SCI5	SCK5		When SCI5.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI5.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PORTC	PC1		PORTC.DDR.B1 = 1
PC2	External bus	A18-A	PF3BUS.A18E = 1 PF5BUS.ADRHMS = 0	SYSCR0.EXBE = 1
	RSPI0	SSLA3-A	PFGSPI.SSL3E = 1 PFGSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	PORTC	PC2		PORTC.DDR.B2 = 1
PC3	External bus	A19-A	PF3BUS.A19E = 1 PF5BUS.ADRHMS = 0	SYSCR0.EXBE = 1
	EtherNET	ET_TX_ER	PFENET.EE = 1 PFENET.ENETE3 = 1	(The signal output state is specified by the peripheral module settings.)
	SCI5	TxD5		SCI5.SCR.TE = 1
	PORTC	PC3		PORTC.DDR.B3 = 1

Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP) (11 / 12)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PC4	External bus	A20	PF3BUS.A20E = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS3#-C	PF0CSE.CS3E = 1 PF2CSS.CS3S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSPI0	SSLA0-A	PFGSPI.SSL0E = 1 PFGSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTC	PC4		PORTC.DDR.B4 = 1
PC5	External bus	A21	PF3BUS.A21E = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS2#-C	PF0CSE.CS2E = 1 PF2CSS.CS2S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSPI0	RSPCKA-A	PFGSPI.RSPCKE = 1 PFGSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	EtherNET	ET_ETXD2	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	PORTC	PC5		PORTC.DDR.B5 = 1
PC6	External bus	A22	PF3BUS.A22E = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS1#-C	PF0CSE.CS1E = 1 PF2CSS.CS1S[1: 0] = 11/10	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSPI0	MOSIA-A	PFGSPI.MOSIE = 1 PFGSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	EtherNET	ET_ETXD3	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	PORTC	PC6		PORTC.DDR.B6 = 1
PC7	External bus	A23	PF3BUS.A23E = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS0#-B	PF0CSE.CS0E = 1 PF2CSS.CS0S = 1	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSPI0	MISOA-A	PFGSPI.MISOE = 1 PFGSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTC	PC7		PORTC.DDR.B7 = 1
PD0	External bus (Data)	D0		SYSCR0.EXBE = 1
	PORTD	PD0		PORTD.DDR.B0 = 1
PD1	External bus (Data)	D1		SYSCR0.EXBE = 1
	PORTD	PD1		PORTD.DDR.B1 = 1
PD2	External bus (Data)	D2		SYSCR0.EXBE = 1
	PORTD	PD2		PORTD.DDR.B2 = 1
PD3	External bus (Data)	D3		SYSCR0.EXBE = 1
	PORTD	PD3		PORTD.DDR.B3 = 1
PD4	External bus (Data)	D4		SYSCR0.EXBE = 1
	PORTD	PD4		PORTD.DDR.B4 = 1
PD5	External bus (Data)	D5		SYSCR0.EXBE = 1
	PORTD	PD5		PORTD.DDR.B5 = 1

Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP) (12 / 12)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PD6	External bus (Data)	D6		SYSCR0.EXBE = 1
	PORTD	PD6		PORTD.DDR.B6 = 1
PD7	External bus (Data)	D7		SYSCR0.EXBE = 1
	PORTD	PD7		PORTD.DDR.B7 = 1
PE0	External bus (Data)	D8	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB1-B	PFHSPI.SSL1E = 1 PFHSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	PORTE	PE0		PORTE.DDR.B0 = 1
PE1	External bus (Data)	D9	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB2-B	PFHSPI.SSL2E = 1 PFHSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	PORTE	PE1		PORTE.DDR.B1 = 1
PE2	External bus (Data)	D10	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB3-B	PFHSPI.SSL3E = 1 PFHSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	PORTE	PE2		PORTE.DDR.B2 = 1
PE3	External bus (Data)	D11	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	PORTE	PE3		PORTE.DDR.B3 = 1
PE4	External bus (Data)	D12	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB0-B	PFHSPI.SSL0E = 1 PFHSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTE	PE4		PORTE.DDR.B4 = 1
PE5	External bus (Data)	D13	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	RSPCKB-B	PFHSPI.RSPCKE = 1 PFHSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTE	PE5		PORTE.DDR.B5 = 1
PE6	External bus (Data)	D14	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	MOSIB-B	PFHSPI.MOSIE = 1 PFHSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTE	PE6		PORTE.DDR.B6 = 1
PE7	External bus (Data)	D15	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	MISOB-B	PFHSPI.MISOE = 1 PFHSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTE	PE7		PORTE.DDR.B7 = 1

Table 17.22 Settings to Enable Output on the Various MTU Pins (1 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins	
MTU0	MTIOC0A	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TIORH.IOA[3] = 0 MTU0.TIORH.IOA[1:0] = 01/10/11	
		PWM mode 1	MTU0.TMDR.MD[3:0] = 0010	MTU0.TIORH.IOA[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	Except MTU0.TCR.CCLR[2:0] = 001 MTU0.TIORH.IOA[1:0] = 01/10/11	
	MTIOC0B	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TIORH.IOB[3] = 0 MTU0.TIORH.IOB[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	Except MTU0.TCR.CCLR[2:0] = 010 MTU0.TIORH.IOB[1:0] = 01/10/11	
	MTIOC0C	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TMDR.BFA = 0 MTU0.TIORL.IOC[3] = 0 MTU0.TIORL.IOC[1:0] = 01/10/11	
		PWM mode 1	MTU0.TMDR.MD[3:0] = 0010	MTU0.TMDR.BFA = 0 MTU0.TMDR.BFB = 0 MTU0.TIORL.IOC[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	MTU0.TMDR.BFA = 0 Except MTU0.TCR.CCLR[2:0] = 101 MTU0.TIORL.IOC[1:0] = 01/10/11	
	MTIOC0D	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TMDR.BFB = 0 MTU0.TIORL.IOD[3] = 0 MTU0.TIORL.IOD[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	MTU0.TMDR.BFB = 0 Except MTU0.TCR.CCLR[2:0] = 110 MTU0.TIORL.IOD[1:0] = 01/10/11	
	MTU1	MTIOC1A	Normal operation	MTU1.TMDR.MD[3:0] = 0000	MTU1.TIOR.IOA[3] = 0 MTU1.TIOR.IOA[1:0] = 01/10/11
			PWM mode 1	MTU1.TMDR.MD[3:0] = 0010	MTU1.TIOR.IOA[1:0] = 01/10/11
PWM mode 2			MTU1.TMDR.MD[3:0] = 0011	Except MTU1.TCR.CCLR[1:0] = 01 MTU1.TIOR.IOA[1:0] = 01/10/11	
Phase count mode 1			MTU1.TMDR.MD[3:0] = 0100	MTU1.TIOR.IOA[3] = 0 MTU1.TIOR.IOA[1:0] = 01/10/11	
Phase count mode 2			MTU1.TMDR.MD[3:0] = 0101		
Phase count mode 3			MTU1.TMDR.MD[3:0] = 0110		
Phase count mode 4			MTU1.TMDR.MD[3:0] = 0111		
MTIOC1B		Normal operation	MTU1.TMDR.MD[3:0] = 0000	MTU1.TIOR.IOB[3] = 0 MTU1.TIOR.IOB[1:0] = 01/10/11	
		PWM mode 2	MTU1.TMDR.MD[3:0] = 0011	Except MTU1.TCR.CCLR[1:0] = 10 MTU1.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 1	MTU1.TMDR.MD[3:0] = 0100	MTU1.TIOR.IOB[3] = 0 MTU1.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 2	MTU1.TMDR.MD[3:0] = 0101		
		Phase count mode 3	MTU1.TMDR.MD[3:0] = 0110		
		Phase count mode 4	MTU1.TMDR.MD[3:0] = 0111		
		MTU2	MTIOC2A	Normal operation	MTU2.TMDR.MD[3:0] = 0000
PWM mode 1				MTU2.TMDR.MD[3:0] = 0010	MTU2.TIOR.IOA[1:0] = 01/10/11
PWM mode 2	MTU2.TMDR.MD[3:0] = 0011			Except MTU2.TCR.CCLR[1:0] = 01 MTU2.TIOR.IOA[1:0] = 01/10/11	
Phase count mode 1	MTU2.TMDR.MD[3:0] = 0100			MTU2.TIOR.IOA[3] = 0 MTU2.TIOR.IOA[1:0] = 01/10/11	
Phase count mode 2	MTU2.TMDR.MD[3:0] = 0101				
Phase count mode 3	MTU2.TMDR.MD[3:0] = 0110				
Phase count mode 4	MTU2.TMDR.MD[3:0] = 0111				

Table 17.22 Settings to Enable Output on the Various MTU Pins (2 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins
MTU2	MTIOC2B	Normal operation	MTU2.TMDR.MD[3:0] = 0000	MTU2.TIOR.IOB[3] = 0 MTU2.TIOR.IOB[1:0] = 01/10/11
		PWM mode 2	MTU2.TMDR.MD[3:0] = 0011	Except MTU2.TCR.CCLR[1:0] = 10 MTU2.TIOR.IOB[1:0] = 01/10/11
		Phase count mode 1	MTU2.TMDR.MD[3:0] = 0100	MTU2.TIOR.IOB[3] = 0
		Phase count mode 2	MTU2.TMDR.MD[3:0] = 0101	MTU2.TIOR.IOB[1:0] = 01/10/11
		Phase count mode 3	MTU2.TMDR.MD[3:0] = 0110	
		Phase count mode 4	MTU2.TMDR.MD[3:0] = 0111	
MTU3	MTIOC3A	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTU3.TIORH.IOA[3] = 0 MTU3.TIORH.IOA[1:0] = 01/10/11
		PWM mode 1	MTU3.TMDR.MD[3:0] = 0010	MTU3.TIORH.IOA[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU3.TMDR.MD[3:0] = 1000	MTUA.TOCR1.PSYE = 1
		Complementary PWM mode 1	MTU3.TMDR.MD[3:0] = 1101	
		Complementary PWM mode 2	MTU3.TMDR.MD[3:0] = 1110	
		Complementary PWM mode 3	MTU3.TMDR.MD[3:0] = 1111	
	MTIOC3B	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTUA.TOER.OE3B = 1 MTU3.TIORH.IOB[3] = 0 MTU3.TIORH.IOB[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU3.TMDR.MD[3:0] = 1000	MTUA.TOER.OE3B = 1
		Complementary PWM mode 1	MTU3.TMDR.MD[3:0] = 1101	
		Complementary PWM mode 2	MTU3.TMDR.MD[3:0] = 1110	
		Complementary PWM mode 3	MTU3.TMDR.MD[3:0] = 1111	
		MTIOC3C	Normal operation	MTU3.TMDR.MD[3:0] = 0000
	PWM mode 1		MTU3.TMDR.MD[3:0] = 0010	MTU3.TMDR.BFA = 0 MTU3.TMDR.BFB = 0 MTU3.TIORL.IOC[1:0] = 01/10/11
	MTIOC3D	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTUA.TOER.OE3D = 1 MTU3.TMDR.BFB = 0 MTU3.TIORL.IOD[3] = 0 MTU3.TIORL.IOD[1:0] = 01/10/11
			MTU3.TMDR.MD[3:0] = 1000	MTUA.TOER.OE3D = 1
MTU3.TMDR.MD[3:0] = 1101				
MTU3.TMDR.MD[3:0] = 1110				
MTU3.TMDR.MD[3:0] = 1111				
MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000			MTUA.TOER.OE4A = 1	
MTU4	MTIOC4A	Normal operation	MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4A = 1 MTU4.TIORH.IOA[3] = 0 MTU4.TIORH.IOA[1:0] = 01/10/11
		PWM mode 1	MTU4.TMDR.MD[3:0] = 0010	MTUA.TOER.OE4A = 1 MTU4.TIORH.IOA[1:0] = 01/10/11
	– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000	MTUA.TOER.OE4A = 1	
	– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101		
	– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110		
	– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111		
	Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4A = 1	
	Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000		
	Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000		
	Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000		

Table 17.22 Settings to Enable Output on the Various MTU Pins (3 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins
MTU4	MTIOC4B	Normal operation	MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4B = 1 MTU4.TIORH.IOB[3] = 0 MTU4.TIORH.IOB[1:0] = 01/10/11
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000	MTUA.TOER.OE4B = 1
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4B = 1
		Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000	
		MTIOC4C	MTIOC4C	Normal operation
PWM mode 1	MTU4.TMDR.MD[3:0] = 0010			MTUA.TOER.OE4C = 1 MTU4.TMDR.BFA = 0 MTU4.TMDR.BFB = 0 MTU4.TIORL.IOC[1:0] = 01/10/11
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000			MTUA.TOER.OE4C = 1 MTU4.TMDR.BFA = 0
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101			
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110			
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111			
Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000			MTUA.TOER.OE4C = 1 MTU4.TMDR.BFA = 0
Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000			
Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000			
Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000			
MTIOC4D	MTIOC4D	Normal operation	MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4D = 1 MTU4.TMDR.BFB = 0 MTU4.TIORL.IOD[3] = 0 MTU4.TIORL.IOD[1:0] = 01/10/11
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000	MTUA.TOER.OE4D = 1
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4D = 1
		Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000	

Table 17.22 Settings to Enable Output on the Various MTU Pins (4 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins		
MTU6	MTIOC6A	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TIORH.IOA[3] = 0 MTU6.TIORH.IOA[1:0] = 01/10/11		
		PWM mode 1	MTU6.TMDR.MD[3:0] = 0010	MTU6.TIORH.IOA[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	Except MTU6.TCR.CCLR[2:0] = 001 MTU6.TIORH.IOA[1:0] = 01/10/11		
	MTIOC6B	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TIORH.IOB[3] = 0 MTU6.TIORH.IOB[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	Except MTU6.TCR.CCLR[2:0] = 010 MTU6.TIORH.IOB[1:0] = 01/10/11		
	MTIOC6C	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TMDR.BFA = 0 MTU6.TIORL.IOC[3] = 0 MTU6.TIORL.IOC[1:0] = 01/10/11		
		PWM mode 1	MTU6.TMDR.MD[3:0] = 0010	MTU6.TMDR.BFA = 0 MTU6.TMDR.BFB = 0 MTU6.TIORL.IOC[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	MTU6.TMDR.BFA = 0 Except MTU6.TCR.CCLR[2:0] = 101 MTU6.TIORL.IOC[1:0] = 01/10/11		
	MTIOC6D	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TMDR.BFB = 0 MTU6.TIORL.IOD[3] = 0 MTU6.TIORL.IOD[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	MTU6.TMDR.BFB = 0 Except MTU6.TCR.CCLR[2:0] = 110 MTU6.TIORL.IOD[1:0] = 01/10/11		
	MTU7	MTIOC7A	Normal operation	MTU7.TMDR.MD[3:0] = 0000	MTU7.TIOR.IOA[3] = 0 MTU7.TIOR.IOA[1:0] = 01/10/11	
			PWM mode 1	MTU7.TMDR.MD[3:0] = 0010	MTU7.TIOR.IOA[1:0] = 01/10/11	
PWM mode 2			MTU7.TMDR.MD[3:0] = 0011	Except MTU7.TCR.CCLR[1:0] = 01 MTU7.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 1			MTU7.TMDR.MD[3:0] = 0100	MTU7.TIOR.IOA[3] = 0 MTU7.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 2			MTU7.TMDR.MD[3:0] = 0101			
Phase count mode 3			MTU7.TMDR.MD[3:0] = 0110			
Phase count mode 4			MTU7.TMDR.MD[3:0] = 0111			
MTIOC7B		Normal operation	MTU7.TMDR.MD[3:0] = 0000	MTU7.TIOR.IOB[3] = 0 MTU7.TIOR.IOB[1:0] = 01/10/11		
		PWM mode 2	MTU7.TMDR.MD[3:0] = 0011	Except MTU7.TCR.CCLR[1:0] = 10 MTU7.TIOR.IOB[1:0] = 01/10/11		
		Phase count mode 1	MTU7.TMDR.MD[3:0] = 0100	MTU7.TIOR.IOB[3] = 0 MTU7.TIOR.IOB[1:0] = 01/10/11		
		Phase count mode 2	MTU7.TMDR.MD[3:0] = 0101			
		Phase count mode 3	MTU7.TMDR.MD[3:0] = 0110			
		Phase count mode 4	MTU7.TMDR.MD[3:0] = 0111			
		MTU8	MTIOC8A	Normal operation	MTU8.TMDR.MD[3:0] = 0000	MTU8.TIOR.IOA[3] = 0 MTU8.TIOR.IOA[1:0] = 01/10/11
				PWM mode 1	MTU8.TMDR.MD[3:0] = 0010	MTU8.TIOR.IOA[1:0] = 01/10/11
PWM mode 2	MTU8.TMDR.MD[3:0] = 0011			Except MTU8.TCR.CCLR[1:0] = 01 MTU8.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 1	MTU8.TMDR.MD[3:0] = 0100			MTU8.TIOR.IOA[3] = 0 MTU8.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 2	MTU8.TMDR.MD[3:0] = 0101					
Phase count mode 3	MTU8.TMDR.MD[3:0] = 0110					
Phase count mode 4	MTU8.TMDR.MD[3:0] = 0111					

Table 17.22 Settings to Enable Output on the Various MTU Pins (5 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins
MTU8	MTIOC8B	Normal operation	MTU8.TMDR.MD[3:0] = 0000	MTU8.TIOR.IOB[3] = 0 MTU8.TIOR.IOB[1:0] = 01/10/11
		PWM mode 2	MTU8.TMDR.MD[3:0] = 0011	Except MTU8.TCR.CCLR[1:0] = 10 MTU8.TIOR.IOB[1:0] = 01/10/11
		Phase count mode 1	MTU8.TMDR.MD[3:0] = 0100	MTU8.TIOR.IOB[3] = 0
		Phase count mode 2	MTU8.TMDR.MD[3:0] = 0101	MTU8.TIOR.IOB[1:0] = 01/10/11
		Phase count mode 3	MTU8.TMDR.MD[3:0] = 0110	
		Phase count mode 4	MTU8.TMDR.MD[3:0] = 0111	
MTU9	MTIOC9A	Normal operation	MTU9.TMDR.MD[3:0] = 0000	MTU9.TIORH.IOA[3] = 0 MTU9.TIORH.IOA[1:0] = 01/10/11
		PWM mode 1	MTU9.TMDR.MD[3:0] = 0010	MTU9.TIORH.IOA[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU9.TMDR.MD[3:0] = 1000	MTUB.TOER1.PSYE = 1
		Complementary PWM mode 1	MTU9.TMDR.MD[3:0] = 1101	
		Complementary PWM mode 2	MTU9.TMDR.MD[3:0] = 1110	
		Complementary PWM mode 3	MTU9.TMDR.MD[3:0] = 1111	
	MTIOC9B	Normal operation	MTU9.TMDR.MD[3:0] = 0000	MTUB.TOER.OE3B = 1 MTU9.TIORH.IOB[3] = 0 MTU9.TIORH.IOB[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU9.TMDR.MD[3:0] = 1000	MTUB.TOER.OE3B = 1
		Complementary PWM mode 1	MTU9.TMDR.MD[3:0] = 1101	
		Complementary PWM mode 2	MTU9.TMDR.MD[3:0] = 1110	
		Complementary PWM mode 3	MTU9.TMDR.MD[3:0] = 1111	
		MTIOC9C	Normal operation	MTU9.TMDR.MD[3:0] = 0000
	PWM mode 1		MTU9.TMDR.MD[3:0] = 0010	MTU9.TMDR.BFA = 0 MTU9.TMDR.BFB = 0 MTU9.TIORL.IOC[1:0] = 01/10/11
	MTIOC9D	Normal operation	MTU9.TMDR.MD[3:0] = 0000	MTUB.TOER.OE3D = 1 MTU9.TMDR.BFB = 0 MTU9.TIORL.IOD[3] = 0 MTU9.TIORL.IOD[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU9.TMDR.MD[3:0] = 1000	MTUB.TOER.OE3D = 1
Complementary PWM mode 1		MTU9.TMDR.MD[3:0] = 1101		
Complementary PWM mode 2		MTU9.TMDR.MD[3:0] = 1110		
Complementary PWM mode 3		MTU9.TMDR.MD[3:0] = 1111		
MTU10		MTIOC10A	Normal operation	MTU10.TMDR.MD[3:0] = 0000
	PWM mode 1		MTU10.TMDR.MD[3:0] = 0010	MTUB.TOER.OE4A = 1 MTU10.TIORH.IOA[1:0] = 01/10/11
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1000	MTUB.TOER.OE4A = 1
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1101	
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1110	
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1111	
	Reset-synchronized PWM mode (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4A = 1
	Complementary PWM mode 1 (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000	
	Complementary PWM mode 2 (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000	
	Complementary PWM mode 3 (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000	

Table 17.22 Settings to Enable Output on the Various MTU Pins (6 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins
MTU10	MTIOC10B	Normal operation	MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4B = 1 MTU10.TIORH.IOB[3] = 0 MTU10.TIORH.IOB[1:0] = 01/10/11
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1000	MTUB.TOER.OE4B = 1
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4B = 1
		Complementary PWM mode 1 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000	
		MTIOC10C		Normal operation
PWM mode 1	MTU10.TMDR.MD[3:0] = 0010			MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0 MTU10.TMDR.BFB = 0 MTU10.TIORL.IOC[1:0] = 01/10/11
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1000			MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1101			
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1110			
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1111			
Reset-synchronized PWM mode (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000			MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0
Complementary PWM mode 1 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000			
Complementary PWM mode 2 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000			
Complementary PWM mode 3 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000			
MTIOC10D		Normal operation	MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4D = 1 MTU10.TMDR.BFB = 0 MTU10.TIORL.IOD[3] = 0 MTU10.TIORL.IOD[1:0] = 01/10/11
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1000	MTUB.TOER.OE4D = 1
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4D = 1
		Complementary PWM mode 1 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000	

17.2.5 Treatment of Unused Pins

The treatment of unused pins is listed in Table 17.23.

Table 17.23 Treatment of Unused Pins (145-Pin TFLGA/144-Pin LQFP)

Pin Name	Treatment
EMLE	Connect this pin to Vss via a pull-down resistor.
MD1, MD0	(Always used as a mode pin)
MDE	(Always used as mode pins)
RES#	Connect this pin to Vcc via a pull-up resistor.
USB0_DP	Leave these pins open.
USB0_DM	
BSCANP	Connect this pin to Vss via a pull-down resistor.
P35/NMI	Connect this pin to Vcc via a pull-up resistor.
EXTAL	(Always used as a clock pin)
XTAL	Leave this pin open.
XCIN	Connect this pin to Vcc via a pull-up resistor or to Vss via a pull-down resistor.
XCOUT	Leave this pin open.
WDTOVF#	Leave this pin open.
Ports 0 to 9, and A to E	<ul style="list-style-type: none"> ▪ Connect these pins to Vcc via a pull-up resistor or to Vss via a pull-down resistor ▪ These pins can be left while PORTn.ICR is in the initial state (the input buffer disabled)*.
VREFH	Connect this pin to AVcc
VREFL	Connect this pin to AVSS

Note: * Do not change the initial value of PORTn.ICR. Changing the initial value may generate shoot-through current.

17.3 I/O Ports [for 100-Pin LQFP]

The RX62N/RX621 Group (100-pin LQFP) has 11 I/O ports (ports 0 to 5 and A to E), which handle 74 I/O pins.

17.3.1 Overview

Table 17.24 gives the specifications of the I/O ports and Table 17.25 lists I/O ports and pin functions.

Table 17.24 Specifications of I/O Ports 100-Pin LQFP)

Item	Description
I/O pins	72
Input pins	2
Number of ports	11 (0 to 5, A to E)
Built-in input pull-up resistor	Ports A, B, C, D, E
Open drain outputs	Ports 0, 1, 2, 3 (P30 to P34), C
5-V tolerance	Port 0 (P07), port 1 (P12, P13, P16), port 2 (P20, P21), port 3 (P33)
Schmitt trigger input pins	All port inputs, CAN inputs, USB inputs, IRQ inputs, MTU inputs, POE inputs, TMR inputs, RIIC inputs, SCI inputs, and A/D trigger inputs
Others	<ul style="list-style-type: none"> Each pin is capable of driving a capacitive load of 30 pF in the case of a TTL load. When configured as an output, a pin is capable of driving a Darlington transistor.

Table 17.25 Port Functions (100-Pin LQFP) (1 / 5)

Port	Description	Function				CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
		Bit	I/O	Input	Output				
Port 0	General I/O port pins, interrupt inputs, A/D converter inputs, and D/A converter outputs	5	P05	IRQ13-A	DA1	—	All input functions	—	√
		7	P07	ADTRG0#-A/ IRQ15-A			All input functions	—	√
Port 1	General I/O port pins, USB I/O signals, MTU I/O signals, TMR I/O signals, interrupt inputs, SCI I/O signals, RIIC I/O signals, PPG I/O signals, and A/D converter inputs	2	P12/SCL0	TMC11/ RxD2-A/IRQ2-B		—	All input functions	√	√
		3	P13/ MTIOC0B/ SDA0	ADTRG1#/ IRQ3-B	PO13/TMO3 /TxD2-A		All input functions	√	√
		4	P14/ MTIOC3A	USB0_OVRCURA/ TMRI2/IRQ4-B	PO15/ USB0_DPUPE-B		All input functions	√	√
		6	P16/ MTIOC3C	USB0_VBUS/ USB0_OVRCURB/ IRQ6-B	TMO2/PO14/ USB0_VBUSE N-B		All input functions	√	√

Table 17.25 Port Functions (100-Pin LQFP) (2 / 5)

Port	Description	Bit	Function				CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
			I/O	Input	Output					
Port 2	General I/O port pins, bus control I/O signals, USB I/O signals, RSPI I/O signals, MTU I/O signals, PPG outputs, TMR I/O signals, SCI I/O signals, A/D converter inputs, and on-chip emulator I/O signals	0	P20/ MTIOC1A/	USB0_ID/ TMR10	PO0/TxD0	—	All input functions	—	√	
		1	P21/ MTIOC1B	TMCI0/RxD0	USB0_EXICEN/ PO1	—	All input functions	—		
		2	P22/ MTIOC3B/ SCK0	MTCLKC-A	USB0_DRPD/ PO2/ TMO0	—	All input functions	—		
		3	P23/ MTIOC3D	MTCLKD-A	USB0_DPUPE- A/ TxD3/PO3	—	All input functions	—		
		4	P24/ MTIOC4A/ SCK3	MTCLKA-A/TMR11	CS4#/ USB0_VBUSE N-A/ PO4	—	All input functions	—		
		5	P25/ MTIOC4C	MTCLKB-A/ ADTRG0#-B/ RxD3	CS5#/ USB0_DPRPD/ PO5	—	All input functions	—		
		6	P26/MOSIB- A/ MTIOC2A		CS6#/PO6/ TMO1/TxD1/ TDO	MOSIB -A	P26, MTIOC2A	—		
		7	P27/ RSPCKB-A/ MTIOC2B/ SCK1	TCK	CS7#/ PO7	RSPCK B-A, TCK	P27, MTIOC2B, SCK1	—		
Port 3	General I/O port pins, CAN I/O signals, RSPI I/O signals, MTU I/O signals, TMR inputs, SCI I/O signals, interrupt inputs, PPG outputs, RTC outputs and on-chip emulator I/O signals	0	P30/MISOB- A/ MTIOC4B-A	TMRI3/RxD1/ IRQ0/TDI	PO8	MISOB -A	P30, MTIOC4B-A TMRI3, RxD1, IRQ0, TDI	—	√	
		1	P31/SSLB0- A/ MTIOC4D-A	TMCI2/IRQ1/ TMS	PO9	SSLB0- A	P31, MTIOC4D-A TMCI2, IRQ1, TMS	—	√	
		2	P32/ MTIOC0C	IRQ2-A	CTX0/TxD6/ PO10/RTCOU	—	All input functions	—	√	
		3	P33/ MTIOC0D	CRX0/RxD6/ IRQ3-A	PO11	—	All input functions	—	√	
		4	P34/ MTIOC0A/ SCK6	TMCI3/IRQ4-A /TRST#	PO12	—	All input functions	—	√	
		5		P35/NMI	—	All input functions	—	—		
Port 4	General I/O port pins, interrupt inputs, and A/D converter inputs	0	P40	AN0/IRQ8	—	—	P40, IRQ8	—	—	
		1	P41	AN1/IRQ9	—	—	P41, IRQ9	—	—	
		2	P42	AN2/IRQ10	—	—	P42, IRQ10	—	—	
		3	P43	AN3/IRQ11	—	—	P43, IRQ11	—	—	
		4	P44	AN4/IRQ12	—	—	P44, IRQ12	—	—	
		5	P45	AN5/IRQ13-B	—	—	P45, IRQ13-B	—	—	
		6	P46	AN6/IRQ14	—	—	P46, IRQ14	—	—	
		7	P47	AN7/IRQ15-B	—	—	P47, IRQ15-B	—	—	

Table 17.25 Port Functions (100-Pin LQFP) (3 / 5)

Port	Description	Bit	Function				CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
			I/O	Input	Output					
Port 5	General I/O port pins, external bus clock outputs, bus control I/O signals, RSPI outputs, MTU I/O signals, SCI I/O signals, and trace outputs	0	P50		WR0#/WR#/SSLB1-A/TxD2-B	—	All input functions	—	—	
		1	P51/SCK2	WAIT#-D	WR1#/BC1#/SSLB2-A	WAIT#-D	P51, SCK2			
		2	P52	RxD2-B	RD#/SSLB3-A	—	All input functions			
		3		P53	BCLK	—	All input functions			
		4	P54/MTIOC4B-B			—	All input functions			
		5	P55/MTIOC4D-B	WAIT#-B		WAIT#-B	P55, MTIOC4D-B			
Port A	General I/O port pins, address outputs, bus control I/O signals, RSPI I/O signals, MTU I/O signals, PPG outputs, and Ether I/O signals	0	PA0/MTIOC6A		A0/BC0#/SSLA1-B/PO16	—	All input functions	√	—	
		1	PA1/MTIOC6B		A1/SSLA2-B/PO17		All input functions			
		2	PA2/MTIOC6C		A2/SSLA3-B/PO18		All input functions			
		3	PA3/MTIOC6D/ET_MDIO		A3/PO19	ET_MDIO	PA3, MTIOC6D			
		4	PA4/SSLA0-B/MTIOC7A		A4/PO20/ET_MDC	SSLA0-B	PA4, MTIOC7A			
		5	PA5/RSPCKA-B/MTIOC7B	ET_LINKSTA	A5/PO21	RSPCKA-B, ET_LINKSTA	PA5, MTIOC7B			
		6	PA6/MOSIA-B/MTIOC8A		A6/PO22/ET_EXOUT	MOSIA-B	PA6, MTIOC8A			
		7	PA7/MISOA-B/MTIOC8B		A7/PO23/ET_WOL	MISOA-B	PA7, MTIOC8B			

Table 17.25 Port Functions (100-Pin LQFP) (4 / 5)

Port	Description	Bit	Function				CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
			I/O	Input	Output					
Port B	General I/O port pins, address outputs, MTU I/O signals, PPG outputs, and Ether I/O signals	0	PB0/ MTIOC9A	ET_ERXD1/ RMII_RXD1	A8/PO24	ET_ER XD1, RMII_R XD1	PB0, MTIOC9A	√	—	
		1	PB1/ MTIOC9C	ET_ERXD0/ RMII_RXD0	A9/PO25	ET_ER XD0, RMII_R XD0	PB1, MTIOC9C			
		2	PB2/ MTIOC9B	ET_RX_CLK/ REF50CK/ MTCLKG-B	A10/PO26	ET_RX _CLK, REF50 CK	PB2, MTIOC9B, MTCLKG-B			
		3	PB3/ MTIOC9D	ET_RX_ER/ RMII_RX_ER/ MTCLKH-B	A11/PO27	ET_RX _ER, RMII_R X_ER	PB3, MTIOC9D, MTCLKH-B			
		4	PB4/ MTIOC10A	MTCLKE-B	A12/PO28/ ET_TX_EN/ RMII_TXD_EN	—	All input functions			
		5	PB5/ MTIOC10C	MTCLKF-B	A13/PO29/ ET_ETXD0/ RMII_TXD0	—	All input functions			
		6	PB6/ MTIOC10B		A14/PO30/ ET_ETXD1/ RMII_TXD1	—	All input functions			
		7	PB7/ MTIOC10D	ET_CR_S/ RMII_CR_S_DV	A15/PO31	ET_CR S, RMII_C RS_DV	PB7, MTIOC10D			
Port C	General I/O port pins, address outputs, bus control outputs, RSPI I/O signals, Ether I/O signals, MTU inputs, and SCI I/O signals	0	PC0	ET_ERXD3/ MTCLKG-A	A16-A/SSLA1-A	ET_ER XD3	PC0, MTCLKG-A	√	√	
		1	PC1/SCK5	ET_ERXD2/ MTCLKH-A	A17-A/SSLA2-A	ET_ER XD2	PC1, SCK5, MTCLKH-A			
		2	PC2	ET_RX_DV/ MTCLKE-A/RxD5	A18-A/SSLA3-A	ET_RX _DV	PC2, MTCLKE-A, RxD5			
		3	PC3	ET_TX_ER/ MTCLKF-A	A19-A/TxD5	ET_TX _ER	PC3, MTCLKF-A			
		4	PC4/SSLA0-A	MTCLKC-B/ ET_TX_CLK	A20-A/CS3#	SSLA0-A, ET_TX _CLK	PC4, MTCLKC-B			
		5	PC5/ RSPCKA-A	WAIT#-C/ MTIC11W-A/ MTCLKD-B	A21-A/CS2#/ ET_ETXD2	RSPCK A-A, WAIT#- C	PC5, MTIC11W-A, MTCLKD-B			
		6	PC6/MOSIA-A	MTIC11V-A/ MTCLKA-B	A22-A/CS1#/ ET_ETXD3	MOSIA -A	PC6, MTIC11V-A MTCLKA-B			
		7	PC7/MISOA-A	ET_COL/ MTIC11U-A/ MTCLKB-B	A23-A/CS0#	MISOA -A, ET_CO L	PC7, MTIC11U-A MTCLKB-B			

Table 17.25 Port Functions (100-Pin LQFP) (5 / 5)

Port	Description	Bit	Function				CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
			I/O	Input	Output					
Port D	General I/O port pins, bidirectional data-bus lines, MTU inputs, and POE inputs	0	PD0/D0	POE7#		D0	PD0, POE7#	√	—	
		1	PD1/D1	POE6#		D1	PD1, POE6#			
		2	PD2/D2	MTIC11W-B/ POE5#		D2	PD2, MTIC11W-B, POE5#			
		3	PD3/D3	MTIC11V-B/ POE4#		D3	PD3, MTIC11V-B, POE4#			
		4	PD4/D4	MTIC11U-B/ POE3#		D4	PD4, MTIC11U-B, POE3#			
		5	PD5/D5	MTIC5W/ POE2#		D5	PD5, MTIC5W, POE2#			
		6	PD6/D6	MTIC5V/ POE1#		D6	PD6, MTIC5V, POE1#			
		7	PD7/D7	MTIC5U/ POE0#		D7	PD7, MTIC5U, POE0#			
Port E	General I/O port pins, bidirectional data-bus lines, interrupt inputs, RSPI I/O signals, and POE inputs	0	PE0/D8		SSLB1-B	D8	PE0	√	—	
		1	PE1/D9		SSLB2-B	D9	PE1			
		2	PE2/D10	POE9#	SSLB3-B	D10	PE2, POE9#			
		3	PE3/D11	POE8#		D11	PE3, POE8#			
		4	PE4/D12/ SSLB0-B			D12, SSLB0-B	PE4			
		5	PE5/D13/ RSPCKB-B	IRQ5		D13, RSPCK B-B	PE5, IRQ5			
		6	PE6/D14/ MOSIB-B	IRQ6-A		D14, MOSIB -B	PE6, IRQ6-A			
		7	PE7/D15/ MISOB-B	IRQ7		D15, MISOB -B	PE7, IRQ7			

17.3.2 Register Descriptions

Table 17.26 lists registers of I/O ports, and Table 17.27 lists valid bits in each register.

Table 17.26 Registers of I/O Ports (100-Pin LQFP) (1 / 2)

Port Symbol	Register Name	Register Symbol	Value after Reset	Address	Access Size
PORT0	Data direction register	DDR	00h	0008 C000h	8
	Data register	DR	00h	0008 C020h	8
	Port register	PORT	Undefined	0008 C040h	8
	Input buffer control register	ICR	00h	0008 C060h	8
	Open drain control register	ODR	00h	0008 C080h	8
PORT1	Data direction register	DDR	00h	0008 C001h	8
	Data register	DR	00h	0008 C021h	8
	Port register	PORT	Undefined	0008 C041h	8
	Input buffer control register	ICR	00h	0008 C061h	8
	Open drain control register	ODR	00h	0008 C081h	8
PORT2	Data direction register	DDR	00h	0008 C002h	8
	Data register	DR	00h	0008 C022h	8
	Port register	PORT	Undefined	0008 C042h	8
	Input buffer control register	ICR	00h	0008 C062h	8
	Open drain control register	ODR	00h	0008 C082h	8
PORT3	Data direction register	DDR	00h	0008 C003h	8
	Data register	DR	00h	0008 C023h	8
	Port register	PORT	Undefined	0008 C043h	8
	Input buffer control register	ICR	00h	0008 C063h	8
	Open drain control register	ODR	00h	0008 C083h	8
PORT4	Data direction register	DDR	00h	0008 C004h	8
	Data register	DR	00h	0008 C024h	8
	Port register	PORT	Undefined	0008 C044h	8
	Input buffer control register	ICR	00h	0008 C064h	8
PORT5	Data direction register	DDR	00h	0008 C005h	8
	Data register	DR	00h	0008 C025h	8
	Port register	PORT	Undefined	0008 C045h	8
	Input buffer control register	ICR	00h	0008 C065h	8
PORTA	Data direction register	DDR	00h	0008 C00Ah	8
	Data register	DR	00h	0008 C02Ah	8
	Port register	PORT	Undefined	0008 C04Ah	8
	Input buffer control register	ICR	00h	0008 C06Ah	8
	Pull-up resistor control register	PCR	00h	0008 C0CAh	8
PORTB	Data direction register	DDR	00h	0008 C00Bh	8
	Data register	DR	00h	0008 C02Bh	8
	Port register	PORT	Undefined	0008 C04Bh	8
	Input buffer control register	ICR	00h	0008 C06Bh	8
	Pull-up resistor control register	PCR	00h	0008 C0CBh	8

Table 17.26 Registers of I/O Ports (100-Pin LQFP) (2 / 2)

Port Symbol	Register Name	Register Symbol	Value after Reset	Address	Access Size
PORTC	Data direction register	DDR	00h	0008 C00Ch	8
	Data register	DR	00h	0008 C02Ch	8
	Port register	PORT	Undefined	0008 C04Ch	8
	Input buffer control register	ICR	00h	0008 C06Ch	8
	Open drain control register	ODR	00h	0008 C08Ch	8
	Pull-up resistor control register	PCR	00h	0008 C0CCh	8
PORTD	Data direction register	DDR	00h	0008 C00Dh	8
	Data register	DR	00h	0008 C02Dh	8
	Port register	PORT	Undefined	0008 C04Dh	8
	Input buffer control register	ICR	00h	0008 C06Dh	8
	Pull-up resistor control register	PCR	00h	0008 C0CDh	8
PORTE	Data direction register	DDR	00h	0008 C00Eh	8
	Data register	DR	00h	0008 C02Eh	8
	Port register	PORT	Undefined	0008 C04Eh	8
	Input buffer control register	ICR	00h	0008 C06Eh	8
	Pull-up resistor control register	PCR	00h	0008 C0CEh	8
IOPORT	Port function control register 0	PF0CSE	00h	0008 C100h	8
	Port function control register 3	PF3BUS	00h	0008 C103h	8
	Port function control register 4	PF4BUS	00h	0008 C104h	8
	Port function control register 5	PF5BUS	00h	0008 C105h	8
	Port function control register 6	PF6BUS	00h	0008 C106h	8
	Port function control register 8	PF8IRQ	00h	0008 C108h	8
	Port function control register 9	PF9IRQ	00h	0008 C109h	8
	Port function control register A	PFAADC	00h	0008 C10Ah	8
	Port function control register C	PFCMTU	00h	0008 C10Ch	8
	Port function control register D	PFDMTU	00h	0008 C10Dh	8
	Port function control register E	PFENET	00h	0008 C10Eh	8
	Port function control register F	PFSCI	00h	0008 C10Fh	8
	Port function control register G	PFGSPI	00h	0008 C110h	8
	Port function control register H	PFHSPI	00h	0008 C111h	8
	Port function control register J	PFJCAN	00h	0008 C113h	8
	Port function control register K	PFKUSB	00h	0008 C114h	8
Port function control register M	PFMPOE	00h	0008 C116h	8	
Port function control register N	PFNPOE	00h	0008 C117h	8	

Table 17.27 Valid Bits in Each Register (100-Pin LQFP) (1 / 2)

Register Symbol	b7	b6	b5	b4	b3	b2	b1	b0
PORT0.DDR	√	x	√	x	x	x	x	x
PORT1.DDR	x	√	x	√	√	√	x	x
PORT2.DDR	√	√	√	√	√	√	√	√
PORT3.DDR	x	x	x	√	√	√	√	√
PORT4.DDR	√	√	√	√	√	√	√	√
PORT5.DDR	x	x	√	√	√	√	√	√
PORTA.DDR	√	√	√	√	√	√	√	√
PORTB.DDR	√	√	√	√	√	√	√	√
PORTC.DDR	√	√	√	√	√	√	√	√
PORTD.DDR	√	√	√	√	√	√	√	√
PORTE.DDR	√	√	√	√	√	√	√	√
PORT0.DR	√	x	√	x	x	x	x	x
PORT1.DR	x	√	x	√	√	√	x	x
PORT2.DR	√	√	√	√	√	√	√	√
PORT3.DR	x	x	x	√	√	√	√	√
PORT4.DR	√	√	√	√	√	√	√	√
PORT5.DR	x	x	√	√	x	√	√	√
PORTA.DR	√	√	√	√	√	√	√	√
PORTB.DR	√	√	√	√	√	√	√	√
PORTC.DR	√	√	√	√	√	√	√	√
PORTD.DR	√	√	√	√	√	√	√	√
PORTE.DR	√	√	√	√	√	√	√	√
PORT0.PORT	√	x	√	x	x	x	x	x
PORT1.PORT	x	√	x	√	√	√	x	x
PORT2.PORT	√	√	√	√	√	√	√	√
PORT3.PORT	x	x	√	√	√	√	√	√
PORT4.PORT	√	√	√	√	√	√	√	√
PORT5.PORT	x	x	√	√	√	√	√	√
PORTA.PORT	√	√	√	√	√	√	√	√
PORTB.PORT	√	√	√	√	√	√	√	√
PORTC.PORT	√	√	√	√	√	√	√	√
PORTD.PORT	√	√	√	√	√	√	√	√
PORTE.PORT	√	√	√	√	√	√	√	√
PORT0.ICR	√	x	√	x	x	x	x	x
PORT1.ICR	x	√	x	√	√	√	x	x
PORT2.ICR	√	√	√	√	√	√	√	√
PORT3.ICR	x	x	x	√	√	√	√	√
PORT4.ICR	√	√	√	√	√	√	√	√
PORT5.ICR	x	x	√	√	√	√	√	√
PORTA.ICR	√	√	√	√	√	√	√	√
PORTB.ICR	√	√	√	√	√	√	√	√
PORTC.ICR	√	√	√	√	√	√	√	√
PORTD.ICR	√	√	√	√	√	√	√	√
PORTE.ICR	√	√	√	√	√	√	√	√

Table 17.27 Valid Bits in Each Register (100-Pin LQFP) (2 / 2)

Register Symbol	b7	b6	b5	b4	b3	b2	b1	b0
PORT0.ODR	√	x	√	x	x	x	x	x
PORT1.ODR	x	√	x	√	√	√	x	x
PORT2.ODR	√	√	√	√	√	√	√	√
PORT3.ODR	x	x	x	√	√	√	√	√
PORTC.ODR	√	√	√	√	√	√	√	√
PORTA.PCR	√	√	√	√	√	√	√	√
PORTB.PCR	√	√	√	√	√	√	√	√
PORTC.PCR	√	√	√	√	√	√	√	√
PORTD.PCR	√	√	√	√	√	√	√	√
PORTE.PCR	√	√	√	√	√	√	√	√
IOPORT.PF0CSE	√	√	√	√	√	√	√	√
IOPORT.PF3BUS	√	√	√	√	√	√	√	√
IOPORT.PF4BUS	√	√	√	√	√	√	√	√
IOPORT.PF5BUS	x	√	x	√	x	x	x	x
IOPORT.PF6BUS	x	x	x	x	x	x	√	√
IOPORT.PF8IRQ	√	x	√	x	x	x	x	x
IOPORT.PF9IRQ	x	√	x	√	√	√	x	x
IOPORT.PFAADC	x	x	x	x	x	x	x	√
IOPORT.PFCMTU	√	x	√	x	x	x	x	x
IOPORT.PFDMTU	√	√	x	x	x	x	x	x
IOPORT.PFENET	√	x	x	√	√	√	√	√
IOPORT.PFFSCI	x	x	x	x	x	√	x	x
IOPORT.PFGSPI	√	√	√	√	√	√	√	√
IOPORT.PFHSPI	√	√	√	√	√	√	√	√
IOPORT.PFJCAN	x	x	x	x	x	x	x	√
IOPORT.PFKUSB	x	x	x	√	√	√	√	√
IOPORT.PFMPOE	√	√	√	√	√	√	√	√
IOPORT.PFNPOE	x	x	x	x	x	x	√	√

17.3.2.1 Data Direction Register (DDR)

Addresses: PORT0.DDR 0008 C000h, PORT1.DDR 0008 C001h, PORT2.DDR 0008 C002h,
PORT3.DDR 0008 C003h, PORT4.DDR 0008 C004h, PORT5.DDR 0008 C005h,
PORTA.DDR 0008 C00Ah, PORTB.DDR 0008 C00Bh, PORTC.DDR 0008 C00Ch,
PORTD.DDR 0008 C00Dh, PORTE.DDR 0008 C00Eh

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

Note: Bits 6 and 4 to 0 are reserved in PORT0.DDR.
Bits 7, 5, 1, and 0 are reserved in PORT1.DDR.
The lower five bits are valid and the upper three bits are reserved in PORT3.DDR.
The lower six bits are valid and the upper two are reserved in PORT5.DDR.
The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 I/O Select	0: An input pin	R/W
b1	B1	Pn1 I/O Select	1: An output pin	R/W
b2	B2	Pn2 I/O Select		R/W
b3	B3	Pn3 I/O Select		R/W
b4	B4	Pn4 I/O Select		R/W
b5	B5	Pn5 I/O Select		R/W
b6	B6	Pn6 I/O Select		R/W
b7	B7	Pn7 I/O Select		R/W

[Legend] n = 0 to 5, A to E

Each DDR is used to select the input or output direction for individual pins of the corresponding port that have been configured to function as general I/O pins.

Each bit of a PORTn.DDR (n = 0 to 5, A to E) corresponds to a pin of PORTn, and the settings can change from bit to bit.

The PORT5.DDR.B3 bit selects P53 input or BCLK output. Setting the PORT5.DDR.B3 bit to 1 specifies output of the BCLK signal on the pin that would otherwise be P53. Operation as a general output is not selectable for this pin.

17.3.2.2 Data Register (DR)

Addresses: PORT0.DR 0008 C020h, PORT1.DR 0008 C021h, PORT2.DR 0008 C022h,
 PORT3.DR 0008 C023h, PORT4.DR 0008 C024h, PORT5.DR 0008 C025h,
 PORTA.DR 0008 C02Ah, PORTB.DR 0008 C02Bh, PORTC.DR 0008 C02Ch,
 PORTD.DR 0008 C02Dh, PORTE.DR 0008 C02Eh

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

Note: Bits 6 and 4 to 0 are reserved in PORT0.DR.
 Bits 7, 5, 1, and 0 are reserved in PORT1.DR.
 The lower five bits are valid and the upper three bits are reserved in PORT3.DR.
 Bits 7, 6, and 3 are reserved in PORT5.DR.
 The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 Output Data Store	Output data are stored.	R/W
b1	B1	Pn1 Output Data Store		R/W
b2	B2	Pn2 Output Data Store		R/W
b3	B3	Pn3 Output Data Store		R/W
b4	B4	Pn4 Output Data Store		R/W
b5	B5	Pn5 Output Data Store		R/W
b6	B6	Pn6 Output Data Store		R/W
b7	B7	Pn7 Output Data Store		R/W

[Legend] n = 0 to 5, A to E

Each DR stores the output data from the individual pins of the corresponding port used as a general I/O port.

The output of the P53 pin is the BCLK signal and the value of the B3 bit in PORT5.DR does not affect the pin.

17.3.2.3 Port Register (PORT)

Addresses: PORT0.PORT 0008 C040h, PORT1.PORT 0008 C041h, PORT2.PORT 0008 C042h, PORT3.PORT 0009 C043h, PORT4.PORT 0008 C044h, PORT5.PORT 0008 C045h, PORTA.PORT 0008 C04Ah, PORTB.PORT 0008 C04Bh, PORTC.PORT 0008 C04Ch, PORTD.PORT 0008 C04Dh, PORTE.PORT 0008 C04Eh

b7	b6	b5	b4	b3	b2	b1	b0
B7	B6	B5	B4	B3	B2	B1	B0

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

Note: Bits 6 and 4 to 0 are reserved in PORT0.PORT.
 Bits 7, 5, 1, and 0 are reserved in PORT1.PORT.
 The lower six bits are valid and the upper two bits are reserved in PORT3.PORT.
 The lower six bits are valid and the upper two bits are reserved in PORT5.PORT.
 The reserved bits are read as 1 and cannot be modified.

Bit	Symbol	Bit Name	Description	R/W
b0	B0*	Pn0	Individual pin states of the corresponding port are reflected.	R
b1	B1*	Pn1		R
b2	B2*	Pn2		R
b3	B3*	Pn3		R
b4	B4*	Pn4		R
b5	B5*	Pn5		R
b6	B6*	Pn6		R
b7	B7*	Pn7		R

[Legend] n = 0 to 5, A to E

Note : * Before reading this register, set the corresponding bit in PORTn.ICR to 1. If this register is read with the corresponding bit in PORTn.ICR set to 0, the read value is undefined.

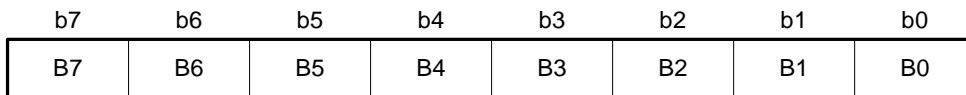
PORT reflects individual pin states of the corresponding port.

When a PORTn.PORT (n = 0 to 5, A to E) is read, the corresponding pin states are read out to here.

The NMI pin state is read out to the P35 bit.

17.3.2.4 Input Buffer Control Register (ICR)

Addresses: PORT0.ICR 0008 C060h, PORT1.ICR 0008 C061h, PORT2.ICR 0008 C062h, PORT3.ICR 0008 C063h, PORT4.ICR 0008 C064h, PORT5.ICR 0008 C065h, PORTA.ICR 0008 C06Ah, PORTB.ICR 0008 C06Bh, PORTC.ICR 0008 C06Ch, PORTD.ICR 0008 C06Dh, PORTE.ICR 0008 C06Eh



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

Note: Bits 6 and 4 to 0 are reserved in PORT0.ICR.
 Bits 7, 5, 1, and 0 are reserved in PORT1.ICR.
 The lower five bits are valid and the upper three bits are reserved in PORT3.ICR.
 The lower six bits are valid and the upper two bits are reserved in PORT5.ICR.
 The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0*	Pn0 Input Buffer Control	0: The input buffer for the corresponding pin is disabled. 1: The input buffer for the corresponding pin is enabled.	R/W
b1	B1*	Pn1 Input Buffer Control		R/W
b2	B2*	Pn2 Input Buffer Control		R/W
b3	B3*	Pn3 Input Buffer Control		R/W
b4	B4*	Pn4 Input Buffer Control		R/W
b5	B5*	Pn5 Input Buffer Control		R/W
b6	B6*	Pn6 Input Buffer Control		R/W
b7	B7*	Pn7 Input Buffer Control		R/W

[Legend] n = 0 to 5, A to E

Note : * For pins being used as input pins, set the corresponding bits to 1. Set the bits corresponding to pins that are not being used for their input functions or are being used as analog I/O pins to 0.

Each ICR controls the input buffers for the individual pins of the corresponding port.

Each bit of a PORTn.ICR (n = 0 to 5, A to E) corresponds to a pin of PORTn, and the settings can change from bit to bit.

When to be used as an input pin for the peripheral module, the input buffer for the corresponding pin should be enabled beforehand by setting the PORTn.ICR bit to 1. If this register is used as an input pin for the peripheral module while the PORTn.ICR bit is 0, the input signal to the peripheral module is fixed high.

Changes in the settings of a PORTn.ICR may generate edges internally, depending on the pin state. For this reason, change the settings of PORTn.ICR while the corresponding input pins are not in use. For example, in the case of IRQi (i = 0 to 15) inputs, change settings of the corresponding PORTn.ICR with interrupts disabled by clearing the IR flag in IRi (i = 64 to 79 ("i" shows an interrupt vector number)) of the interrupt control unit (ICU) to 0, and then enable the corresponding interrupts. If a change to a PORTn.ICR setting does generate an edge, negate the edge.

17.3.2.5 Open Drain Control Register (ODR)

Addresses: PORT0.ODR 0008 C080h, PORT1.ODR 0008 C081h, PORT2.ODR 0008 C082h, PORT3.ODR 0008 C083h, PORTC.ODR 0008 C08Ch

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

Note: Bits 6 and 4 to 0 are reserved in PORT0.ODR.
 Bits 7, 5, 1, and 0 are reserved in PORT1.ODR.
 The lower five bits are valid and the upper three bits are reserved in PORT3.ODR.
 The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 Output Type Select	0: CMOS output pin	R/W
b1	B1	Pn1 Output Type Select	1: NMOS open-drain output pin	R/W
b2	B2	Pn2 Output Type Select		R/W
b3	B3	Pn3 Output Type Select		R/W
b4	B4	Pn4 Output Type Select		R/W
b5	B5	Pn5 Output Type Select		R/W
b6	B6	Pn6 Output Type Select		R/W
b7	B7	Pn7 Output Type Select		R/W

[Legend] n = 0 to 3, C

Each ODR is used to select an output type for the individual pins.

17.3.2.6 Pull-Up Resistor Control Register (PCR)

Addresses: PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTC.PCR 0008 C0CCh,
PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh

	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	Pn0 Input Pull-Up Resistor Control	0: Input pull-up resistor is off. 1: Input pull-up resistor is on.	R/W
b1	B1	Pn1 Input Pull-Up Resistor Control		R/W
b2	B2	Pn2 Input Pull-Up Resistor Control		R/W
b3	B3	Pn3 Input Pull-Up Resistor Control		R/W
b4	B4	Pn4 Input Pull-Up Resistor Control		R/W
b5	B5	Pn5 Input Pull-Up Resistor Control		R/W
b6	B6	Pn6 Input Pull-Up Resistor Control		R/W
b7	B7	Pn7 Input Pull-Up Resistor Control		R/W

[Legend] n = A to E

Each PCR controls enabled/disabled of input pull-up resistor for individual pins of the corresponding port.

When in input pin state, for the pins corresponding to bits where the value in PORTn.PCR is 1, input pull-up resistor is turned on. Table 17.28 summarizes the input pull-up resistor states.

Table 17.28 Input Pull-Up Resistor States (100-Pin LQFP)

Port	Pin State	Reset or Hardware Standby Mode	In Other Operations
Port A	Port input	Disabled	
	Address output	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	Enabled/Disabled
Port B	Address output	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port C	Address output	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port D	Data I/O	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port E	Data I/O	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled

[Legend]

Disabled: Input pull-up resistor is always disabled.

Enabled/Disabled: Input pull-up resistor is enabled when the PORTm.PCR.Bj bit (m = 9, and A to E, j = 0 to 7) is set to 1, and disabled when the bit is cleared to 0.

17.3.2.7 Port Function Control Register 0 (PF0CSE)

Address: 0008 C100h

	b7	b6	b5	b4	b3	b2	b1	b0
	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CS0E	CS0 Output Enable	0: Designated as an I/O port pin.	R/W
b1	CS1E	CS1 Output Enable	1: Designated as the CSn# output pin (n = 0 to 7)	R/W
b2	CS2E	CS2 Output Enable		R/W
b3	CS3E	CS3 Output Enable		R/W
b4	CS4E	CS4 Output Enable		R/W
b5	CS5E	CS5 Output Enable		R/W
b6	CS6E	CS6 Output Enable		R/W
b7	CS7E	CS7 Output Enable		R/W

PF0CSE enables or disables CSn# output.

CSnE Bit (CSn Output Enable) (n = 0 to 7)

Each bit enables or disables the corresponding CSn# output.

To output a CSn signal, set the corresponding CSnE bit in PF0CSE to 1.

17.3.2.8 Port Function Control Register 3 (PF3BUS)

Address: 0008 C103h

	b7	b6	b5	b4	b3	b2	b1	b0
	A23E	A22E	A21E	A20E	A19E	A18E	A17E	A16E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	A16E	Address A16 Output Enable	0: A16 output is disabled. 1: A16 output is enabled.	R/W
b1	A17E	Address A17 Output Enable	0: A17 output is disabled. 1: A17 output is enabled.	R/W
b2	A18E	Address A18 Output Enable	0: A18 output is disabled. 1: A18 output is enabled.	R/W
b3	A19E	Address A19 Output Enable	0: A19 output is disabled. 1: A19 output is enabled.	R/W
b4	A20E	Address A20 Output Enable	0: A20 output is disabled. 1: A20 output is enabled.	R/W
b5	A21E	Address A21 Output Enable	0: A21 output is disabled. 1: A21 output is enabled.	R/W
b6	A22E	Address A22 Output Enable	0: A22 output is disabled. 1: A22 output is enabled.	R/W
b7	A23E	Address A23 Output Enable	0: A23 output is disabled. 1: A23 output is enabled.	R/W

PF3BUS enables or disables address outputs.

AnE Bit (Address An Output Enable) (n = 16 to 23)

Each bit enables or disables an address output (An).

17.3.2.9 Port Function Control Register 4 (PF4BUS)

Address: 0008 C104h

b7	b6	b5	b4	b3	b2	b1	b0
A15E	A14E	A13E	A12E	A11E	A10E	ADRLE[1:0]	
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ADRLE[1: 0]	Address Lower A9 to A0 Output Enable	^{b1 b0} 0 0: A9 to A0 output is disabled. 0 1: A9 to A4 output is disabled, A3 to A0 output is enabled 1 0: A9 to A8 output is disabled, A7 to A0 output is enabled 1 1: A9 to A0 output is enabled	R/W
b2	A10E	Address A10 Output Enable	0: A10 output is disabled. 1: A10 output is enabled.	R/W
b3	A11E	Address A11 Output Enable	0: A11 output is disabled. 1: A11 output is enabled.	R/W
b4	A12E	Address A12 Output Enable	0: A12 output is disabled. 1: A12 output is enabled.	R/W
b5	A13E	Address A13 Output Enable	0: A13 output is disabled. 1: A13 output is enabled.	R/W
b6	A14E	Address A14 Output Enable	0: A14 output is disabled. 1: A14 output is enabled.	R/W
b7	A15E	Address A15 Output Enable	0: A15 output is disabled. 1: A15 output is enabled.	R/W

PF4BUS enables or disables address outputs.

ADRLE[1: 0] Bits (Address Lower A9 to A0 Output Enable)

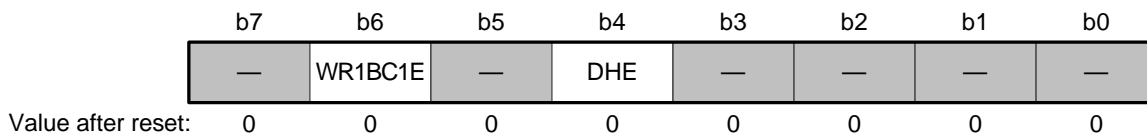
These bits enable or disable an address output (A9 to A0).

AnE Bit (Address An Output Enable) (n = 10 to 15)

Each bit enables or disables an address output (An).

17.3.2.10 Port Function Control Register 5 (PF5BUS)

Address: 0008 C105h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	DHE	D15-to-D8 Enable	0: PE7 to PE0 are designated as I/O port pins. 1: PE7 to PE0 are designated as D15 to D8 pins (function as part of the external data bus)	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b6	WR1BC1E	WR1#/BC1# Output Enable	0: P51 is designated as an I/O port pin. 1: P51 is designated as the WR1# or BC1# pin.	R/W
b7	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

DHE Bit (D15-to-D8 Enable)

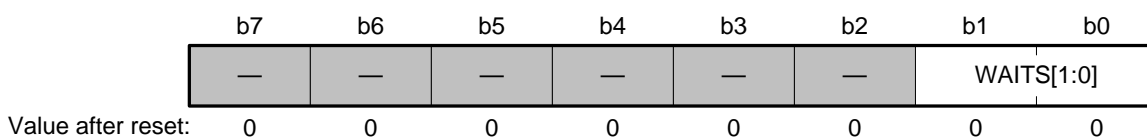
This bit enables or disables the input and output of data signals D15 to D8.

WR1BC1E Bit (WR1#/BC1# Output Enable)

This bit enables or disables WR1#/BC1# output.

17.3.2.11 Port Function Control Register 6 (PF6BUS)

Address: 0008 C106h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	WAITS[1:0]	WAIT Select	b1 b0 0 x: P55 is designated as the WAIT#-B input pin. 1 0: PC5 is designated as the WAIT#-C input pin. 1 1: P51 is designated as the WAIT#-D input pin.	R/W
b7 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

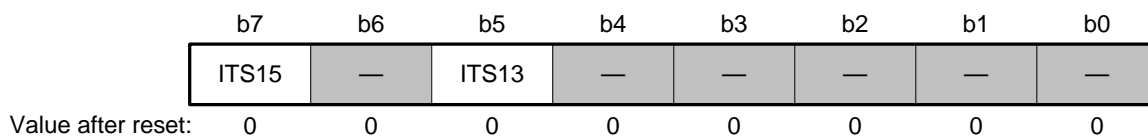
[Legend] x: Don't care

WAITS Bits (WAIT Select)

These bits select a pin for a WAIT# input.

17.3.2.12 Port Function Control Register 8 (PF8IRQ)

Address: 0008 C108h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b5	ITS13	IRQ13 Pin Select	0: P05 is designated as the IRQ13-A input pin. 1: P45 is designated as the IRQ13-B input pin.	R/W
b6	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b7	ITS15	IRQ15 Pin Select	0: P07 is designated as the IRQ15-A input pin. 1: P47 is designated as the IRQ15-B input pin.	R/W

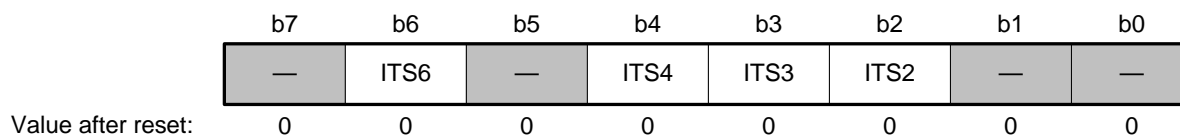
PF8IRQ is used to select pins for IRQ13 and IRQ15 inputs.

ITS_i Bit (IRQ_i Pin Select) (i = 13, 15)

Each bit selects a pin for an IRQ_i input.

17.3.2.13 Port Function Control Register 9 (PF9IRQ)

Address: 0008 C109h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b2	ITS2	IRQ2 Pin Select	0: P32 is designated as the IRQ2-A input pin. 1: P12 is designated as the IRQ2-B input pin.	R/W
b3	ITS3	IRQ3 Pin Select	0: P33 is designated as the IRQ3-A input pin. 1: P13 is designated as the IRQ3-B input pin.	R/W
b4	ITS4	IRQ4 Pin Select	0: P34 is designated as the IRQ4-A input pin. 1: P14 is designated as the IRQ4-B input pin.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b6	ITS6	IRQ6 Pin Select	0: PE6 is designated as the IRQ6-A input pin. 1: P16 is designated as the IRQ6-B input pin.	R/W
b7	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

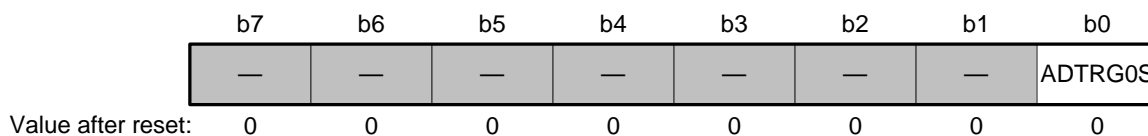
PF9IRQ is used to select pins for IRQ2 to IRQ4, IRQ6 inputs.

ITS_i Bit (IRQ_i Pin Select) (i = 2 to 4, 6)

Each bit selects a pin for an IRQ_i input.

17.3.2.14 Port Function Control Register A (PFAADC)

Address: 0008 C10Ah



Bit	Symbol	Bit Name	Description	R/W
b0	ADTRG0S	ADTRG0# Input Select	0: P07 is designated as the ADTRG0#-A input pin. 1: P25 is designated as the ADTRG0#-B input pin.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

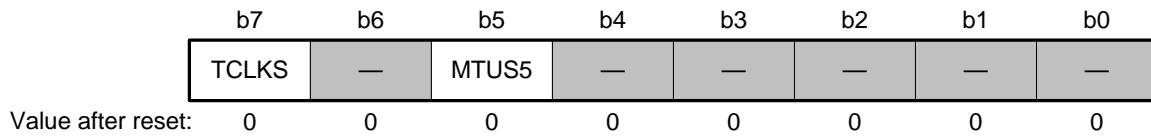
PFAADC is used to select a pin for ADTRG0# input.

ADTRG0S Bit (ADTRG0# Input Select)

This bit selects a pin for an ADTRG0# input.

17.3.2.15 Port Function Control Register C (PFCMTU)

Address:0008 C10Ch



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b5	MTUS5	MTU Pin Select 5	0: P30 is designated as the MTIOC4B-A pin. P31 is designated as the MTIOC4D-A pin. 1: P54 is designated as the MTIOC4B-B pin. P55 is designated as the MTIOC4D-B pin.	R/W
b6	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b7	TCLKS	MTCLK Pin Select	0: P24 is designated as the MTCLKA-A pin. P25 is designated as the MTCLKB-A pin. P22 is designated as the MTCLKC-A pin. P23 is designated as the MTCLKD-A pin. 1: PC6 is designated as the MTCLKA-B pin. PC7 is designated as the MTCLKB-B pin. PC4 is designated as the MTCLKC-B pin. PC5 is designated as the MTCLKD-B pin.	R/W

PFCMTU is used to select pins for MTU unit 0.

MTUS_j Bit (MTU Pin Select) (j = 5)

Each bit selects a pin for an MTU input/output.

TCLKS Bit (MTCLK Pin Select)

This bit selects a pin for a MTCLK input of the MTU.

17.3.2.16 Port Function Control Register D (PFDMTU)

Address: 0008 C10Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	TCLKS	MTUS6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b6	MTUS6	MTU Pin Select 6	0: PC7 is designated as the MTIC11U-A pin. PC6 is designated as the MTIC11V-A pin. PC5 is designated as the MTIC11W-A pin. 1: PD4 is designated as the MTIC11U-B pin. PD3 is designated as the MTIC11V-B pin. PD2 is designated as the MTIC11W-B pin.	R/W
b7	TCLKS	MTCLK Pin Select	0: PC2 is designated as the MTCLKE-A pin. PC3 is designated as the MTCLKF-A pin. PC0 is designated as the MTCLKG-A pin. PC1 is designated as the MTCLKH-A pin. 1: PB4 is designated as the MTCLKE-B pin. PB5 is designated as the MTCLKF-B pin. PB2 is designated as the MTCLKG-B pin. PB3 is designated as the MTCLKH-B pin.	R/W

PFDMTU is used to select pins for MTU unit 1.

MTUS_j Bit (MTU Pin Select) (j = 6)

Each bit selects a pin for a MTU input.

TCLKS Bit (MTCLK Pin Select)

This bit selects a pin for a MTCLK input of the MTU

17.3.2.17 Port Function Control Register E (PFENET)

Address: 0008 C10Eh

b7	b6	b5	b4	b3	b2	b1	b0
EE	—	—	PHYMODE	ENETE3	ENETE2	ENETE1	ENETE0
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ENETE0	Ethernet Pin Enable 0	0: The ET_WOL pin is disabled. 1: The ET_WOL pin is enabled.	R/W
b1	ENETE1	Ethernet Pin Enable 1	0: The ET_LINKSTA pin is disabled. 1: The ET_LINKSTA pin is enabled.	R/W
b2	ENETE2	Ethernet Pin Enable 2	0: The ET_EXOUT pin is disabled. 1: The ET_EXOUT pin is enabled.	R/W
b3	ENETE3	Ethernet Pin Enable 3	0: The ET_TX_ER pin is disabled. 1: The ET_TX_ER pin is enabled.	R/W
b4	PHYMODE	Ethernet Mode Setting	0: RMI mode 1: MII mode	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	EE	Ethernet Pins Enable	0: All pin functions for the Ethernet interface are disabled. 1: All pin functions for the Ethernet interface are enabled.	R/W

PFENET is used to select I/O pins for the Ethernet interface.

ENETE0 to ENETE3 Bits (Ethernet Pin Enable)

These bits select pins for the Ethernet interface.

PHYMODE Bit (Ethernet Mode Setting)

This bit selects the PHY mode for the Ethernet interface.

Table 17.29 lists the relationship between the setting of the PHYMODE bit and the mode of the Ethernet interface.

EE Bit (Ethernet Pins Enable)

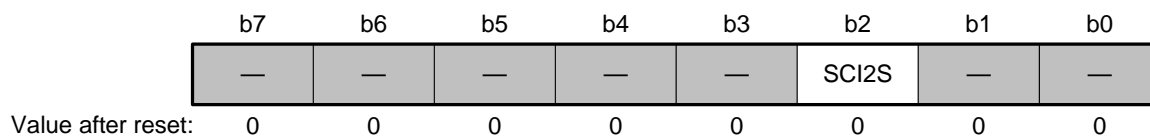
This bit enables or disables all pin functions for the Ethernet interface.

Table 17.29 Relationship between the PHYMODE Bit Setting and Ethernet Mode (100-Pin LQFP)

PHYMODE	Ethernet Mode	Pin to be Used for the Ethernet Interface	Pin Allocation	Remark
0	RMII mode	ET_MDC	PA4	
		ET_MDIO	PA3	
		ET_WOL	PA7	Enabled when ENETE0 = 1
		ET_LINKSTA	PA5	Enabled when ENETE1 = 1
		ET_EXOUT	PA6	Enabled when ENETE2 = 1
		REF50CK	PB2	
		RMII_TXD0	PB5	
		RMII_TXD1	PB6	
		RMII_TXD_EN	PB4	
		RMII_RXD0	PB1	
		RMII_RXD1	PB0	
		RMII_RX_ER	PB3	
		RMII_CRS_DV	PB7	
		1	MII mode	ET_MDC
ET_MDIO	PA3			
ET_WOL	PA7			Enabled when ENETE0 = 1
ET_LINKSTA	PA5			Enabled when ENETE1 = 1
ET_EXOUT	PA6			Enabled when ENETE2 = 1
ET_TX_CLK	PC4			
ET_ETXD0	PB5			
ET_ETXD1	PB6			
ET_ETXD2	PC5			
ET_ETXD3	PC6			
ET_TX_EN	PB4			
ET_TX_ER	PC3			Enabled when ENETE3 = 1
ET_COL	PC7			
ET_CRS	PB7			
ET_RX_CLK	PB2			
ET_ERXD0	PB1			
ET_ERXD1	PB0			
ET_ERXD2	PC1			
ET_ERXD3	PC0			
ET_RX_DV	PC2			
ET_RX_ER	PB3			

17.3.2.18 Port Function Control Register F (PFFSCI)

Address: 0008 C10Fh



Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b2	SCI2S	SCI2 Pin Select	0: P12 is designated as the RxD2-A pin. P51 is designated as the SCK2 pin. P13 is designated as the TxD2-A pin. 1: P52 is designated as the RxD2-B pin. P51 is designated as the SCK2 pin. P50 is designated as the TxD2-B pin.	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PFFSCI is used to select pins for SCI.

SCInS Bit (SCIn Pin Select) (n = 2)

Each bit selects a pin for an SCI channel-n input/output.

17.3.2.19 Port Function Control Register G (PFGSPI)

Address:0008 C110h

	b7	b6	b5	b4	b3	b2	b1	b0
	SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	RSPIS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPIS	RSPI Pin Select	0: PC7 is designated as the MISOA-A pin. PC6 is designated as the MOSIA-A pin. PC5 is designated as the RSPCKA-A pin. PC4 is designated as the SSLA0-A pin. PC0 is designated as the SSLA1-A pin. PC1 is designated as the SSLA2-A pin. PC2 is designated as the SSLA3-A pin. 1: PA7 is designated as the MISOA-B pin. PA6 is designated as the MOSIA-B pin. PA5 is designated as the RSPCKA-B pin. PA4 is designated as the SSLA0-B pin. PA0 is designated as the SSLA1-B pin. PA1 is designated as the SSLA2-B pin. PA2 is designated as the SSLA3-B pin.	R/W
b1	RSPCKE	RSPCKA Output Enable	0: The RSPCKA pin is disabled. 1: The RSPCKA pin is enabled.	R/W
b2	MOSIE	MOSIA Output Enable	0: The MOSIA pin is disabled. 1: The MOSIA pin is enabled.	R/W
b3	MISOE	MISOA Output Enable	0: The MISOA pin is disabled. 1: The MISOA pin is enabled.	R/W
b4	SSL0E	SSLA0 Output Enable	0: The SSLA0 pin is disabled. 1: The SSLA0 pin is enabled.	R/W
b5	SSL1E	SSLA1 Output Enable	0: The SSLA1 pin is disabled. 1: The SSLA1 pin is enabled.	R/W
b6	SSL2E	SSLA2 Output Enable	0: The SSLA2 pin is disabled. 1: The SSLA2 pin is enabled.	R/W
b7	SSL3E	SSLA3 Output Enable	0: The SSLA3 pin is disabled. 1: The SSLA3 pin is enabled.	R/W

PFGSPI is used to select I/O pins for RSPI channel 0.

RSPIS Bit (RSPI Pin Select)

This bit selects a pin for an RSPI input/output.

As an enable bit is provided for each RSPI input/output pin, the input/output pin is selectable while the corresponding enable bit is 1. Otherwise, the pin cannot be selected.

RSPCKE Bit (RSPCKA Output Enable)

This bit enables or disables the output of the RSPCKA pin. Set this bit to 1 to use the RSPCKA pin.

MOSIE Bit (MOSIA Output Enable)

This bit enables or disables the output of the MOSIA pin. Set this bit to 1 to use the MOSIA pin.

MISOE Bit (MISOA Output Enable)

This bit enables or disables the output of the MISOA pin. Set this bit to 1 to use the MISOA pin.

SSL0E Bit (SSLA0 Output Enable)

This bit enables or disables the output of the SSLA0 pin. Set this bit to 1 to use the SSLA0 pin.

SSL1E Bit (SSLA1 Output Enable)

This bit enables or disables the output of the SSLA1 pin. Set this bit to 1 to use the SSLA1 pin.

SSL2E Bit (SSLA2 Output Enable)

This bit enables or disables the output of the SSLA2 pin. Set this bit to 1 to use the SSLA2 pin.

SSL3E Bit (SSLA3 Output Enable)

This bit enables or disables the output of the SSLA3 pin. Set this bit to 1 to use the SSLA3 pin.

17.3.2.20 Port Function Control Register H (PFHSPI)

Address: 0008 C111h

b7	b6	b5	b4	b3	b2	b1	b0
SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	RSPIS
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPIS	RSPI Pin Select	0: P30 is designated as the MISOB-A pin. P26 is designated as the MOSIB-A pin. P27 is designated as the RSPCKB-A pin. P31 is designated as the SSLB0-A pin. P50 is designated as the SSLB1-A pin. P51 is designated as the SSLB2-A pin. P52 is designated as the SSLB3-A pin. 1: PE7 is designated as the MISOB-B pin. PE6 is designated as the MOSIB-B pin. PE5 is designated as the RSPCKB-B pin. PE4 is designated as the SSLB0-B pin. PE0 is designated as the SSLB1-B pin. PE1 is designated as the SSLB2-B pin. PE2 is designated as the SSLB3-B pin.	R/W
b1	RSPCKE	RSPCKB Output Enable	0: The RSPCKB pin is disabled. 1: The RSPCKB pin is enabled	R/W
b2	MOSIE	MOSIB Output Enable	0: The MOSIB pin is disabled. 1: The MOSIB pin is enabled	R/W
b3	MISOE	MISOB Output Enable	0: The MISOB pin is disabled. 1: The MISOB pin is enabled	R/W
b4	SSL0E	SSLB0 Output Enable	0: The SSLB0 pin is disabled. 1: The SSLB0 pin is enabled	R/W
b5	SSL1E	SSLB1 Output Enable	0: The SSLB1 pin is disabled. 1: The SSLB1 pin is enabled	R/W
b6	SSL2E	SSLB2 Output Enable	0: The SSLB2 pin is disabled. 1: The SSLB2 pin is enabled	R/W
b7	SSL3E	SSLB3 Output Enable	0: The SSLB3 pin is disabled. 1: The SSLB3 pin is enabled	R/W

PFHSPI is used to select I/O pins for RSPI channel 1.

RSPIS Bit (RSPI Pin Select)

This bit selects a pin for an RSPI input/output.

As an enable bit is provided for each RSPI input/output pin, the input/output pin is selectable while the corresponding enable bit is 1. Otherwise, the pin cannot be selected.

RSPCKE Bit (RSPCKB Output Enable)

This bit enables or disables the output of the RSPCKB pin. Set this bit to 1 to use the RSPCKB pin.

MOSIE Bit (MOSIB Output Enable)

This bit enables or disables the output of the MOSIB pin. Set this bit to 1 to use the MOSIB pin.

MISOE Bit (MISOB Output Enable)

This bit enables or disables the output of the MISOB pin. Set this bit to 1 to use the MISOB pin.

SSL0E Bit (SSLB0 Output Enable)

This bit enables or disables the output of the SSLB0 pin. Set this bit to 1 to use the SSLB0 pin.

SSL1E Bit (SSLB1 Output Enable)

This bit enables or disables the output of the SSLB1 pin. Set this bit to 1 to use the SSLB1 pin.

SSL2E Bit (SSLB2 Output Enable)

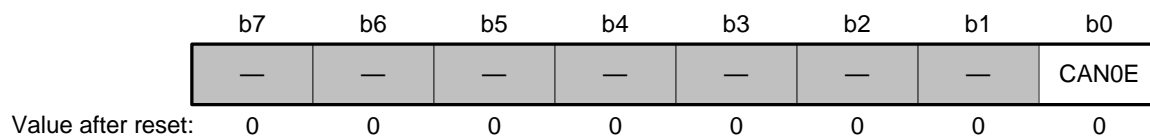
This bit enables or disables the output of the SSLB2 pin. Set this bit to 1 to use the SSLB2 pin.

SSL3E Bit (SSLB3 Output Enable)

This bit enables or disables the output of the SSLB3 pin. Set this bit to 1 to use the SSLB3 pin.

17.3.2.21 Port Function Control Register J (PFJCAN)

Address: 0008 C113h



Bit	Symbol	Bit Name	Description	R/W
b0	CAN0E	CAN0 Pins Enable	0: The CTX0 and CRX0 pins are disabled. 1: The CTX0 and CRX0 pins are enabled.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PFJCAN is used to select I/O pins for the CAN.

CANnE Bit (CANn Pins Enable) (n = 0)

This bit enables or disables the CANn pins. Set this bit to 1 to use the CANn pins.

17.3.2.22 Port Function Control Register K (PFKUSB)

Address:0008 C114h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	USBE	PDHZS	PUPHZS	USBMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	USBMD[1:0]	USB Mode Setting	b1 b0 0 0: Select function mode for the USB0 pins. 0 1: Select host mode for the USB0 pins. 1 0: Select host/function mode for the USB0 pin. (as an optional function)* 1 1: Select OTG mode for the USB0 pins.	R/W
b2	PUPHZS	PUPHZ Select	0: USB0_DPUPE pin is for output of the high and low levels (external pull-up control signal). 1: USB0_DPUPE pin is for high-level output or the Hi-Z state (pull-up output is from the USB0_DP pin).	R/W
b3	PDHZS	PDHZ Select	0: USB0_DPRPD and USB0_DRPD pins are for output of the high and low levels (external pull-down control signals). 1: USB0_DPRPD and USB0_DRPD pins are for low-level output or the Hi-Z state (pull-down output is from the USB0_DP and USB0_DM pins).	R/W
b4	USBE	USB Enable	0: All pin functions for USB0 are disabled. 1: All pin functions for USB0 are enabled.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note : * Please contact a Renesas Electronics sales office for details of the optional function.

PFKUSB is used to set I/O pins for the USB0.

USBMD[1:0] Bits (USB Mode Setting)

These bits select a mode for the USB.

Table 17.30 lists the relationship between the setting of the USBMD[1:0] bits and the mode of the USB.

PUPHZS Bit (PUPHZ Select)

This bit selects the output mode (external pull-up control/pin pull-up output) for the DPUPE pin of the USB.

When the PUPHZS bit is set to 0, the control signal output mode for the external pull-up IC is selected and the high-active control signal is output from the DPUPE pin. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the low-level output state.

When the PUPHZS bit is set to 1, the output mode where the DP pin is directly pulled-up is selected. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the high-impedance state.

PDHVS Bit (PDHZ Select)

This bit selects the output mode (external pull-down control/pin pull-down output) for the DPRPD and DRPD pins of the USB.

When the PDHVS bit is set to 0, the control signal output mode for the external pull-down IC is selected and the high-active control signal is output from the DPRPD and DRPD pins. When the DP and DM pins are pulled down, the DPRPD and DRPD pins are set to the high-level output state. When the pulled-down state of the DP and DM pins is canceled, the DPRPD and DRPD pins are set to the low-level output state.

When the PDHVS bit is set to 1, the output mode where the DP and DM pins are directly pulled-down is selected. When the DP and DM pins are pulled down, the DPRPD and DRPD pins are set to the low-level output state. When the pulled-down state of the DP and DM pins is canceled, the DPRPD and DRPD pins are set to the high-impedance state.

USBE Bit (USB Enable)

This bit enables all pin functions for the USB0.

Table 17.30 Relationship between the USBMD[1:0] Bits Setting and USB Mode (USB0)

USBMD1	USBMD0	USB0 Mode	Pin to be Used for the USB	Pin Allocation	Remarks
0	0	Function mode	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_VBUS	P16	
			USB0_DPUPE-B	P14	Selection of -B side
0	1	Host mode	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_OVRCURA	P14	
			USB0_VBUSEN-B	P16	Selection of -B side
1	0	Host/function mode (as an optional function)*	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_OVRCURA	P15	
			USB0_VBUS	P17	
			USB0_DRPD	P30	
			USB0_DPUPE-A	P10	Selection of -A side
			USB0_VBUSEN-A	P11	Selection of -A side
1	1	OTG mode	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_OVRCURA	P14	
			USB0_OVRCURB	P16	
			USB0_DPRPD	P25	
			USB0_DRPD	P22	
			USB0_EXICEN	P21	
			USB0_ID	P20	
			USB0_DPUPE-A	P23	Selection of -A side
USB0_VBUSEN-A	P24	Selection of -A side			

Note: * Please contact a Renesas Electronics sales office for details of the optional function.

17.3.2.23 Port Function Control Register M (PFMPOE)

Address:0008 C116h

	b7	b6	b5	b4	b3	b2	b1	b0
	POE7E	POE6E	POE5E	POE4E	POE3E	POE2E	POE1E	POE0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	POE0E	POE0 Input Enable	0: Designated as an I/O port pin.	R/W*
b1	POE1E	POE1 Input Enable	1: Designated as the POEn# input pin (n = 0 to 7)	R/W*
b2	POE2E	POE2 Input Enable		R/W*
b3	POE3E	POE3 Input Enable		R/W*
b4	POE4E	POE4 Input Enable		R/W*
b5	POE5E	POE5 Input Enable		R/W*
b6	POE6E	POE6 Input Enable		R/W*
b7	POE7E	POE7 Input Enable		R/W*

Note:*The first round of writing after a reset is enabled; though subsequent writing is not allowed.

PFMPOE enables or disables POE input pins.

To prevent the error in the system, be sure to write to this register after a reset.

The first round of writing after a reset is only enabled.

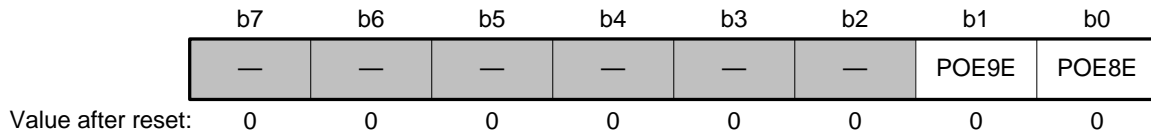
POEnE Bit (POEn Input Enable) (n = 0 to 7)

Each bit enables or disables the corresponding POEn# input.

To use POEn#, set the corresponding POEnE bit to 1.

17.3.2.24 Port Function Control Register N (PFNPOE)

Address: 0008 C117h



Bit	Symbol	Bit Name	Description	R/W
b0	POE8E	POE8 Input Enable	0: Designated as an I/O port pin.	R/W*
b1	POE9E	POE9 Input Enable	1: Designated as the POEn# input pin (n = 8, 9)	R/W*
b7 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note : * The first round of writing after a reset is enabled; though subsequent writing is not allowed.

PFNPOE enables or disables POE input pins.

To prevent the error in the system, be sure to write to this register after a reset.

The first round of writing after a reset is only enabled.

POEnE Bit (POEn Input Enable) (n = 8, 9)

Each bit enables or disables the corresponding POEn# input

To use POEn#, set the corresponding POEnE bit to 1.

17.3.3 Settings of Ports


When individual pins for peripheral modules are enabled, the settings for each port are modified.

An input pin for the peripheral module is specified independently by the peripheral module. To use an input pin for the peripheral module, the corresponding bit in the input buffer control register (ICR) should be set to 1 to enable the input buffer, except for the port register read, data bus input, NMI, and POE pin inputs.

The pins that function as output pins and I/O pins should be enabled for respective peripheral modules. If a conflict occurs among the output signal enable settings for peripheral modules, that are multiplexed to the same port, the priority will be handled according to the port-multiplexed priority.

Table 17.31 lists the port-multiplexed priority for peripheral modules.

Table 17.31 Port-Multiplexed Priority for Peripheral Modules (100-Pin LQFP)

Priority	Module Name	Output Pins	
High  Low	1	External bus (Data)	D0 to D15 (Data bus)
	2	External bus	RD#, WR#, WR0# to WR1#, BC0# to BC1#, BCLK, A0 to A23 (Address bus)
	3	External bus (CS)	CS0# to CS7# (Chip select)
	4	RSPI0, RSP11	RSPCKn, MOSIn, MISOn, SSLn0 to SSLn3 (n = A, B)
	5	USB0	USB0_DPUPE, USB0_VBUSEN, USB0_EXICEN, USB0_DRPD, USB0_DPRPD
	6	CAN0	CTX0
	7	EtherNET	ET_MDC, ET_MDIO, ET_EXOUT, ET_WOL, ET_TX_EN, ET_TX_ER, ET_ETXD0 to ET_ETXD3, RMII_TXD_EN, RMII_TXD0, RMII_TXD1
	8	MTU0 to MTU4, MTU6 to MTU10	MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D, MTIOC1A, MTIOC1B, MTIOC2A, MTIOC2B, MTIOC3A, MTIOC3B, MTIOC3C, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6A, MTIOC6B, MTIOC6C, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC8A, MTIOC8B, MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D, MTIOC10A, MTIOC10B, MTIOC10C, MTIOC10D
	9	TMR0 to TMR3	TMO0 to TMO3
	10	SCI0 to SCI3, SCI5 to SCI6	SCK0 to SCK3, SCK5 to SCK6, TxD0 to TxD3, TxD5 to TxD6
	11	RTC	RTCOU0
	12	PPG0, PPG1	PO0 to PO15, PO16 to PO31
	13	RIIC0	SCL0, SDA0
	14	DA	DA1
	15	I/O PORT	P05, P07, P12 to P14, P16, P20 to P27, P30 to P34, P50 to P52, P54 to P55, PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7

17.3.4 List of Output Enable Settings

Table 17.32 lists the output enable settings for each port.

For details on the applicable output signals, see descriptions of the registers for each peripheral module.

Setting the port function control register changes the functions of peripheral-module pins with names ending in A to D.

Table 17.32 Output Enable Settings for Each Port (100-Pin LQFP) (1 / 10)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P05	DA	DA1		DACR.DAOE1 = 1
	PORT0	P05		PORT0.DDR.B5 = 1
P07	PORT0	P07		PORT0.DDR.B7 = 1
P12	RIIC0	SCL0		RIIC0.ICCR1.ICE = 1
	PORT1	P12		PORT1.DDR.B2 = 1
P13	MTU0	MTIOC0B		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	TMR3	TMO3		TMO3.TCSR.OSA[1: 0] = 01/10/11 or TMO3.TCSR.OSB[1: 0] = 01/10/11
	SCI2	TxD2-A	PFFSCI.SCI2S = 0	SCI2.SCR.TE = 1
	PPG0	PO13		PPG0.NDERH.NDER13 = 1
	RIIC0	SDA0		RIIC0.ICCR1.ICE = 1
	PORT1	P13		PORT1.DDR.B3 = 1
	P14	USB0	USB0_DPUPE-B	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 00
MTU3		MTIOC3A		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
PPG0		PO15		PPG0.NDERH.NDER15 = 1
PORT1		P14		PORT1.DDR.B4 = 1
P16	USB0	USB0_VBUSEN-B	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 01	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3C		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	TMR2	TMO2		TMO2.TCSR.OSA[1: 0] = 01/10/11 or TMO2.TCSR.OSB[1: 0] = 01/10/11
	PPG0	PO14		PPG0.NDERH.NDER14 = 1
	PORT1	P16		PORT1.DDR.B6 = 1
P20	MTU1	MTIOC1A		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	SCI0	TxD0		SCI0.SCR.TE = 1
	PPG0	PO0		PPG0.NDERL.NDER0 = 1
	PORT2	P20		PORT2.DDR.B0 = 1
P21	USB0	USB0_EXICEN	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 11	(The signal output state is specified by the peripheral module settings.)
	MTU1	MTIOC1B		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO1		PPG0.NDERL.NDER1 = 1
	PORT2	P21		PORT2.DDR.B1 = 1

Table 17.32 Output Enable Settings for Each Port (100-Pin LQFP) (2 / 10)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P22	USB0	USB0_DRPD	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 10/ 11	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3B		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	TMR0	TMO0		TMO0.TCSR.OSA[1: 0] = 01/10/11 or TMO0.TCSR.OSB[1: 0] = 01/10/11
	SCI0	SCK0		When SCI0.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI0.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO2		PPG0.NDERL.NDER2 = 1
	PORT2	P22		PORT2.DDR.B2 = 1
P23	USB0	USB0_DPUPE-A	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 10/ 11	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3D		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	SCI3	TxD3		SCI3.SCR.TE = 1
	PPG0	PO3		PPG0.NDERL.NDER3 = 1
	PORT2	P23		PORT2.DDR.B3 = 1
	P24	External bus (CS)	CS4#	PF0CSE.CS4E = 1
USB0		USB0_VBUSEN-A	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 10/ 11	(The signal output state is specified by the peripheral module settings.)
MTU4		MTIOC4A		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
SCI3		SCK3		When SCI3.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI3.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
PPG0		PO4		PPG0.NDERL.NDER4 = 1
PORT2		P24		PORT2.DDR.B4 = 1
P25		External bus (CS)	CS5#	PF0CSE.CS5E = 1
	USB0	USB0_DPRPD	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 11	(The signal output state is specified by the peripheral module settings.)
	MTU4	MTIOC4C		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO5		PPG0.NDERL.NDER5 = 1
	PORT2	P25		PORT2.DDR.B5 = 1

Table 17.32 Output Enable Settings for Each Port (100-Pin LQFP) (3 / 10)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P26	External bus (CS)	CS6#	PF0CSE.CS6E = 1	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSP11	MOSIB-A	PFHSPI.MOSIE = 1 PFHSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU2	MTIOC2A		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	TMR1	TMO1		TMO1.TCSR.OSA[1: 0] = 01/10/11 or TMO1.TCSR.OSB[1: 0] = 01/10/11
	SCI1	TxD1		SCI1.SCR.TE = 1
	PPG0	PO6		PPG0.NDERL.NDER6 = 1
	PORT2	P26		PORT2.DDR.B6 = 1
P27	External bus (CS)	CS7#	PF0CSE.CS7E = 1	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSP11	RSPCKB-A	PFHSPI.RSPCKE = 1 PFHSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU2	MTIOC2B		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	SCI1	SCK1		When SCI1.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI1.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO7		PPG0.NDERL.NDER7 = 1
	PORT2	P27		PORT2.DDR.B7 = 1
	P30	RSP11	MISOB-A	PFHSPI.MISOE = 1 PFHSPI.RSPIS = 0
MTU4		MTIOC4B-A	PFCMTU.MTUS5 = 0	For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
PPG0		PO8		PPG0.NDERH.NDER8 = 1
PORT3		P30		PORT3.DDR.B0 = 1
P31	RSP11	SSLB0-A	PFHSPI.SSL0E = 1 PFHSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU4	MTIOC4D-A	PFCMTU.MTUS5 = 0	For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO9		PPG0.NDERH.NDER9 = 1
	PORT3	P31		PORT3.DDR.B1 = 1
P32	CAN0	CTX0	PFJCAN.CAN0E = 1	(The signal output state is specified by the peripheral module settings.)
	MTU0	MTIOC0C		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	SCI6	TxD6		SCI6.SCR.TE = 1
	RTC	RTCOUT		RCR2.RTCOE = 1
	PPG0	PO10		PPG0.NDERH.NDER10 = 1
	PORT3	P32		PORT3.DDR.B2 = 1

Table 17.32 Output Enable Settings for Each Port (100-Pin LQFP) (4 / 10)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P33	MTU0	MTIOC0D		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO11		PPG0.NDERH.NDER11 = 1
	PORT3	P33		PORT3.DDR.B3 = 1
P34	MTU0	MTIOC0A		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	SCI6	SCK6		When SCI6.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI6.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO12		PPG0.NDERH.NDER12 = 1
	PORT3	P34		PORT3.DDR.B4 = 1
P35	(NA)	(NA)		
P40	PORT4	P40		PORT4.DDR.B0 = 1
P41	PORT4	P41		PORT4.DDR.B1 = 1
P42	PORT4	P42		PORT4.DDR.B2 = 1
P43	PORT4	P43		PORT4.DDR.B3 = 1
P44	PORT4	P44		PORT4.DDR.B4 = 1
P45	PORT4	P45		PORT4.DDR.B5 = 1
P46	PORT4	P46		PORT4.DDR.B6 = 1
P47	PORT4	P47		PORT4.DDR.B7 = 1
P50	External bus	WR# WR0#		SYSCR0.EXBE = 1
	RSP11	SSLB1-A	PFHSP1.SSL1E = 1 PFHSP1.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	SCI2	TxD2-B	PFFSCI.SCI2S = 1	SCI2.SCR.TE = 1
	PORT5	P50		PORT5.DDR.B0 = 1
P51	External bus	WR1# BC1#	PF5BUS.WR1BC1E = 1	SYSCR0.EXBE = 1
	RSP11	SSLB2-A	PFHSP1.SSL2E = 1 PFHSP1.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	SCI2	SCK2		When SCI2.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI2.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PORT5	P51		PORT5.DDR.B1 = 1
P52	External bus	RD#	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB3-A	PFHSP1.SSL3E = 1 PFHSP1.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	PORT5	P52		PORT5.DDR.B2 = 1
P53	External bus	BCLK		PORT5.DDR.B3 = 1
P54	MTU4	MTIOC4B-B	PFCMTU.MTUS5 = 1	For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PORT5	P54		PORT5.DDR.B4 = 1

Table 17.32 Output Enable Settings for Each Port (100-Pin LQFP) (5 / 10)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P55	EtherNET	ET_EXOUT	PFENET.EE = 1 PFENET.ENETE2 = 1	(The signal output state is specified by the peripheral module settings.)
	MTU4	MTIOC4D-B	PFCMTU.MTUS5 = 1	For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PORT5	P55		PORT5.DDR.B5 = 1
PA0	External bus	A0 BC0#	PF4BUS.ADRLE[1: 0] = 01/ 10/11	SYSCR0.EXBE = 1
	RSPI0	SSLA1-B	PFGSPI.SSL1E = 1 PFGSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	MTU6	MTIOC6A		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO16		PPG1.NDERL.NDER0 = 1
	PORTA	PA0		PORTA.DDR.B0 = 1
PA1	External bus	A1	PF4BUS.ADRLE[1: 0] = 01/ 10/11	SYSCR0.EXBE = 1
	RSPI0	SSLA2-B	PFGSPI.SSL2E = 1 PFGSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	MTU6	MTIOC6B		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO17		PPG1.NDERL.NDER1 = 1
	PORTA	PA1		PORTA.DDR.B1 = 1
PA2	External bus	A2	PF4BUS.ADRLE[1: 0] = 01/ 10/11	SYSCR0.EXBE = 1
	RSPI0	SSLA3-B	PFGSPI.SSL3E = 1 PFGSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	MTU6	MTIOC6C		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO18		PPG1.NDERL.NDER2 = 1
	PORTA	PA2		PORTA.DDR.B2 = 1
PA3	External bus	A3	PF4BUS.ADRLE[1: 0] = 01/ 10/11	SYSCR0.EXBE = 1
	EtherNET	ET_MDIO	PFENET.EE = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU6	MTIOC6D		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO19		PPG1.NDERL.NDER3 = 1
	PORTA	PA3		PORTA.DDR.B3 = 1
PA4	External bus	A4	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPI0	SSLA0-B	PFGSPI.SSL0E = 1 PFGSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	EtherNET	ET_MDC	PFENET.EE = 1	(The signal output state is specified by the peripheral module settings.)
	MTU7	MTIOC7A		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO20		PPG1.NDERL.NDER4 = 1
	PORTA	PA4		PORTA.DDR.B4 = 1

Table 17.32 Output Enable Settings for Each Port (100-Pin LQFP) (6 / 10)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PA5	External bus	A5	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPI0	RSPCKA-B	PFGSPI.RSPCKE = 1 PFGSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU7	MTIOC7B		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO21		PPG1.NDERL.NDER5 = 1
	PORTA	PA5		PORTA.DDR.B5 = 1
PA6	External bus	A6	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPI0	MOSIA-B	PFGSPI.MOSIE = 1 PFGSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	EtherNET	ET_EXOUT	PFENET.EE = 1 PFENET.ENETE2 = 1	(The signal output state is specified by the peripheral module settings.)
	MTU8	MTIOC8A		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO22		PPG1.NDERL.NDER6 = 1
	PORTA	PA6		PORTA.DDR.B6 = 1
PA7	External bus	A7	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPI0	MISOA-B	PFGSPI.MISOE = 1 PFGSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	EtherNET	ET_WOL	PFENET.EE = 1 PFENET.ENETE0 = 1	(The signal output state is specified by the peripheral module settings.)
	MTU8	MTIOC8B		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO23		PPG1.NDERL.NDER7 = 1
	PORTA	PA7		PORTA.DDR.B7 = 1
PB0	External bus	A8	PF4BUS.ADRLE[1: 0] = 11	SYSCR0.EXBE = 1
	MTU9	MTIOC9A		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO24		PPG1.NDERH.NDER8 = 1
	PORTB	PB0		PORTB.DDR.B0 = 1
PB1	External bus	A9	PF4BUS.ADRLE[1: 0] = 11	SYSCR0.EXBE = 1
	MTU9	MTIOC9C		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO25		PPG1.NDERH.NDER9 = 1
	PORTB	PB1		PORTB.DDR.B1 = 1
PB2	External bus	A10	PF4BUS.A10E = 1	SYSCR0.EXBE = 1
	MTU9	MTIOC9B		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO26		PPG1.NDERH.NDER10 = 1
	PORTB	PB2		PORTB.DDR.B2 = 1
PB3	External bus	A11	PF4BUS.A11E = 1	SYSCR0.EXBE = 1
	MTU9	MTIOC9D		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO27		PPG1.NDERH.NDER11 = 1
	PORTB	PB3		PORTB.DDR.B3 = 1

Table 17.32 Output Enable Settings for Each Port (100-Pin LQFP) (7 / 10)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PB4	External bus	A12	PF4BUS.A12E = 1	SYSCR0.EXBE = 1
	EtherNET	ET_TX_EN	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	EtherNET	RMII_TXD_EN	PFENET.EE = 1 PFENET.PHYMODE = 0	(The signal output state is specified by the peripheral module settings.)
	MTU10	MTIOC10A		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO28		PPG1.NDERH.NDER12 = 1
	PORTB	PB4		PORTB.DDR.B4 = 1
PB5	External bus	A13	PF4BUS.A13E = 1	SYSCR0.EXBE = 1
	EtherNET	ET_ETXD0	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	EtherNET	RMII_TXD0	PFENET.EE = 1 PFENET.PHYMODE = 0	(The signal output state is specified by the peripheral module settings.)
	MTU10	MTIOC10C		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO29		PPG1.NDERH.NDER13 = 1
	PORTB	PB5		PORTB.DDR.B5 = 1
PB6	External bus	A14	PF4BUS.A14E = 1	SYSCR0.EXBE = 1
	EtherNET	ET_ETXD1	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	EtherNET	RMII_TXD1	PFENET.EE = 1 PFENET.PHYMODE = 0	(The signal output state is specified by the peripheral module settings.)
	MTU10	MTIOC10B		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO30		PPG1.NDERH.NDER14 = 1
	PORTB	PB6		PORTB.DDR.B6 = 1
PB7	External bus	A15	PF4BUS.A15E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10D		For the MTU settings, see Table 17.33, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO31		PPG1.NDERH.NDER15 = 1
	PORTB	PB7		PORTB.DDR.B7 = 1
PC0	External bus	A16	PF3BUS.A16E = 1	SYSCR0.EXBE = 1
	RSPI0	SSLA1-A	PFGSPI.SSL1E = 1 PFGSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	PORTC	PC0		PORTC.DDR.B0 = 1
PC1	External bus	A17	PF3BUS.A17E = 1	SYSCR0.EXBE = 1
	RSPI0	SSLA2-A	PFGSPI.SSL2E = 1 PFGSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	SCI5	SCK5		When SCI5.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1:0] = 01 or SMR.GM = 1 When SCI5.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1:0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PORTC	PC1		PORTC.DDR.B1 = 1
PC2	External bus	A18	PF3BUS.A18E = 1	SYSCR0.EXBE = 1
	RSPI0	SSLA3-A	PFGSPI.SSL3E = 1 PFGSPI.RSPIS = 0	(The signal output state is specified by the peripheral module settings.)
	PORTC	PC2		PORTC.DDR.B2 = 1

Table 17.32 Output Enable Settings for Each Port (100-Pin LQFP) (8 / 10)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PC3	External bus	A19	PF3BUS.A19E = 1	SYSCR0.EXBE = 1
	EtherNET	ET_TX_ER	PFENET.EE = 1 PFENET.ENETE3 = 1	(The signal output state is specified by the peripheral module settings.)
	SCI5	TxD5		SCI5.SCR.TE = 1
	PORTC	PC3		PORTC.DDR.B3 = 1
PC4	External bus	A20	PF3BUS.A20E = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS3#	PF0CSE.CS3E = 1	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSPI0	SSLA0-A	PFGSPI.SSL0E = 1 PFGSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTC	PC4		PORTC.DDR.B4 = 1
PC5	External bus	A21	PF3BUS.A21E = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS2#	PF0CSE.CS2E = 1	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSPI0	RSPCKA-A	PFGSPI.RSPCKE = 1 PFGSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	EtherNET	ET_ETXD2	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	PORTC	PC5		PORTC.DDR.B5 = 1
PC6	External bus	A22	PF3BUS.A22E = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS1#	PF0CSE.CS1E = 1	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSPI0	MOSIA-A	PFGSPI.MOSIE = 1 PFGSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	EtherNET	ET_ETXD3	PFENET.EE = 1 PFENET.PHYMODE = 1	(The signal output state is specified by the peripheral module settings.)
	PORTC	PC6		PORTC.DDR.B6 = 1
PC7	External bus	A23	PF3BUS.A23E = 1	SYSCR0.EXBE = 1
	External bus (CS)	CS0#	PF0CSE.CS0E = 1	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSPI0	MISOA-A	PFGSPI.MISOE = 1 PFGSPI.RSPIS = 0	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTC	PC7		PORTC.DDR.B7 = 1
PD0	External bus (Data)	D0		SYSCR0.EXBE = 1
	PORTD	PD0		PORTD.DDR.B0 = 1
PD1	External bus (Data)	D1		SYSCR0.EXBE = 1
	PORTD	PD1		PORTD.DDR.B1 = 1
PD2	External bus (Data)	D2		SYSCR0.EXBE = 1
	PORTD	PD2		PORTD.DDR.B2 = 1
PD3	External bus (Data)	D3		SYSCR0.EXBE = 1
	PORTD	PD3		PORTD.DDR.B3 = 1

Table 17.32 Output Enable Settings for Each Port (100-Pin LQFP) (9 / 10)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PD4	External bus (Data)	D4		SYSCR0.EXBE = 1
	PORTD	PD4		PORTD.DDR.B4 = 1
PD5	External bus (Data)	D5		SYSCR0.EXBE = 1
	PORTD	PD5		PORTD.DDR.B5 = 1
PD6	External bus (Data)	D6		SYSCR0.EXBE = 1
	PORTD	PD6		PORTD.DDR.B6 = 1
PD7	External bus (Data)	D7		SYSCR0.EXBE = 1
	PORTD	PD7		PORTD.DDR.B7 = 1
PE0	External bus (Data)	D8	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB1-B	PFHSPI.SSL1E = 1 PFHSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	PORTE	PE0		PORTE.DDR.B0 = 1
PE1	External bus (Data)	D9	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB2-B	PFHSPI.SSL2E = 1 PFHSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	PORTE	PE1		PORTE.DDR.B1 = 1
PE2	External bus (Data)	D10	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB3-B	PFHSPI.SSL3E = 1 PFHSPI.RSPIS = 1	(The signal output state is specified by the peripheral module settings.)
	PORTE	PE2		PORTE.DDR.B2 = 1
PE3	External bus (Data)	D11	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	PORTE	PE3		PORTE.DDR.B3 = 1
PE4	External bus (Data)	D12	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	SSLB0-B	PFHSPI.SSL0E = 1 PFHSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTE	PE4		PORTE.DDR.B4 = 1
PE5	External bus (Data)	D13	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	RSPCKB-B	PFHSPI.RSPCKE = 1 PFHSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTE	PE5		PORTE.DDR.B5 = 1
PE6	External bus (Data)	D14	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	MOSIB-B	PFHSPI.MOSIE = 1 PFHSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTE	PE6		PORTE.DDR.B6 = 1

Table 17.32 Output Enable Settings for Each Port (100-Pin LQFP) (10 / 10)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PE7	External bus (Data)	D15	PF5BUS.DHE = 1	SYSCR0.EXBE = 1
	RSP11	MISOB-B	PFHSPI.MISOE = 1 PFHSPI.RSPIS = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	PORTE	PE7		PORTE.DDR.B7 = 1

Table 17.33 Settings to Enable Output on the Various MTU Pins (1 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins	
MTU0	MTIOC0A	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TIORH.IOA[3] = 0 MTU0.TIORH.IOA[1:0] = 01/10/11	
		PWM mode 1	MTU0.TMDR.MD[3:0] = 0010	MTU0.TIORH.IOA[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	Except MTU0.TCR.CCLR[2:0] = 001 MTU0.TIORH.IOA[1:0] = 01/10/11	
	MTIOC0B	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TIORH.IOB[3] = 0 MTU0.TIORH.IOB[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	Except MTU0.TCR.CCLR[2:0] = 010 MTU0.TIORH.IOB[1:0] = 01/10/11	
	MTIOC0C	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TMDR.BFA = 0 MTU0.TIORL.IOC[3] = 0 MTU0.TIORL.IOC[1:0] = 01/10/11	
		PWM mode 1	MTU0.TMDR.MD[3:0] = 0010	MTU0.TMDR.BFA = 0 MTU0.TMDR.BFB = 0 MTU0.TIORL.IOC[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	MTU0.TMDR.BFA = 0 Except MTU0.TCR.CCLR[2:0] = 101 MTU0.TIORL.IOC[1:0] = 01/10/11	
	MTIOC0D	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TMDR.BFB = 0 MTU0.TIORL.IOD[3] = 0 MTU0.TIORL.IOD[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	MTU0.TMDR.BFB = 0 Except MTU0.TCR.CCLR[2:0] = 110 MTU0.TIORL.IOD[1:0] = 01/10/11	
	MTU1	MTIOC1A	Normal operation	MTU1.TMDR.MD[3:0] = 0000	MTU1.TIOR.IOA[3] = 0 MTU1.TIOR.IOA[1:0] = 01/10/11
			PWM mode 1	MTU1.TMDR.MD[3:0] = 0010	MTU1.TIOR.IOA[1:0] = 01/10/11
PWM mode 2			MTU1.TMDR.MD[3:0] = 0011	Except MTU1.TCR.CCLR[1:0] = 01 MTU1.TIOR.IOA[1:0] = 01/10/11	
Phase count mode 1			MTU1.TMDR.MD[3:0] = 0100	MTU1.TIOR.IOA[3] = 0 MTU1.TIOR.IOA[1:0] = 01/10/11	
Phase count mode 2			MTU1.TMDR.MD[3:0] = 0101		
Phase count mode 3			MTU1.TMDR.MD[3:0] = 0110		
Phase count mode 4			MTU1.TMDR.MD[3:0] = 0111		
MTIOC1B		Normal operation	MTU1.TMDR.MD[3:0] = 0000	MTU1.TIOR.IOB[3] = 0 MTU1.TIOR.IOB[1:0] = 01/10/11	
		PWM mode 2	MTU1.TMDR.MD[3:0] = 0011	Except MTU1.TCR.CCLR[1:0] = 10 MTU1.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 1	MTU1.TMDR.MD[3:0] = 0100	MTU1.TIOR.IOB[3] = 0 MTU1.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 2	MTU1.TMDR.MD[3:0] = 0101		
		Phase count mode 3	MTU1.TMDR.MD[3:0] = 0110		
		Phase count mode 4	MTU1.TMDR.MD[3:0] = 0111		
		MTU2	MTIOC2A	Normal operation	MTU2.TMDR.MD[3:0] = 0000
PWM mode 1				MTU2.TMDR.MD[3:0] = 0010	MTU2.TIOR.IOA[1:0] = 01/10/11
PWM mode 2	MTU2.TMDR.MD[3:0] = 0011			Except MTU2.TCR.CCLR[1:0] = 01 MTU2.TIOR.IOA[1:0] = 01/10/11	
Phase count mode 1	MTU2.TMDR.MD[3:0] = 0100			MTU2.TIOR.IOA[3] = 0 MTU2.TIOR.IOA[1:0] = 01/10/11	
Phase count mode 2	MTU2.TMDR.MD[3:0] = 0101				
Phase count mode 3	MTU2.TMDR.MD[3:0] = 0110				
Phase count mode 4	MTU2.TMDR.MD[3:0] = 0111				

Table 17.33 Settings to Enable Output on the Various MTU Pins (2 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins	
MTU2	MTIOC2B	Normal operation	MTU2.TMDR.MD[3:0] = 0000	MTU2.TIOR.IOB[3] = 0 MTU2.TIOR.IOB[1:0] = 01/10/11	
		PWM mode 2	MTU2.TMDR.MD[3:0] = 0011	Except MTU2.TCR.CCLR[1:0] = 10 MTU2.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 1	MTU2.TMDR.MD[3:0] = 0100	MTU2.TIOR.IOB[3] = 0	
		Phase count mode 2	MTU2.TMDR.MD[3:0] = 0101	MTU2.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 3	MTU2.TMDR.MD[3:0] = 0110		
		Phase count mode 4	MTU2.TMDR.MD[3:0] = 0111		
MTU3	MTIOC3A	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTU3.TIORH.IOA[3] = 0 MTU3.TIORH.IOA[1:0] = 01/10/11	
		PWM mode 1	MTU3.TMDR.MD[3:0] = 0010	MTU3.TIORH.IOA[1:0] = 01/10/11	
		Reset-synchronized PWM mode	MTU3.TMDR.MD[3:0] = 1000	MTUA.TOCR1.PSYE = 1	
		Complementary PWM mode 1	MTU3.TMDR.MD[3:0] = 1101		
		Complementary PWM mode 2	MTU3.TMDR.MD[3:0] = 1110		
		Complementary PWM mode 3	MTU3.TMDR.MD[3:0] = 1111		
	MTIOC3B	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTUA.TOER.OE3B = 1 MTU3.TIORH.IOB[3] = 0 MTU3.TIORH.IOB[1:0] = 01/10/11	
		Reset-synchronized PWM mode	MTU3.TMDR.MD[3:0] = 1000	MTUA.TOER.OE3B = 1	
		Complementary PWM mode 1	MTU3.TMDR.MD[3:0] = 1101		
		Complementary PWM mode 2	MTU3.TMDR.MD[3:0] = 1110		
		Complementary PWM mode 3	MTU3.TMDR.MD[3:0] = 1111		
		MTIOC3C	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTU3.TMDR.BFA = 0 MTU3.TIORL.IOC[3] = 0 MTU3.TIORL.IOC[1:0] = 01/10/11
PWM mode 1	MTU3.TMDR.MD[3:0] = 0010		MTU3.TMDR.BFA = 0 MTU3.TMDR.BFB = 0 MTU3.TIORL.IOC[1:0] = 01/10/11		
MTIOC3D	Normal operation	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTUA.TOER.OE3D = 1 MTU3.TMDR.BFB = 0 MTU3.TIORL.IOD[3] = 0 MTU3.TIORL.IOD[1:0] = 01/10/11	
		Reset-synchronized PWM mode	MTU3.TMDR.MD[3:0] = 1000	MTUA.TOER.OE3D = 1	
		Complementary PWM mode 1	MTU3.TMDR.MD[3:0] = 1101		
		Complementary PWM mode 2	MTU3.TMDR.MD[3:0] = 1110		
		Complementary PWM mode 3	MTU3.TMDR.MD[3:0] = 1111		
		MTU4	MTIOC4A	Normal operation	MTU4.TMDR.MD[3:0] = 0000
PWM mode 1	MTU4.TMDR.MD[3:0] = 0010			MTUA.TOER.OE4A = 1 MTU4.TIORH.IOA[1:0] = 01/10/11	
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000			MTUA.TOER.OE4A = 1	
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101				
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110				
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111				
	Reset-synchronized PWM mode (linked operation with MTU3)		Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4A = 1
			Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000	
			Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000	
			Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000	

Table 17.33 Settings to Enable Output on the Various MTU Pins (3 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins
MTU4	MTIOC4B	Normal operation	MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4B = 1 MTU4.TIORH.IOB[3] = 0 MTU4.TIORH.IOB[1:0] = 01/10/11
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000	MTUA.TOER.OE4B = 1
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4B = 1
		Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000	
		MTIOC4C	MTIOC4C	Normal operation
PWM mode 1	MTU4.TMDR.MD[3:0] = 0010			MTUA.TOER.OE4C = 1 MTU4.TMDR.BFA = 0 MTU4.TMDR.BFB = 0 MTU4.TIORL.IOC[1:0] = 01/10/11
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000			MTUA.TOER.OE4C = 1 MTU4.TMDR.BFA = 0
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101			
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110			
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111			
Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000			MTUA.TOER.OE4C = 1 MTU4.TMDR.BFA = 0
Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000			
Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000			
Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000			
MTIOC4D	MTIOC4D	Normal operation	MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4D = 1 MTU4.TMDR.BFB = 0 MTU4.TIORL.IOD[3] = 0 MTU4.TIORL.IOD[1:0] = 01/10/11
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000	MTUA.TOER.OE4D = 1
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4D = 1
		Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000	

Table 17.33 Settings to Enable Output on the Various MTU Pins (4 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins		
MTU6	MTIOC6A	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TIORH.IOA[3] = 0 MTU6.TIORH.IOA[1:0] = 01/10/11		
		PWM mode 1	MTU6.TMDR.MD[3:0] = 0010	MTU6.TIORH.IOA[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	Except MTU6.TCR.CCLR[2:0] = 001 MTU6.TIORH.IOA[1:0] = 01/10/11		
	MTIOC6B	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TIORH.IOB[3] = 0 MTU6.TIORH.IOB[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	Except MTU6.TCR.CCLR[2:0] = 010 MTU6.TIORH.IOB[1:0] = 01/10/11		
	MTIOC6C	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TMDR.BFA = 0 MTU6.TIORL.IOC[3] = 0 MTU6.TIORL.IOC[1:0] = 01/10/11		
		PWM mode 1	MTU6.TMDR.MD[3:0] = 0010	MTU6.TMDR.BFA = 0 MTU6.TMDR.BFB = 0 MTU6.TIORL.IOC[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	MTU6.TMDR.BFA = 0 Except MTU6.TCR.CCLR[2:0] = 101 MTU6.TIORL.IOC[1:0] = 01/10/11		
	MTIOC6D	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TMDR.BFB = 0 MTU6.TIORL.IOD[3] = 0 MTU6.TIORL.IOD[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	MTU6.TMDR.BFB = 0 Except MTU6.TCR.CCLR[2:0] = 110 MTU6.TIORL.IOD[1:0] = 01/10/11		
	MTU7	MTIOC7A	Normal operation	MTU7.TMDR.MD[3:0] = 0000	MTU7.TIOR.IOA[3] = 0 MTU7.TIOR.IOA[1:0] = 01/10/11	
			PWM mode 1	MTU7.TMDR.MD[3:0] = 0010	MTU7.TIOR.IOA[1:0] = 01/10/11	
PWM mode 2			MTU7.TMDR.MD[3:0] = 0011	Except MTU7.TCR.CCLR[1:0] = 01 MTU7.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 1			MTU7.TMDR.MD[3:0] = 0100	MTU7.TIOR.IOA[3] = 0 MTU7.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 2			MTU7.TMDR.MD[3:0] = 0101			
Phase count mode 3			MTU7.TMDR.MD[3:0] = 0110			
Phase count mode 4			MTU7.TMDR.MD[3:0] = 0111			
MTIOC7B		Normal operation	MTU7.TMDR.MD[3:0] = 0000	MTU7.TIOR.IOB[3] = 0 MTU7.TIOR.IOB[1:0] = 01/10/11		
		PWM mode 2	MTU7.TMDR.MD[3:0] = 0011	Except MTU7.TCR.CCLR[1:0] = 10 MTU7.TIOR.IOB[1:0] = 01/10/11		
		Phase count mode 1	MTU7.TMDR.MD[3:0] = 0100	MTU7.TIOR.IOB[3] = 0 MTU7.TIOR.IOB[1:0] = 01/10/11		
		Phase count mode 2	MTU7.TMDR.MD[3:0] = 0101			
		Phase count mode 3	MTU7.TMDR.MD[3:0] = 0110			
		Phase count mode 4	MTU7.TMDR.MD[3:0] = 0111			
		MTU8	MTIOC8A	Normal operation	MTU8.TMDR.MD[3:0] = 0000	MTU8.TIOR.IOA[3] = 0 MTU8.TIOR.IOA[1:0] = 01/10/11
				PWM mode 1	MTU8.TMDR.MD[3:0] = 0010	MTU8.TIOR.IOA[1:0] = 01/10/11
PWM mode 2	MTU8.TMDR.MD[3:0] = 0011			Except MTU8.TCR.CCLR[1:0] = 01 MTU8.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 1	MTU8.TMDR.MD[3:0] = 0100			MTU8.TIOR.IOA[3] = 0 MTU8.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 2	MTU8.TMDR.MD[3:0] = 0101					
Phase count mode 3	MTU8.TMDR.MD[3:0] = 0110					
Phase count mode 4	MTU8.TMDR.MD[3:0] = 0111					

Table 17.33 Settings to Enable Output on the Various MTU Pins (5 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins
MTU8	MTIOC8B	Normal operation	MTU8.TMDR.MD[3:0] = 0000	MTU8.TIOR.IOB[3] = 0 MTU8.TIOR.IOB[1:0] = 01/10/11
		PWM mode 2	MTU8.TMDR.MD[3:0] = 0011	Except MTU8.TCR.CCLR[1:0] = 10 MTU8.TIOR.IOB[1:0] = 01/10/11
		Phase count mode 1	MTU8.TMDR.MD[3:0] = 0100	MTU8.TIOR.IOB[3] = 0
		Phase count mode 2	MTU8.TMDR.MD[3:0] = 0101	MTU8.TIOR.IOB[1:0] = 01/10/11
		Phase count mode 3	MTU8.TMDR.MD[3:0] = 0110	
		Phase count mode 4	MTU8.TMDR.MD[3:0] = 0111	
MTU9	MTIOC9A	Normal operation	MTU9.TMDR.MD[3:0] = 0000	MTU9.TIORH.IOA[3] = 0 MTU9.TIORH.IOA[1:0] = 01/10/11
		PWM mode 1	MTU9.TMDR.MD[3:0] = 0010	MTU9.TIORH.IOA[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU9.TMDR.MD[3:0] = 1000	MTUB.TOER1.PSYE = 1
		Complementary PWM mode 1	MTU9.TMDR.MD[3:0] = 1101	
		Complementary PWM mode 2	MTU9.TMDR.MD[3:0] = 1110	
		Complementary PWM mode 3	MTU9.TMDR.MD[3:0] = 1111	
	MTIOC9B	Normal operation	MTU9.TMDR.MD[3:0] = 0000	MTUB.TOER.OE3B = 1 MTU9.TIORH.IOB[3] = 0 MTU9.TIORH.IOB[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU9.TMDR.MD[3:0] = 1000	MTUB.TOER.OE3B = 1
		Complementary PWM mode 1	MTU9.TMDR.MD[3:0] = 1101	
		Complementary PWM mode 2	MTU9.TMDR.MD[3:0] = 1110	
		Complementary PWM mode 3	MTU9.TMDR.MD[3:0] = 1111	
		MTIOC9C	Normal operation	MTU9.TMDR.MD[3:0] = 0000
	PWM mode 1		MTU9.TMDR.MD[3:0] = 0010	MTU9.TMDR.BFA = 0 MTU9.TMDR.BFB = 0 MTU9.TIORL.IOC[1:0] = 01/10/11
	MTIOC9D	Normal operation	MTU9.TMDR.MD[3:0] = 0000	MTUB.TOER.OE3D = 1 MTU9.TMDR.BFB = 0 MTU9.TIORL.IOD[3] = 0 MTU9.TIORL.IOD[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU9.TMDR.MD[3:0] = 1000	MTUB.TOER.OE3D = 1
Complementary PWM mode 1		MTU9.TMDR.MD[3:0] = 1101		
Complementary PWM mode 2		MTU9.TMDR.MD[3:0] = 1110		
Complementary PWM mode 3		MTU9.TMDR.MD[3:0] = 1111		
MTU10		MTIOC10A	Normal operation	MTU10.TMDR.MD[3:0] = 0000
	PWM mode 1		MTU10.TMDR.MD[3:0] = 0010	MTUB.TOER.OE4A = 1 MTU10.TIORH.IOA[1:0] = 01/10/11
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1000	MTUB.TOER.OE4A = 1
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1101	
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1110	
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1111	
	Reset-synchronized PWM mode (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4A = 1
	Complementary PWM mode 1 (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000	
	Complementary PWM mode 2 (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000	
	Complementary PWM mode 3 (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000	

Table 17.33 Settings to Enable Output on the Various MTU Pins (6 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins		
MTU10	MTIOC10B	Normal operation	MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4B = 1 MTU10.TIORH.IOB[3] = 0 MTU10.TIORH.IOB[1:0] = 01/10/11		
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1000	MTUB.TOER.OE4B = 1		
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1101			
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1110			
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1111			
		Reset-synchronized PWM mode (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4B = 1		
		Complementary PWM mode 1 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000			
		Complementary PWM mode 2 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000			
		Complementary PWM mode 3 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000			
		MTIOC10C	MTIOC10C	Normal operation	MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0 MTU10.TIORL.IOC[3] = 0 MTU10.TIORL.IOC[1:0] = 01/10/11
PWM mode 1	MTU10.TMDR.MD[3:0] = 0010			MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0 MTU10.TMDR.BFB = 0 MTU10.TIORL.IOC[1:0] = 01/10/11		
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1000			MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0		
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1101					
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1110					
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1111					
Reset-synchronized PWM mode (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000			MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0		
Complementary PWM mode 1 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000					
Complementary PWM mode 2 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000					
Complementary PWM mode 3 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000					
MTIOC10D	MTIOC10D			Normal operation	MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4D = 1 MTU10.TMDR.BFB = 0 MTU10.TIORL.IOD[3] = 0 MTU10.TIORL.IOD[1:0] = 01/10/11
				– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1000	MTUB.TOER.OE4D = 1
				– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1101	
				– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1111			
		Reset-synchronized PWM mode (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4D = 1		
		Complementary PWM mode 1 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000			
		Complementary PWM mode 2 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000			
		Complementary PWM mode 3 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000			

17.3.5 Treatment of Unused Pins

The treatment of unused pins is listed in Table 17.34.

Table 17.34 Treatment of Unused Pins (100-Pin LQFP)

Pin Name	Treatment
EMLE	Connect this pin to Vss via a pull-down resistor.
MD1, MD0	(Always used as a mode pin)
MDE	(Always used as mode pins)
RES#	Connect this pin to Vcc via a pull-up resistor.
USB0_DP	Leave these pins open.
USB0_DM	
P35/NMI	Connect this pin to Vcc via a pull-up resistor.
EXTAL	(Always used as a clock pin)
XTAL	Leave this pin open.
XCIN	Connect this pin to Vcc via a pull-up resistor or to Vss via a pull-down resistor.
XCOU	Leave this pin open.
Ports 0 to 5, and A to E	<ul style="list-style-type: none"> • Connect these pins to Vcc via a pull-up resistor or to Vss via a pull-down resistor • These pins can be left while PORTn.ICR is in the initial state (the input buffer disabled)*.
VREFH	Connect this pin to AVcc
VREFL	Connect this pin to AVSS

Note: * Do not change the initial value of PORTn.ICR. Changing the initial value may generate shoot-through current.

17.4 I/O Ports [for 85-Pin TFLGA]

17.4.1 Overview

The RX62N/RX621 Group (85-pin TFLGA) has 10 I/O ports (ports 0 to 5 and A to D), which handle 60 I/O pins.

Table 17.35 gives the specifications of the I/O ports and Table 17.36 lists I/O ports and pin functions.

Table 17.35 Specifications of I/O Ports (85-Pin TFLGA)

Item	Description
I/O pins	58
Input pins	2
Number of ports	10 (0 to 5, A to D)
Built-in input pull-up resistor	Ports A, B, C, D
Open drain outputs	Ports 0, 1, 2, 3 (P30 to P34), C
5-V tolerance	Port 1 (P12, P13, P16), port 2 (P20, P21), port 3 (P33)
Schmitt trigger input pins	All port inputs, CAN inputs, USB inputs, IRQ inputs, MTU inputs, TMR inputs, RIIC inputs, SCI inputs, and A/D trigger inputs
Others	<ul style="list-style-type: none"> • Each pin is capable of driving a capacitive load of 30 pF in the case of a TTL load. • When configured as an output, a pin is capable of driving a Darlington transistor.

Table 17.36 Port Functions (85-Pin TFLGA) (1 / 2)

Port	Description	Bit	Function			CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability
			I/O	Input	Output				
Port 0	General I/O port pins, interrupt inputs, and D/A converter outputs	3	P03	IRQ11-A	DA0	—	All input functions	—	√
		5	P05	IRQ13-A	DA1	—	All input functions	—	√
Port 1	General I/O port pins, USB I/O signals, MTU I/O signals, TMR I/O signals, interrupt inputs, SCI I/O signals, RIIC I/O signals, PPG I/O signals, and A/D converter inputs	2	P12/SCL0	TMC11/ RxD2-A/IRQ2-B		—	All input functions	—	√
		3	P13/MTIOC0B/ SDA0	ADTRG1#/IRQ3-B	PO13/TMO3 /TxD2-A	—	All input functions	—	√
		4	P14/MTIOC3A	USB0_OVRCURA/ TMR12/IRQ4-B	PO15/ USB0_DPUPE-B	—	All input functions	—	√
		6	P16/MTIOC3C	USB0_VBUS/ USB0_OVRCURB/ IRQ6-B	TMO2/PO14/ USB0_VBUSEN-B	—	All input functions	—	√
Port 2	General I/O port pins, bus control I/O signals, USB I/O signals, RSPI I/O signals, MTU I/O signals, PPG outputs, TMR I/O signals, SCI I/O signals, RIIC I/O signals, A/D converter inputs, and on-chip emulator I/O signals	0	P20/MTIOC1A/ SDA1	USB0_ID/ TMR10	PO0/TxD0	—	All input functions	—	√
		1	P21/MTIOC1B/ SCL1	TMC10/RxD0	USB0_EXICEN/ PO1	—	All input functions	—	√
		2	P22/MTIOC3B/ SCK0	MTCLKC	USB0_DRPD/ PO2/ TMO0	—	All input functions	—	√
		3	P23/MTIOC3D	MTCLKD	USB0_DPUPE-A/ TxD3-B/PO3	—	All input functions	—	√
		4	P24/MTIOC4A/ SCK3	MTCLKA-A/TMR11	CS4#/ USB0_VBUSEN- A/ PO4	—	All input functions	—	√
		5	P25/MTIOC4C	MTCLKB/ ADTRG0#/ RxD3	CS5#/ USB0_DPRPD/ PO5	—	All input functions	—	√
		6	P26/MOSIB/ MTIOC2A		CS6#/PO6/ TMO1/TxD1/TDO	MOSIB	P26, MTIOC2A	—	√
Port 3	General I/O port pins, CAN I/O signals, USB I/O signals, RSPI I/O signals, MTU I/O signals, TMR inputs, SCI I/O signals, interrupt inputs, PPG outputs, RTC outputs and on-chip emulator I/O signals	0	P30/MISOB/ MTIOC4B	TMRI3/RxD1/ IRQ0/TDI	PO8	MISOB	P30, MTIOC4B TMR13, RxD1, IRQ0, TDI	—	√
		1	P31/SSLB0/ MTIOC4D	TMC12/IRQ1/ TMS	PO9	SSLB0	P31, MTIOC4D TMC12, IRQ1, TMS	—	√
		2	P32/MTIOC0C	IRQ2-A	CTX0/TxD6/ PO10/RTCOUT	—	All input functions	—	√
		3	P33/MTIOC0D	CRX0/RxD6/ IRQ3-A	PO11	—	All input functions	—	√
		4	P34/MTIOC0A/ SCK6	TMC13/IRQ4-A /TRST#	PO12	—	All input functions	—	√
Port 4	General I/O port pins, interrupt inputs, and A/D converter inputs	5		P35/NMI		—	All input functions	—	—
		0	P40	AN0/IRQ8		—	P40, IRQ8	—	—
		1	P41	AN1/IRQ9		—	P41, IRQ9	—	—
		2	P42	AN2/IRQ10		—	P42, IRQ10	—	—
		3	P43	AN3/IRQ11-B		—	P43, IRQ11-B	—	—
		4	P44	AN4/IRQ12		—	P44, IRQ12	—	—
		5	P45	AN5/IRQ13-B		—	P45, IRQ13-B	—	—
6	P46	AN6/IRQ14		—	P46, IRQ14	—	—		
	7	P47	AN7/IRQ15		—	P47, IRQ15	—	—	

Table 17.36 Port Functions (85-Pin TFLGA) (2 / 2)

Port	Description	Bit	Function			CMOS Input	Schmitt Trigger Input	Input Pull-Up Resistor Function	Open Drain Output Capability	
			I/O	Input	Output					
Port 5	General I/O port pins, external bus clock outputs, bus control I/O signals, RSPI outputs, and SCI I/O signals	0	P50		WR0#/SSLB1/TxD2-B	—	All input functions	—	—	
		1	P51/SCK2	WAIT#	SSLB2	WAIT#	P51, SCK2			
		2	P52	RxD2-B	RD#/SSLB3	—	All input functions			
		3		P53	BCLK	—	All input functions			
Port A	General I/O port pins, address outputs, RSPI I/O signals, MTU I/O signals, and PPG outputs	0	PA0/MTIOC6A		A0/SSLA1/PO16	—	All input functions	√	—	
		1	PA1/MTIOC6B		A1/SSLA2/PO17	—	All input functions			
		2	PA2/MTIOC6C		A2/SSLA3/PO18	—	All input functions			
		3	PA3/MTIOC6D		A3/PO19	—	All input functions			
		4	PA4/SSLA0/MTIOC7A		A4/PO20	SSLA0	PA4, MTIOC7A			
		5	PA5/RSPCKA/MTIOC7B		A5/PO21	RSPCKA	PA5, MTIOC7B			
		6	PA6/MOSIA/MTIOC8A		A6/PO22	MOSIA	PA6, MTIOC8A			
		7	PA7/MISOA/MTIOC8B		A7/PO23	MISOA	PA7, MTIOC8B			
Port B	General I/O port pins, address outputs, MTU I/O signals, and PPG outputs	0	PB0/MTIOC9A		A8/PO24	—	All input functions	√	—	
		1	PB1/MTIOC9C		A9/PO25	—	All input functions			
		2	PB2/MTIOC9B	MTCLKG-B	A10/PO26	—	All input functions			
		3	PB3/MTIOC9D	MTCLKH-B	A11/PO27	—	All input functions			
		4	PB4/MTIOC10A	MTCLKE-B	A12/PO28	—	All input functions			
		5	PB5/MTIOC10C	MTCLKF-B	A13/PO29	—	All input functions			
		6	PB6/MTIOC10B		A14/PO30	—	All input functions			
		7	PB7/MTIOC10D		A15/PO31	—	All input functions			
Port C	General I/O port pins, address outputs, bus control outputs, MTU inputs, and SCI I/O signals	0	PC0	MTCLKG-A	A16	—	All input functions	√	√	
		1	PC1/SCK5	MTCLKH-A	A17	—	All input functions	√	√	
		2	PC2	MTCLKE-A/RxD5	A18	—	All input functions	√	√	
		3	PC3	MTCLKF-A	A19/TxD5	—	All input functions	√	√	
Port D	General I/O port pins, bidirectional data-bus lines, and MTU inputs	0	PD0/D0			D0	PD0	√	—	
		1	PD1/D1			D1	PD1			
		2	PD2/D2	MTIC11W			D2	PD2, MTIC11W		
		3	PD3/D3	MTIC11V			D3	PD3, MTIC11V		
		4	PD4/D4	MTIC11U			D4	PD4, MTIC11U		
		5	PD5/D5	MTIC5W			D5	PD5, MTIC5W		
		6	PD6/D6	MTIC5V			D6	PD6, MTIC5V		
		7	PD7/D7	MTIC5U			D7	PD7, MTIC5U		

17.4.2 Register Descriptions

Table 17.37 lists registers of I/O ports, and Table 17.38 lists valid bits in each register.

Table 17.37 Registers of I/O Ports (85-Pin TFLGA) (1 / 2)

Port Symbol	Register Name	Register Symbol	Value after Reset	Address	Access Size
PORT0	Data direction register	DDR	00h	0008 C000h	8
	Data register	DR	00h	0008 C020h	8
	Port register	PORT	Undefined	0008 C040h	8
	Input buffer control register	ICR	00h	0008 C060h	8
	Open drain control register	ODR	00h	0008 C080h	8
PORT1	Data direction register	DDR	00h	0008 C001h	8
	Data register	DR	00h	0008 C021h	8
	Port register	PORT	Undefined	0008 C041h	8
	Input buffer control register	ICR	00h	0008 C061h	8
	Open drain control register	ODR	00h	0008 C081h	8
PORT2	Data direction register	DDR	00h	0008 C002h	8
	Data register	DR	00h	0008 C022h	8
	Port register	PORT	Undefined	0008 C042h	8
	Input buffer control register	ICR	00h	0008 C062h	8
	Open drain control register	ODR	00h	0008 C082h	8
PORT3	Data direction register	DDR	00h	0008 C003h	8
	Data register	DR	00h	0008 C023h	8
	Port register	PORT	Undefined	0008 C043h	8
	Input buffer control register	ICR	00h	0008 C063h	8
	Open drain control register	ODR	00h	0008 C083h	8
PORT4	Data direction register	DDR	00h	0008 C004h	8
	Data register	DR	00h	0008 C024h	8
	Port register	PORT	Undefined	0008 C044h	8
	Input buffer control register	ICR	00h	0008 C064h	8
PORT5	Data direction register	DDR	00h	0008 C005h	8
	Data register	DR	00h	0008 C025h	8
	Port register	PORT	Undefined	0008 C045h	8
	Input buffer control register	ICR	00h	0008 C065h	8
PORTA	Data direction register	DDR	00h	0008 C00Ah	8
	Data register	DR	00h	0008 C02Ah	8
	Port register	PORT	Undefined	0008 C04Ah	8
	Input buffer control register	ICR	00h	0008 C06Ah	8
	Pull-up resistor control register	PCR	00h	0008 C0CAh	8
PORTB	Data direction register	DDR	00h	0008 C00Bh	8
	Data register	DR	00h	0008 C02Bh	8
	Port register	PORT	Undefined	0008 C04Bh	8
	Input buffer control register	ICR	00h	0008 C06Bh	8
	Pull-up resistor control register	PCR	00h	0008 C0CBh	8

Table 17.37 Registers of I/O Ports (85-Pin TFLGA) (2 / 2)

Port Symbol	Register Name	Register Symbol	Value after Reset	Address	Access Size
PORTC	Data direction register	DDR	00h	0008 C00Ch	8
	Data register	DR	00h	0008 C02Ch	8
	Port register	PORT	Undefined	0008 C04Ch	8
	Input buffer control register	ICR	00h	0008 C06Ch	8
	Open drain control register	ODR	00h	0008 C08Ch	8
	Pull-up resistor control register	PCR	00h	0008 C0CCh	8
PORTD	Data direction register	DDR	00h	0008 C00Dh	8
	Data register	DR	00h	0008 C02Dh	8
	Port register	PORT	Undefined	0008 C04Dh	8
	Input buffer control register	ICR	00h	0008 C06Dh	8
	Pull-up resistor control register	PCR	00h	0008 C0CDh	8
IOPORT	Port function control register 0	PF0CSE	00h	0008 C100h	8
	Port function control register 3	PF3BUS	00h	0008 C103h	8
	Port function control register 4	PF4BUS	00h	0008 C104h	8
	Port function control register 8	PF8IRQ	00h	0008 C108h	8
	Port function control register 9	PF9IRQ	00h	0008 C109h	8
	Port function control register D	PFDMTU	00h	0008 C10Dh	8
	Port function control register F	PFFSCI	00h	0008 C10Fh	8
	Port function control register G	PFGSPI	00h	0008 C110h	8
	Port function control register H	PFHSPI	00h	0008 C111h	8
	Port function control register J	PFJCAN	00h	0008 C113h	8
Port function control register K	PFKUSB	00h	0008 C114h	8	

Table 17.38 Valid Bits in Each Register (85-Pin TFLGA) (1 / 2)

Register Symbol	b7	b6	b5	b4	b3	b2	b1	b0
PORT0.DDR	x	x	√	x	√	x	x	x
PORT1.DDR	x	√	x	√	√	√	x	x
PORT2.DDR	√	√	√	√	√	√	√	√
PORT3.DDR	x	x	x	√	√	√	√	√
PORT4.DDR	√	√	√	√	√	√	√	√
PORT5.DDR	x	x	x	x	√	√	√	√
PORTA.DDR	√	√	√	√	√	√	√	√
PORTB.DDR	√	√	√	√	√	√	√	√
PORTC.DDR	x	x	x	x	√	√	√	√
PORTD.DDR	√	√	√	√	√	√	√	√
PORT0.DR	x	x	√	x	√	x	x	x
PORT1.DR	x	√	x	√	√	√	x	x
PORT2.DR	√	√	√	√	√	√	√	√
PORT3.DR	x	x	x	√	√	√	√	√
PORT4.DR	√	√	√	√	√	√	√	√
PORT5.DR	x	x	x	x	x	√	√	√
PORTA.DR	√	√	√	√	√	√	√	√
PORTB.DR	√	√	√	√	√	√	√	√
PORTC.DR	x	x	x	x	√	√	√	√
PORTD.DR	√	√	√	√	√	√	√	√
PORT0.PORT	x	x	√	x	√	x	x	x
PORT1.PORT	x	√	x	√	√	√	x	x
PORT2.PORT	√	√	√	√	√	√	√	√
PORT3.PORT	x	x	√	√	√	√	√	√
PORT4.PORT	√	√	√	√	√	√	√	√
PORT5.PORT	x	x	x	x	√	√	√	√
PORTA.PORT	√	√	√	√	√	√	√	√
PORTB.PORT	√	√	√	√	√	√	√	√
PORTC.PORT	x	x	x	x	√	√	√	√
PORTD.PORT	√	√	√	√	√	√	√	√
PORT0.ICR	x	x	√	x	√	x	x	x
PORT1.ICR	x	√	x	√	√	√	√	x
PORT2.ICR	√	√	√	√	√	√	√	√
PORT3.ICR	x	x	x	√	√	√	√	√
PORT4.ICR	√	√	√	√	√	√	√	√
PORT5.ICR	x	x	x	x	√	√	√	√
PORTA.ICR	√	√	√	√	√	√	√	√
PORTB.ICR	√	√	√	√	√	√	√	√
PORTC.ICR	x	x	x	x	√	√	√	√
PORTD.ICR	√	√	√	√	√	√	√	√

Table 17.38 Valid Bits in Each Register (85-Pin TFLGA) (2 / 2)

Register Symbol	b7	b6	b5	b4	b3	b2	b1	b0
PORT0.ODR	x	x	√	x	√	x	x	x
PORT1.ODR	x	√	x	√	√	√	x	x
PORT2.ODR	√	√	√	√	√	√	√	√
PORT3.ODR	x	x	x	√	√	√	√	√
PORTC.ODR	x	x	x	x	√	√	√	√
PORTA.PCR	√	√	√	√	√	√	√	√
PORTB.PCR	√	√	√	√	√	√	√	√
PORTC.PCR	x	x	x	x	√	√	√	√
PORTD.PCR	√	√	√	√	√	√	√	√
PORTF0CSE	√	√	√	√	x	x	x	x
IOPORT.PF3BUS	x	x	x	x	√	√	√	√
IOPORT.PF4BUS	√	√	√	√	√	√	√	√
IOPORT.PF8IRQ	x	x	√	x	√	x	x	x
IOPORT.PF9IRQ	x	√	x	√	√	√	x	x
IOPORT.PFDMTU	√	x	x	x	x	x	x	x
IOPORT.PFFSCI	x	x	x	x	x	√	x	x
IOPORT.PFGSPI	√	√	√	√	√	√	√	x
IOPORT.PFHSPI	√	√	√	√	√	√	√	x
IOPORT.PFJCAN	x	x	x	x	x	x	x	√
IOPORT.PFKUSB	x	x	x	√	√	√	√	√

17.4.2.1 Data Direction Register (DDR)

Addresses: PORT0.DDR 0008 C000h, PORT1.DDR 0008 C001h, PORT2.DDR 0008 C002h,
PORT3.DDR 0008 C003h, PORT4.DDR 0008 C004h, PORT5.DDR 0008 C005h,
PORTA.DDR 0008 C00Ah, PORTB.DDR 0008 C00Bh, PORTC.DDR 0008 C00Ch,
PORTD.DDR 0008 C00Dh

b7	b6	b5	b4	b3	b2	b1	b0
B7	B6	B5	B4	B3	B2	B1	B0

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

Note: Bits 7, 6, 4, and 2 to 0 are reserved in PORT0.DDR.
Bits 7, 5, 1, and 0 are reserved in PORT1.DDR.
The lower five bits are valid and the upper three bits are reserved in PORT3.DDR.
The lower four bits are valid and the upper four bits are reserved in PORT5.DDR.
The lower four bits are valid and the upper four bits are reserved in PORTC.DDR.
The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 I/O Select	0: An input pin	R/W
b1	B1	Pn1 I/O Select	1: An output pin	R/W
b2	B2	Pn2 I/O Select		R/W
b3	B3	Pn3 I/O Select		R/W
b4	B4	Pn4 I/O Select		R/W
b5	B5	Pn5 I/O Select		R/W
b6	B6	Pn6 I/O Select		R/W
b7	B7	Pn7 I/O Select		R/W

[Legend] n = 0 to 5, A to D

Each DDR is used to select the input or output direction for individual pins of the corresponding port that have been configured to function as general I/O pins.

Each bit of a PORTn.DDR (n = 0 to 5, A to D) corresponds to a pin of PORTn, and the settings can change from bit to bit. The PORT5.DDR.B3 bit selects P53 input or BCLK output. Setting the PORT5.DDR.B3 bit to 1 specifies output of the BCLK signal on the pin that would otherwise be P53. Operation as a general output is not selectable for this pin.

17.4.2.2 Data Register (DR)

Addresses: PORT0.DR 0008 C020h, PORT1.DR 0008 C021h, PORT2.DR 0008 C022h,
PORT3.DR 0008 C023h, PORT4.DR 0008 C024h, PORT5.DR 0008 C025h,
PORTA.DR 0008 C02Ah, PORTB.DR 0008 C02Bh, PORTC.DR 0008 C02Ch,
PORTD.DR 0008 C02Dh

b7	b6	b5	b4	b3	b2	b1	b0
B7	B6	B5	B4	B3	B2	B1	B0

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

Note: Bits 7, 6, 4, and 2 to 0 are reserved in PORT0.DR.
Bits 7, 5, 1, and 0 are reserved in PORT1.DR.
The lower five bits are valid and the upper three bits are reserved in PORT3.DR.
The lower three bits are valid and the upper five bits are reserved in PORT5.DR.
The lower four bits are valid and the upper four bits are reserved in PORTC.DR.
The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 Output Data Store	Output data are stored.	R/W
b1	B1	Pn1 Output Data Store		R/W
b2	B2	Pn2 Output Data Store		R/W
b3	B3	Pn3 Output Data Store		R/W
b4	B4	Pn4 Output Data Store		R/W
b5	B5	Pn5 Output Data Store		R/W
b6	B6	Pn6 Output Data Store		R/W
b7	B7	Pn7 Output Data Store		R/W

[Legend] n = 0 to 5, A to D

Each DR stores the output data from the individual pins of the corresponding port used as a general I/O port.
The output of the P53 pin is the BCLK signal and the value of the B3 bit in PORT5.DR does not affect the pin.

17.4.2.3 Port Register (PORT)

Addresses: PORT0.PORT 0008 C040h, PORT1.PORT 0008 C041h, PORT2.PORT 0008 C042h,
PORT3.PORT 0009 C043h, PORT4.PORT 0008 C044h, PORT5.PORT 0008 C045h,
PORTA.PORT 0008 C04Ah, PORTB.PORT 0008 C04Bh, PORTC.PORT 0008 C04Ch,
PORTD.PORT 0008 C04Dh

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

Note: Bits 7, 6, 4, and 2 to 0 are reserved in PORT0.PORT.
Bits 7, 5, 1, and 0 are reserved in PORT1.PORT.
The lower six bits are valid and the upper two bits are reserved in PORT3.PORT.
The lower four bits are valid and the upper four bits are reserved in PORT5.PORT.
The lower four bits are valid and the upper four bits are reserved in PORTC.PORT.
The reserved bits are read as 1 and cannot be modified.

Bit	Symbol	Bit Name	Description	R/W
b0	B0*	Pn0	Individual pin states of the corresponding port are reflected.	R
b1	B1*	Pn1		R
b2	B2*	Pn2		R
b3	B3*	Pn3		R
b4	B4*	Pn4		R
b5	B5*	Pn5		R
b6	B6*	Pn6		R
b7	B7*	Pn7		R

[Legend] n = 0 to 5, A to D

Note : * Before reading this register, set the corresponding bit in PORTn.ICR to 1. If this register is read with the corresponding bit in PORTn.ICR set to 0, the read value is undefined.

PORT reflects individual pin states of the corresponding port.

When a PORTn.PORT (n = 0 to 5, A to D) is read, the corresponding pin states are read out to here.

The NMI pin state is read out to the P35 bit.

17.4.2.4 Input Buffer Control Register (ICR)

Addresses: PORT0.ICR 0008 C060h, PORT1.ICR 0008 C061h, PORT2.ICR 0008 C062h,
PORT3.ICR 0008 C063h, PORT4.ICR 0008 C064h, PORT5.ICR 0008 C065h,
PORTA.ICR 0008 C06Ah, PORTB.ICR 0008 C06Bh, PORTC.ICR 0008 C06Ch,
PORTD.ICR 0008 C06Dh

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

Note: Bits 7, 6, 4, and 2 to 0 are reserved in PORT0.ICR.

Bits 7, 5, 1, and 0 are reserved in PORT1.ICR.

The lower five bits are valid and the upper three bits are reserved in PORT3.ICR.

The lower four bits are valid and the upper four bits are reserved in PORT5.ICR.

The lower four bits are valid and the upper four bits are reserved in PORTC.ICR.

The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0*	Pn0 Input Buffer Control	0: The input buffer for the corresponding pin is disabled.	R/W
b1	B1*	Pn1 Input Buffer Control	1: The input buffer for the corresponding pin is enabled.	R/W
b2	B2*	Pn2 Input Buffer Control		R/W
b3	B3*	Pn3 Input Buffer Control		R/W
b4	B4*	Pn4 Input Buffer Control		R/W
b5	B5*	Pn5 Input Buffer Control		R/W
b6	B6*	Pn6 Input Buffer Control		R/W
b7	B7*	Pn7 Input Buffer Control		R/W

[Legend] n = 0 to 5, A to D

Note : * For pins being used as input pins, set the corresponding bits to 1. Set the bits corresponding to pins that are not being used for their input functions or are being used as analog I/O pins to 0.

Each ICR controls the input buffers for the individual pins of the corresponding port.

Each bit of a PORTn.ICR (n = 0 to 5, A to D) corresponds to a pin of PORTn, and the settings can change from bit to bit.

When to be used as an input pin for the peripheral module, the input buffer for the corresponding pin should be enabled beforehand by setting the PORTn.ICR bit to 1. If this register is used as an input pin for the peripheral module while the PORTn.ICR bit is 0, the input signal to the peripheral module is fixed high.

Changes in the settings of a PORTn.ICR may generate edges internally, depending on the pin state. For this reason, change the settings of PORTn.ICR while the corresponding input pins are not in use. For example, in the case of IRQ_i (i = 0 to 15) inputs, change settings of the corresponding PORTn.ICR with interrupts disabled by clearing the IR flag in IRI (i = 64 to 79 ("i" shows an interrupt vector number)) of the interrupt control unit (ICU) to 0, and then enable the corresponding interrupts. If a change to a PORTn.ICR setting does generate an edge, negate the edge.

17.4.2.5 Open Drain Control Register (ODR)

Addresses: PORT0.ODR 0008 C080h, PORT1.ODR 0008 C081h, PORT2.ODR 0008 C082h,
PORT3.ODR 0008 C083h, PORTC.ODR 0008 C08Ch

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

Note: Bits 7, 6, 4, and 2 to 0 are reserved in PORT0.ODR.

Bits 7, 5, 1, and 0 are reserved in PORT1.ODR.

The lower five bits are valid and the upper three bits are reserved in PORT3.ODR.

The lower four bits are valid and the upper four bits are reserved in PORTC.ODR.

The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 Output Type Select	0: CMOS output pin	R/W
b1	B1	Pn1 Output Type Select	1: NMOS open-drain output pin	R/W
b2	B2	Pn2 Output Type Select		R/W
b3	B3	Pn3 Output Type Select		R/W
b4	B4	Pn4 Output Type Select		R/W
b5	B5	Pn5 Output Type Select		R/W
b6	B6	Pn6 Output Type Select		R/W
b7	B7	Pn7 Output Type Select		R/W

[Legend] n = 0 to 3, C

Each ODR is used to select an output type for the individual pins.

17.4.2.6 Pull-Up Resistor Control Register (PCR)

Addresses: PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTC.PCR 0008 C0CCh,
PORTD.PCR 0008 C0CDh

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Note: The lower four bits are valid and the upper four bits are reserved in PORTC.PCR.
The reserved bits are read as 0. The write value should always be 0.

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pn0 Input Pull-Up Resistor Control	0: Input pull-up resistor is off. 1: Input pull-up resistor is on.	R/W
b1	B1	Pn1 Input Pull-Up Resistor Control		R/W
b2	B2	Pn2 Input Pull-Up Resistor Control		R/W
b3	B3	Pn3 Input Pull-Up Resistor Control		R/W
b4	B4	Pn4 Input Pull-Up Resistor Control		R/W
b5	B5	Pn5 Input Pull-Up Resistor Control		R/W
b6	B6	Pn6 Input Pull-Up Resistor Control		R/W
b7	B7	Pn7 Input Pull-Up Resistor Control		R/W

[Legend] n = A to D

Each PCR controls enabled/disabled of input pull-up resistor for individual pins of the corresponding port.

When in input pin state, for the pins corresponding to bits where the value in PORTn.PCR is 1, input pull-up resistor is turned on. Table 17.39 summarizes the input pull-up resistor states.

Table 17.39 Input Pull-Up Resistor States (85-Pin TFLGA)

Port	Pin State	Reset or Hardware Standby Mode	In Other Operations
Port A	Port input	Disabled	
	Address output	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	Enabled/Disabled
Port B	Address output	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port C	Address output	Disabled	
	Peripheral module output	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled
Port D	Data I/O	Disabled	
	Port output	Disabled	
	Port input	Disabled	Enabled/Disabled

[Legend]

Disabled: Input pull-up resistor is always disabled.

Enabled/Disabled: Input pull-up resistor is enabled when the PORTm.PCR.Bj bit (m = 9, and A to E, j = 0 to 7) is set to 1, and disabled when the bit is cleared to 0.

17.4.2.7 Port Function Control Register 0 (PF0CSE)

Address: 0008 C100h

	b7	b6	b5	b4	b3	b2	b1	b0
	CS7E	CS6E	CS5E	CS4E	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	CS4E	CS4 Output Enable	0: Designated as an I/O port pin.	R/W
b5	CS5E	CS5 Output Enable	1: Designated as the CSn# output pin (n = 4 to 7)	R/W
b6	CS6E	CS6 Output Enable		R/W
b7	CS7E	CS7 Output Enable		R/W

PF0CSE enables or disables CSn# output.

CSnE Bit (CSn Output Enable) (n = 4 to 7)

Each bit enables or disables the corresponding CSn# output.

To output a CSn signal, set the corresponding CSnE bit in PF0CSE to 1.

17.4.2.8 Port Function Control Register 3 (PF3BUS)

Address: 0008 C103h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	A19E	A18E	A17E	A16E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	A16E	Address A16 Output Enable	0: A16 output is disabled. 1: A16 output is enabled.	R/W
b1	A17E	Address A17 Output Enable	0: A17 output is disabled. 1: A17 output is enabled.	R/W
b2	A18E	Address A18 Output Enable	0: A18 output is disabled. 1: A18 output is enabled.	R/W
b3	A19E	Address A19 Output Enable	0: A19 output is disabled. 1: A19 output is enabled.	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PF3BUS enables or disables address outputs.

AnE Bit (Address An Output Enable) (n = 16 to 19)

Each bit enables or disables an address output (An).

17.4.2.9 Port Function Control Register 4 (PF4BUS)

Address: 0008 C104h

b7	b6	b5	b4	b3	b2	b1	b0
A15E	A14E	A13E	A12E	A11E	A10E	ADREL[1:0]	
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ADRLE[1: 0]	Address Lower A9 to A0 Output Enable	b1 b0 0 0: A9 to A0 output is disabled. 0 1: A9 to A4 output is disabled, A3 to A0 output is enabled 1 0: A9 to A8 output is disabled, A7 to A0 output is enabled 1 1: A9 to A0 output is enabled	R/W
b2	A10E	Address A10 Output Enable	0: A10 output is disabled. 1: A10 output is enabled.	R/W
b3	A11E	Address A11 Output Enable	0: A11 output is disabled. 1: A11 output is enabled.	R/W
b4	A12E	Address A12 Output Enable	0: A12 output is disabled. 1: A12 output is enabled.	R/W
b5	A13E	Address A13 Output Enable	0: A13 output is disabled. 1: A13 output is enabled.	R/W
b6	A14E	Address A14 Output Enable	0: A14 output is disabled. 1: A14 output is enabled.	R/W
b7	A15E	Address A15 Output Enable	0: A15 output is disabled. 1: A15 output is enabled.	R/W

PF4BUS enables or disables address outputs.

ADRLE[1: 0] Bits (Address Lower A9 to A0 Output Enable)

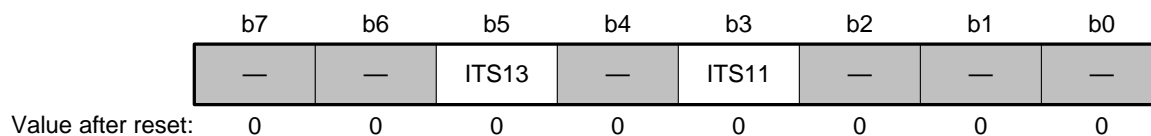
These bits enable or disable an address output (A9 to A0).

AnE Bit (Address An Output Enable) (n = 10 to 15)

Each bit enables or disables an address output (An).

17.4.2.10 Port Function Control Register 8 (PF8IRQ)

Address: 0008 C108h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b3	ITS11	IRQ11 Pin Select	0: P03 is designated as the IRQ11-A input pin. 1: P43 is designated as the IRQ11-B input pin.	R/W
b4	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b5	ITS13	IRQ13 Pin Select	0: P05 is designated as the IRQ13-A input pin. 1: P45 is designated as the IRQ13-B input pin.	R/W
b7, b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PF8IRQ is used to select pins for IRQ13 and IRQ15 inputs.

ITS_i Bit (IRQ_i Pin Select) (i = 11, 13)

Each bit selects a pin for an IRQ_i input.

17.4.2.11 Port Function Control Register 9 (PF9IRQ)

Address: 0008 C109h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ITS4	ITS3	ITS2	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b2	ITS2	IRQ2 Pin Select	0: P32 is designated as the IRQ2-A input pin. 1: P12 is designated as the IRQ2-B input pin.	R/W
b3	ITS3	IRQ3 Pin Select	0: P33 is designated as the IRQ3-A input pin. 1: P13 is designated as the IRQ3-B input pin.	R/W
b4	ITS4	IRQ4 Pin Select	0: P34 is designated as the IRQ4-A input pin. 1: P14 is designated as the IRQ4-B input pin.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PF9IRQ is used to select pins for IRQ2 to IRQ4 inputs.

ITS_i Bit (IRQ_i Pin Select) (i = 2 to 4)

Each bit selects a pin for an IRQ_i input.

17.4.2.12 Port Function Control Register D (PFDMTU)

Address: 0008 C10Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	TCLKS	—	—	—	—	—	—	—
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 always read as 0. The write value should always be 0.	R/W
b7	TCLKS	MTCLK Pin Select	0: PC2 is designated as the MTCLKE-A pin. PC3 is designated as the MTCLKF-A pin. PC0 is designated as the MTCLKG-A pin. PC1 is designated as the MTCLKH-A pin. 1: PB4 is designated as the MTCLKE-B pin. PB5 is designated as the MTCLKF-B pin. PB2 is designated as the MTCLKG-B pin. PB3 is designated as the MTCLKH-B pin.	R/W

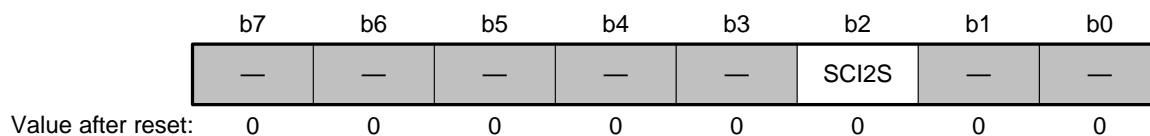
PFDMTU is used to select pins for MTU unit 1.

TCLKS Bit (MTCLK Pin Select)

This bit selects a pin for a MTCLK input of the MTU

17.4.2.13 Port Function Control Register F (PFFSCI)

Address: 0008 C10Fh



Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b2	SCI2S	SCI2 Pin Select	0: P12 is designated as the RxD2-A pin. P51 is designated as the SCK2 pin. P13 is designated as the TxD2-A pin. 1: P52 is designated as the RxD2-B pin. P51 is designated as the SCK2 pin. P50 is designated as the TxD2-B pin.	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PFFSCI is used to select pins for SCI.

SCInS Bit (SCIn Pin Select) (n = 2)

Each bit selects a pin for an SCI channel-n input/output.

17.4.2.14 Port Function Control Register G (PFGSPI)

Address: 0008 C110h

	b7	b6	b5	b4	b3	b2	b1	b0
	SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b1	RSPCKE	RSPCK A Output Enable	0: The RSPCKA pin is disabled. 1: The RSPCKA pin is enabled.	R/W
b2	MOSIE	MOSIA Output Enable	0: The MOSIA pin is disabled. 1: The MOSIA pin is enabled.	R/W
b3	MISOE	MISOA Output Enable	0: The MISOA pin is disabled. 1: The MISOA pin is enabled.	R/W
b4	SSL0E	SSLA0 Output Enable	0: The SSLA0 pin is disabled. 1: The SSLA0 pin is enabled.	R/W
b5	SSL1E	SSLA1 Output Enable	0: The SSLA1 pin is disabled. 1: The SSLA1 pin is enabled.	R/W
b6	SSL2E	SSLA2 Output Enable	0: The SSLA2 pin is disabled. 1: The SSLA2 pin is enabled.	R/W
b7	SSL3E	SSLA3 Output Enable	0: The SSLA3 pin is disabled. 1: The SSLA3 pin is enabled.	R/W

PFGSPI is used to select I/O pins for RSPI channel 0.

RSPCKE Bit (RSPCKA Output Enable)

This bit enables or disables the output of the RSPCKA pin. Set this bit to 1 to use the RSPCKA pin.

MOSIE Bit (MOSIA Output Enable)

This bit enables or disables the output of the MOSIA pin. Set this bit to 1 to use the MOSIA pin.

MISOE Bit (MISOA Output Enable)

This bit enables or disables the output of the MISOA pin. Set this bit to 1 to use the MISOA pin.

SSL0E Bit (SSLA0 Output Enable)

This bit enables or disables the output of the SSLA0 pin. Set this bit to 1 to use the SSLA0 pin.

SSL1E Bit (SSLA1 Output Enable)

This bit enables or disables the output of the SSLA1 pin. Set this bit to 1 to use the SSLA1 pin.

SSL2E Bit (SSLA2 Output Enable)

This bit enables or disables the output of the SSLA2 pin. Set this bit to 1 to use the SSLA2 pin.

SSL3E Bit (SSLA3 Output Enable)

This bit enables or disables the output of the SSLA3 pin. Set this bit to 1 to use the SSLA3 pin.

17.4.2.15 Port Function Control Register H (PFHSPI)

Address: 0008 C111h

	b7	b6	b5	b4	b3	b2	b1	b0
	SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b1	RSPCKE	RSPCKB Output Enable	0: The RSPCKB pin is disabled. 1: The RSPCKB pin is enabled	R/W
b2	MOSIE	MOSIB Output Enable	0: The MOSIB pin is disabled. 1: The MOSIB pin is enabled	R/W
b3	MISOE	MISOB Output Enable	0: The MISOB pin is disabled. 1: The MISOB pin is enabled	R/W
b4	SSL0E	SSLB0 Output Enable	0: The SSLB0 pin is disabled. 1: The SSLB0 pin is enabled	R/W
b5	SSL1E	SSLB1 Output Enable	0: The SSLB1 pin is disabled. 1: The SSLB1 pin is enabled	R/W
b6	SSL2E	SSLB2 Output Enable	0: The SSLB2 pin is disabled. 1: The SSLB2 pin is enabled	R/W
b7	SSL3E	SSLB3 Output Enable	0: The SSLB3 pin is disabled. 1: The SSLB3 pin is enabled	R/W

PFHSPI is used to select I/O pins for RSPI channel 1.

RSPCKE Bit (RSPCKB Output Enable)

This bit enables or disables the output of the RSPCKA pin. Set this bit to 1 to use the RSPCKA pin.

MOSIE Bit (MOSIB Output Enable)

This bit enables or disables the output of the MOSIA pin. Set this bit to 1 to use the MOSIA pin.

MISOE Bit (MISOB Output Enable)

This bit enables or disables the output of the MISOA pin. Set this bit to 1 to use the MISOA pin.

SSL0E Bit (SSLB0 Output Enable)

This bit enables or disables the output of the SSLA0 pin. Set this bit to 1 to use the SSLA0 pin.

SSL1E Bit (SSLB1 Output Enable)

This bit enables or disables the output of the SSLA1 pin. Set this bit to 1 to use the SSLA1 pin.

SSL2E Bit (SSLB2 Output Enable)

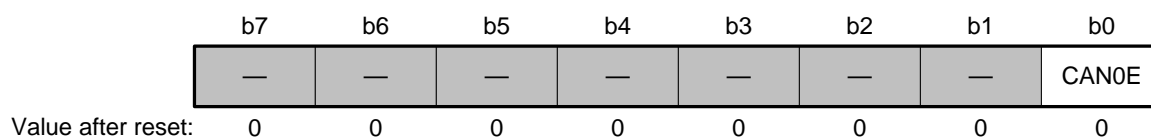
This bit enables or disables the output of the SSLA2 pin. Set this bit to 1 to use the SSLA2 pin.

SSL3E Bit (SSLB3 Output Enable)

This bit enables or disables the output of the SSLA3 pin. Set this bit to 1 to use the SSLA3 pin.

17.4.2.16 Port Function Control Register J (PFJCAN)

Address: 0008 C113h



Bit	Symbol	Bit Name	Description	R/W
b0	CAN0E	CAN0 Pins Enable	0: The CTX0 and CRX0 pins are disabled. 1: The CTX0 and CRX0 pins are enabled.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PFJCAN is used to select I/O pins for the CAN.

CANnE Bit (CANn Pins Enable) (n = 0)

This bit enables or disables the CANn pins. Set this bit to 1 to use the CANn pins.

17.4.2.17 Port Function Control Register K (PFKUSB)

Address: 0008 C114h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	USBE	PDHZS	PUPHZS	USBMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	USBMD[1:0]	USB Mode Setting	b1 b0 0 0: Select function mode for the USB0 pins. 0 1: Select host mode for the USB0 pins. 1 0: Select host/function mode for the USB0 pin. (as an optional function)* 1 1: Select OTG mode for the USB0 pins.	R/W
b2	PUPHZS	PUPHZ Select	0: USB0_DPUPE pin is for output of the high and low levels (external pull-up control signal). 1: USB0_DPUPE pin is for high-level output or the Hi-Z state (pull-up output is from the USB0_DP pin).	R/W
b3	PDHZS	PDHZ Select	0: USB0_DPRPD and USB0_DRPD pins are for output of the high and low levels (external pull-down control signals). 1: USB0_DPRPD and USB0_DRPD pins are for low-level output or the Hi-Z state (pull-down output is from the USB0_DP and USB0_DM pins).	R/W
b4	USBE	USB Enable	0: All pin functions for USB0 are disabled. 1: All pin functions for USB0 are enabled.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note : * Please contact a Renesas Electronics sales office for details of the optional function.

PFKUSB is used to set I/O pins for the USB0.

USBMD[1:0] Bits (USB Mode Setting)

These bits select a mode for the USB.

Table 17.40 lists the relationship between the setting of the USBMD[1:0] bits and the mode of the USB.

PUPHZS Bit (PUPHZ Select)

This bit selects the output mode (external pull-up control/pin pull-up output) for the DPUPE pin of the USB.

When the PUPHZS bit is set to 0, the control signal output mode for the external pull-up IC is selected and the high-active control signal is output from the DPUPE pin. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the low-level output state.

When the PUPHZS bit is set to 1, the output mode where the DP pin is directly pulled-up is selected. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the high-impedance state.

PDHVS Bit (PDHZ Select)

This bit selects the output mode (external pull-down control/pin pull-down output) for the DPRPD and DRPD pins of the USB.

When the PDHVS bit is set to 0, the control signal output mode for the external pull-down IC is selected and the high-active control signal is output from the DPRPD and DRPD pins. When the DP and DM pins are pulled down, the DPRPD and DRPD pins are set to the high-level output state. When the pulled-down state of the DP and DM pins is canceled, the DPRPD and DRPD pins are set to the low-level output state.

When the PDHVS bit is set to 1, the output mode where the DP and DM pins are directly pulled-down is selected. When the DP and DM pins are pulled down, the DPRPD and DRPD pins are set to the low-level output state. When the pulled-down state of the DP and DM pins is canceled, the DPRPD and DRPD pins are set to the high-impedance state.

USBE Bit (USB Enable)

This bit enables all pin functions for the USB0.

Table 17.40 Relationship between the USBMD[1:0] Bits Setting and USB Mode (USB0)

USBMD1	USBMD0	USB0 Mode	Pin to be Used for the USB	Pin Allocation	Remarks
0	0	Function mode	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_VBUS	P16	
			USB0_DPUPE-B	P14	Selection of -B side
0	1	Host mode	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_OVRCURA	P14	
			USB0_VBUSEN-B	P16	Selection of -B side
1	0	Host/function mode (as an optional function)*	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_OVRCURA	P14	
			USB0_VBUS	P16	
			USB0_DRPD	P22	
			USB0_DPUPE-A	P23	Selection of -A side
1	1	OTG mode	USB0_DP	USB0_DP	
			USB0_DM	USB0_DM	
			USB0_OVRCURA	P14	
			USB0_OVRCURB	P16	
			USB0_DPRPD	P25	
			USB0_DRPD	P22	
			USB0_EXICEN	P21	
			USB0_ID	P20	
USB0_DPUPE-A	P23	Selection of -A side			
			USB0_VBUSEN-A	P24	Selection of -A side

Note 1. Contact your Renesas sales representative regarding the optional function.

17.4.3 Settings of Ports


When individual pins for peripheral modules are enabled, the settings for each port are modified.

An input pin for the peripheral module is specified independently by the peripheral module. To use an input pin for the peripheral module, the corresponding bit in the input buffer control register (ICR) should be set to 1 to enable the input buffer, except for the port register read, data bus input, NMI, and POE pin inputs.

The pins that function as output pins and I/O pins should be enabled for respective peripheral modules. If a conflict occurs among the output signal enable settings for peripheral modules, that are multiplexed to the same port, the priority will be handled according to the port-multiplexed priority.

Table 17.41 lists the port-multiplexed priority for peripheral modules.

Table 17.41 Port-Multiplexed Priority for Peripheral Modules (85-Pin TFLGA)

Priority	Module Name	Output Pins	
High  Low	1	External bus (Data)	D0 to D7 (Data bus)
	2	External bus	RD#, WR0#, BCLK, A0 to A19 (Address bus)
	3	External bus (CS)	CS4# to CS7# (Chip select)
	4	RSPI0, RSPI1	RSPCKn, MOSIn, MISOn, SSLn0 to SSLn3 (n = A, B)
	5	USB0	USB0_DPUPE, USB0_VBUSEN, USB0_EXICEN, USB0_DRPD, USB0_DPRPD
	6	CAN0	CTX0
	7	MTU0 to MTU4, MTU6 to MTU10	MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D, MTIOC1A, MTIOC1B, MTIOC2A, MTIOC2B, MTIOC3A, MTIOC3B, MTIOC3C, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6A, MTIOC6B, MTIOC6C, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC8A, MTIOC8B, MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D, MTIOC10A, MTIOC10B, MTIOC10C, MTIOC10D
	8	TMR0 to TMR3	TMO0 to TMO3
	9	SCI0 to SCI3, SCI5 to SCI6	SCK0 to SCK3, SCK5 to SCK6, TxD0 to TxD3, TxD5 to TxD6
	10	RTC	RTCOUT
	11	PPG0, PPG1	PO0 to PO15, PO16 to PO31
	12	RIIC0, RIIC1	SCL0 to SCL1, SDA0 to SDA1
	13	DA	DA0, DA1
	14	I/O PORT	P03, P05, P12 to P14, P16, P20 to P27, P30 to P34, P50 to P52, PA0 to PA7, PB0 to PB7, PC0 to PC3, PD0 to PD7

17.4.4 List of Output Enable Settings

Table 17.42 lists the output enable settings for each port.

For details on the applicable output signals, see descriptions of the registers for each peripheral module.

Setting the port function control register changes the functions of peripheral-module pins with names ending in A to D.

Table 17.42 Output Enable Settings for Each Port (85-Pin TFLGA) (1 / 7)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P03	DA	DA0		DACR.DAOE0 = 1
	PORT0	P03		PORT0.DDR.B3 = 1
P05	DA	DA1		DACR.DAOE1 = 1
	PORT0	P05		PORT0.DDR.B5 = 1
P12	RIIC0	SCL0		RIIC0.ICCR1.ICE = 1
	PORT1	P12		PORT1.DDR.B2 = 1
P13	MTU0	MTIOC0B		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	TMR3	TMO3		TMO3.TCSR.OSA[1: 0] = 01/10/11 or TMO3.TCSR.OSB[1: 0] = 01/10/11
	SCI2	TxD2-A	PFFSCI.SCI2S = 0	SCI2.SCR.TE = 1
	PPG0	PO13		PPG0.NDERH.NDER13 = 1
	RIIC0	SDA0		RIIC0.ICCR1.ICE = 1
	PORT1	P13		PORT1.DDR.B3 = 1
P14	USB0	USB0_DPUPE-B	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 00	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3A		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO15		PPG0.NDERH.NDER15 = 1
	PORT1	P14		PORT1.DDR.B4 = 1
P16	USB0	USB0_VBUSEN-B	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 01	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3C		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	TMR2	TMO2		TMO2.TCSR.OSA[1: 0] = 01/10/11 or TMO2.TCSR.OSB[1: 0] = 01/10/11
	PPG0	PO14		PPG0.NDERH.NDER14 = 1
	PORT1	P16		PORT1.DDR.B6 = 1
P20	MTU1	MTIOC1A		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	SCI0	TxD0		SCI0.SCR.TE = 1
	PPG0	PO0		PPG0.NDERL.NDER0 = 1
	RIIC1	SDA1		RIIC1.ICCR1.ICE = 1
	PORT2	P20		PORT2.DDR.B0 = 1
P21	USB0	USB0_EXICEN	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 11	(The signal output state is specified by the peripheral module settings.)
	MTU1	MTIOC1B		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO1		PPG0.NDERL.NDER1 = 1
	RIIC1	SCL1		RIIC1.ICCR1.ICE = 1
	PORT2	P21		PORT2.DDR.B1 = 1

Table 17.42 Output Enable Settings for Each Port (85-Pin TFLGA) (2 / 7)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P22	USB0	USB0_DRPD	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 10/11	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3B		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	TMR0	TMO0		TMO0.TCSR.OSA[1: 0] = 01/10/11 or TMO0.TCSR.OSB[1: 0] = 01/10/11
	SCI0	SCK0		When SCI0.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI0.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO2		PPG0.NDERL.NDER2 = 1
	PORT2	P22		PORT2.DDR.B2 = 1
P23	USB0	USB0_DPUPE-A	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 10/11	(The signal output state is specified by the peripheral module settings.)
	MTU3	MTIOC3D		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	SCI3	TxD3		SCI3.SCR.TE = 1
	PPG0	PO3		PPG0.NDERL.NDER3 = 1
	PORT2	P23		PORT2.DDR.B3 = 1
P24	External bus (CS)	CS4#	PF0CSE.CS4E = 1	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	USB0	USB0_VBUSEN-A	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 10/11	(The signal output state is specified by the peripheral module settings.)
	MTU4	MTIOC4A		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	SCI3	SCK3		When SCI3.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI3.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO4		PPG0.NDERL.NDER4 = 1
	PORT2	P24		PORT2.DDR.B4 = 1
P25	External bus (CS)	CS5#	PF0CSE.CS5E = 1	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	USB0	USB0_DPRPD	PFKUSB.USBE = 1 PFKUSB.USBMD[1: 0] = 11	(The signal output state is specified by the peripheral module settings.)
	MTU4	MTIOC4C		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO5		PPG0.NDERL.NDER5 = 1
	PORT2	P25		PORT2.DDR.B5 = 1

Table 17.42 Output Enable Settings for Each Port (85-Pin TFLGA) (3 / 7)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P26	External bus (CS)	CS6#	PF0CSE.CS6E = 1	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSP11	MOSIB	PFHSPI.MOSIE = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU2	MTIOC2A		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	TMR1	TMO1		TMO1.TCSR.OSA[1: 0] = 01/10/11 or TMO1.TCSR.OSB[1: 0] = 01/10/11
	SCI1	TxD1		SCI1.SCR.TE = 1
	PPG0	PO6		PPG0.NDERL.NDER6 = 1
	PORT2	P26		PORT2.DDR.B6 = 1
P27	External bus (CS)	CS7#	PF0CSE.CS7E = 1	SYSCR0.EXBE = 1 (The external bus controller should also be set.)
	RSP11	RSPCKB	PFHSPI.RSPCKE = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU2	MTIOC2B		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	SCI1	SCK1		When SCI1.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI1.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO7		PPG0.NDERL.NDER7 = 1
	PORT2	P27		PORT2.DDR.B7 = 1
	P30	RSP11	MISOB	PFHSPI.MISOE = 1
MTU4		MTIOC4B		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
PPG0		PO8		PPG0.NDERH.NDER8 = 1
PORT3		P30		PORT3.DDR.B0 = 1
P31	RSP11	SSLB0	PFHSPI.SSL0E = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU4	MTIOC4D		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO9		PPG0.NDERH.NDER9 = 1
	PORT3	P31		PORT3.DDR.B1 = 1
P32	CAN0	CTX0	PFJCAN.CAN0E = 1	(The signal output state is specified by the peripheral module settings.)
	MTU0	MTIOC0C		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	SCI6	TxD6		SCI6.SCR.TE = 1
	RTC	RTCOUT		RCR2.RTCOE = 1
	PPG0	PO10		PPG0.NDERH.NDER10 = 1
	PORT3	P32		PORT3.DDR.B2 = 1

Table 17.42 Output Enable Settings for Each Port (85-Pin TFLGA) (4 / 7)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
P33	MTU0	MTIOC0D		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG0	PO11		PPG0.NDERH.NDER11 = 1
	PORT3	P33		PORT3.DDR.B3 = 1
P34	MTU0	MTIOC0A		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	SCI6	SCK6		When SCI6.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI6.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PPG0	PO12		PPG0.NDERH.NDER12 = 1
	PORT3	P34		PORT3.DDR.B4 = 1
P35	(NA)	(NA)		
P40	PORT4	P40		PORT4.DDR.B0 = 1
P41	PORT4	P41		PORT4.DDR.B1 = 1
P42	PORT4	P42		PORT4.DDR.B2 = 1
P43	PORT4	P43		PORT4.DDR.B3 = 1
P44	PORT4	P44		PORT4.DDR.B4 = 1
P45	PORT4	P45		PORT4.DDR.B5 = 1
P46	PORT4	P46		PORT4.DDR.B6 = 1
P47	PORT4	P47		PORT4.DDR.B7 = 1
P50	External bus	WR0#		SYSCR0.EXBE = 1
	RSPI1	SSLB1	PFHSPI.SSL1E = 1	(The signal output state is specified by the peripheral module settings.)
	SCI2	TxD2-B	PFFSCI.SCI2S = 1	SCI2.SCR.TE = 1
	PORT5	P50		PORT5.DDR.B0 = 1
P51	RSPI1	SSLB2	PFHSPI.SSL2E = 1	(The signal output state is specified by the peripheral module settings.)
	SCI2	SCK2		While SCI2.SCMR.SMIF = 1: SMR.GM = 0, SCR.CKE[1: 0] = 01 or SMR.GM = 1 When SCI2.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1: 0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PORT5	P51		PORT5.DDR.B1 = 1
P52	External bus	RD#	PF6BUS.MDSDE = 1	SYSCR0.EXBE = 1
	RSPI1	SSLB3	PFHSPI.SSL3E = 1	(The signal output state is specified by the peripheral module settings.)
	PORT5	P52		PORT5.DDR.B2 = 1
P53	External bus	BCLK		PORT5.DDR.B3 = 1
PA0	External bus	A0	PF4BUS.ADRLE[1: 0] = 01/10/11	SYSCR0.EXBE = 1
	RSPI0	SSLA1	PFGSPI.SSL1E = 1	(The signal output state is specified by the peripheral module settings.)
	MTU6	MTIOC6A		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO16		PPG1.NDERL.NDER0 = 1
	PORTA	PA0		PORTA.DDR.B0 = 1

Table 17.42 Output Enable Settings for Each Port (85-Pin TFLGA) (5 / 7)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PA1	External bus	A1	PF4BUS.ADRLE[1: 0] = 01/10/11	SYSCR0.EXBE = 1
	RSPI0	SSLA2	PFGSPI.SSL2E = 1	(The signal output state is specified by the peripheral module settings.)
	MTU6	MTIOC6B		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO17		PPG1.NDERL.NDER1 = 1
	PORTA	PA1		PORTA.DDR.B1 = 1
PA2	External bus	A2	PF4BUS.ADRLE[1: 0] = 01/10/11	SYSCR0.EXBE = 1
	RSPI0	SSLA3	PFGSPI.SSL3E = 1	(The signal output state is specified by the peripheral module settings.)
	MTU6	MTIOC6C		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO18		PPG1.NDERL.NDER2 = 1
	PORTA	PA2		PORTA.DDR.B2 = 1
PA3	External bus	A3	PF4BUS.ADRLE[1: 0] = 01/10/11	SYSCR0.EXBE = 1
	MTU6	MTIOC6D		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO19		PPG1.NDERL.NDER3 = 1
	PORTA	PA3		PORTA.DDR.B3 = 1
PA4	External bus	A4	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPI0	SSLA0	PFGSPI.SSL0E = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU7	MTIOC7A		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO20		PPG1.NDERL.NDER4 = 1
	PORTA	PA4		PORTA.DDR.B4 = 1
PA5	External bus	A5	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPI0	RSPCKA	PFGSPI.RSPCKE = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU7	MTIOC7B		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO21		PPG1.NDERL.NDER5 = 1
	PORTA	PA5		PORTA.DDR.B5 = 1
PA6	External bus	A6	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPI0	MOSIA	PFGSPI.MOSIE = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU8	MTIOC8A		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO22		PPG1.NDERL.NDER6 = 1
	PORTA	PA6		PORTA.DDR.B6 = 1

Table 17.42 Output Enable Settings for Each Port (85-Pin TFLGA) (6 / 7)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PA7	External bus	A7	PF4BUS.ADRLE[1: 0] = 10/11	SYSCR0.EXBE = 1
	RSPIO	MISOA	PFGSPI.MISOE = 1	(In addition to the pin enable setting, input/output switching function for the peripheral module is provided.)
	MTU8	MTIOC8B		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO23		PPG1.NDERL.NDER7 = 1
	PORTA	PA7		PORTA.DDR.B7 = 1
PB0	External bus	A8	PF4BUS.ADRLE[1: 0] = 11	SYSCR0.EXBE = 1
	MTU9	MTIOC9A		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO24		PPG1.NDERH.NDER8 = 1
	PORTB	PB0		PORTB.DDR.B0 = 1
PB1	External bus	A9	PF4BUS.ADRLE[1: 0] = 11	SYSCR0.EXBE = 1
	MTU9	MTIOC9C		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO25		PPG1.NDERH.NDER9 = 1
	PORTB	PB1		PORTB.DDR.B1 = 1
PB2	External bus	A10	PF4BUS.A10E = 1	SYSCR0.EXBE = 1
	MTU9	MTIOC9B		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO26		PPG1.NDERH.NDER10 = 1
	PORTB	PB2		PORTB.DDR.B2 = 1
PB3	External bus	A11	PF4BUS.A11E = 1	SYSCR0.EXBE = 1
	MTU9	MTIOC9D		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO27		PPG1.NDERH.NDER11 = 1
	PORTB	PB3		PORTB.DDR.B3 = 1
PB4	External bus	A12	PF4BUS.A12E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10A		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO28		PPG1.NDERH.NDER12 = 1
	PORTB	PB4		PORTB.DDR.B4 = 1
PB5	External bus	A13	PF4BUS.A13E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10C		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO29		PPG1.NDERH.NDER13 = 1
	PORTB	PB5		PORTB.DDR.B5 = 1
PB6	External bus	A14	PF4BUS.A14E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10B		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO30		PPG1.NDERH.NDER14 = 1
	PORTB	PB6		PORTB.DDR.B6 = 1
PB7	External bus	A15	PF4BUS.A15E = 1	SYSCR0.EXBE = 1
	MTU10	MTIOC10D		For the MTU settings, see Table 17.43, Settings to Enable Output on the Various MTU Pins.
	PPG1	PO31		PPG1.NDERH.NDER15 = 1
	PORTB	PB7		PORTB.DDR.B7 = 1

Table 17.42 Output Enable Settings for Each Port (85-Pin TFLGA) (7 / 7)

Port	Module Name	Output Signal Name	Port Function Register Settings	Peripheral Module Settings
PC0	External bus	A16	PF3BUS.A16E = 1	SYSCR0.EXBE = 1
	PORTC	PC0		PORTC.DDR.B0 = 1
PC1	External bus	A17	PF3BUS.A17E = 1	SYSCR0.EXBE = 1
	SCI5	SCK5		When SCI5.SCMR.SMIF = 1: While SMR.GM = 0, SCR.CKE[1:0] = 01 or SMR.GM = 1 When SCI5.SCMR.SMIF = 0: While SMR.CM = 0, SCR.CKE[1:0] = 01 or SMR.CM = 1, SCR.CKE[1] = 0
	PORTC	PC1		PORTC.DDR.B1 = 1
PC2	External bus	A18	PF3BUS.A18E = 1	SYSCR0.EXBE = 1
	PORTC	PC2		PORTC.DDR.B2 = 1
PC3	External bus	A19	PF3BUS.A19E = 1	SYSCR0.EXBE = 1
	SCI5	TxD5		SCI5.SCR.TE = 1
	PORTC	PC3		PORTC.DDR.B3 = 1
PD0	External bus (Data)	D0		SYSCR0.EXBE = 1
	PORTD	PD0		PORTD.DDR.B0 = 1
PD1	External bus (Data)	D1		SYSCR0.EXBE = 1
	PORTD	PD1		PORTD.DDR.B1 = 1
PD2	External bus (Data)	D2		SYSCR0.EXBE = 1
	PORTD	PD2		PORTD.DDR.B2 = 1
PD3	External bus (Data)	D3		SYSCR0.EXBE = 1
	PORTD	PD3		PORTD.DDR.B3 = 1
PD4	External bus (Data)	D4		SYSCR0.EXBE = 1
	PORTD	PD4		PORTD.DDR.B4 = 1
PD5	External bus (Data)	D5		SYSCR0.EXBE = 1
	PORTD	PD5		PORTD.DDR.B5 = 1
PD6	External bus (Data)	D6		SYSCR0.EXBE = 1
	PORTD	PD6		PORTD.DDR.B6 = 1
PD7	External bus (Data)	D7		SYSCR0.EXBE = 1
	PORTD	PD7		PORTD.DDR.B7 = 1

Table 17.43 Settings to Enable Output on the Various MTU Pins (1 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins	
MTU0	MTIOC0A	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TIORH.IOA[3] = 0 MTU0.TIORH.IOA[1:0] = 01/10/11	
		PWM mode 1	MTU0.TMDR.MD[3:0] = 0010	MTU0.TIORH.IOA[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	Except MTU0.TCR.CCLR[2:0] = 001 MTU0.TIORH.IOA[1:0] = 01/10/11	
	MTIOC0B	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TIORH.IOB[3] = 0 MTU0.TIORH.IOB[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	Except MTU0.TCR.CCLR[2:0] = 010 MTU0.TIORH.IOB[1:0] = 01/10/11	
	MTIOC0C	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TMDR.BFA = 0 MTU0.TIORL.IOC[3] = 0 MTU0.TIORL.IOC[1:0] = 01/10/11	
		PWM mode 1	MTU0.TMDR.MD[3:0] = 0010	MTU0.TMDR.BFA = 0 MTU0.TMDR.BFB = 0 MTU0.TIORL.IOC[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	MTU0.TMDR.BFA = 0 Except MTU0.TCR.CCLR[2:0] = 101 MTU0.TIORL.IOC[1:0] = 01/10/11	
	MTIOC0D	Normal operation	MTU0.TMDR.MD[3:0] = 0000	MTU0.TMDR.BFB = 0 MTU0.TIORL.IOD[3] = 0 MTU0.TIORL.IOD[1:0] = 01/10/11	
		PWM mode 2	MTU0.TMDR.MD[3:0] = 0011	MTU0.TMDR.BFB = 0 Except MTU0.TCR.CCLR[2:0] = 110 MTU0.TIORL.IOD[1:0] = 01/10/11	
	MTU1	MTIOC1A	Normal operation	MTU1.TMDR.MD[3:0] = 0000	MTU1.TIOR.IOA[3] = 0 MTU1.TIOR.IOA[1:0] = 01/10/11
			PWM mode 1	MTU1.TMDR.MD[3:0] = 0010	MTU1.TIOR.IOA[1:0] = 01/10/11
PWM mode 2			MTU1.TMDR.MD[3:0] = 0011	Except MTU1.TCR.CCLR[1:0] = 01 MTU1.TIOR.IOA[1:0] = 01/10/11	
Phase count mode 1			MTU1.TMDR.MD[3:0] = 0100	MTU1.TIOR.IOA[3] = 0 MTU1.TIOR.IOA[1:0] = 01/10/11	
Phase count mode 2			MTU1.TMDR.MD[3:0] = 0101		
Phase count mode 3			MTU1.TMDR.MD[3:0] = 0110		
Phase count mode 4			MTU1.TMDR.MD[3:0] = 0111		
MTIOC1B		Normal operation	MTU1.TMDR.MD[3:0] = 0000	MTU1.TIOR.IOB[3] = 0 MTU1.TIOR.IOB[1:0] = 01/10/11	
		PWM mode 2	MTU1.TMDR.MD[3:0] = 0011	Except MTU1.TCR.CCLR[1:0] = 10 MTU1.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 1	MTU1.TMDR.MD[3:0] = 0100	MTU1.TIOR.IOB[3] = 0 MTU1.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 2	MTU1.TMDR.MD[3:0] = 0101		
		Phase count mode 3	MTU1.TMDR.MD[3:0] = 0110		
		Phase count mode 4	MTU1.TMDR.MD[3:0] = 0111		
MTU2		MTIOC2A	Normal operation	MTU2.TMDR.MD[3:0] = 0000	MTU2.TIOR.IOA[3] = 0 MTU2.TIOR.IOA[1:0] = 01/10/11
			PWM mode 1	MTU2.TMDR.MD[3:0] = 0010	MTU2.TIOR.IOA[1:0] = 01/10/11
	PWM mode 2		MTU2.TMDR.MD[3:0] = 0011	Except MTU2.TCR.CCLR[1:0] = 01 MTU2.TIOR.IOA[1:0] = 01/10/11	
	Phase count mode 1		MTU2.TMDR.MD[3:0] = 0100	MTU2.TIOR.IOA[3] = 0 MTU2.TIOR.IOA[1:0] = 01/10/11	
	Phase count mode 2		MTU2.TMDR.MD[3:0] = 0101		
	Phase count mode 3		MTU2.TMDR.MD[3:0] = 0110		
	Phase count mode 4		MTU2.TMDR.MD[3:0] = 0111		

Table 17.43 Settings to Enable Output on the Various MTU Pins (2 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins	
MTU2	MTIOC2B	Normal operation	MTU2.TMDR.MD[3:0] = 0000	MTU2.TIOR.IOB[3] = 0 MTU2.TIOR.IOB[1:0] = 01/10/11	
		PWM mode 2	MTU2.TMDR.MD[3:0] = 0011	Except MTU2.TCR.CCLR[1:0] = 10 MTU2.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 1	MTU2.TMDR.MD[3:0] = 0100	MTU2.TIOR.IOB[3] = 0	
		Phase count mode 2	MTU2.TMDR.MD[3:0] = 0101	MTU2.TIOR.IOB[1:0] = 01/10/11	
		Phase count mode 3	MTU2.TMDR.MD[3:0] = 0110		
		Phase count mode 4	MTU2.TMDR.MD[3:0] = 0111		
MTU3	MTIOC3A	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTU3.TIORH.IOA[3] = 0 MTU3.TIORH.IOA[1:0] = 01/10/11	
		PWM mode 1	MTU3.TMDR.MD[3:0] = 0010	MTU3.TIORH.IOA[1:0] = 01/10/11	
		Reset-synchronized PWM mode	MTU3.TMDR.MD[3:0] = 1000	MTUA.TOCR1.PSYE = 1	
		Complementary PWM mode 1	MTU3.TMDR.MD[3:0] = 1101		
		Complementary PWM mode 2	MTU3.TMDR.MD[3:0] = 1110		
		Complementary PWM mode 3	MTU3.TMDR.MD[3:0] = 1111		
	MTIOC3B	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTUA.TOER.OE3B = 1 MTU3.TIORH.IOB[3] = 0 MTU3.TIORH.IOB[1:0] = 01/10/11	
		Reset-synchronized PWM mode	MTU3.TMDR.MD[3:0] = 1000	MTUA.TOER.OE3B = 1	
		Complementary PWM mode 1	MTU3.TMDR.MD[3:0] = 1101		
		Complementary PWM mode 2	MTU3.TMDR.MD[3:0] = 1110		
		Complementary PWM mode 3	MTU3.TMDR.MD[3:0] = 1111		
		MTIOC3C	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTU3.TMDR.BFA = 0 MTU3.TIORL.IOC[3] = 0 MTU3.TIORL.IOC[1:0] = 01/10/11
PWM mode 1	MTU3.TMDR.MD[3:0] = 0010		MTU3.TMDR.BFA = 0 MTU3.TMDR.BFB = 0 MTU3.TIORL.IOC[1:0] = 01/10/11		
MTIOC3D	Normal operation	Normal operation	MTU3.TMDR.MD[3:0] = 0000	MTUA.TOER.OE3D = 1 MTU3.TMDR.BFB = 0 MTU3.TIORL.IOD[3] = 0 MTU3.TIORL.IOD[1:0] = 01/10/11	
		Reset-synchronized PWM mode	MTU3.TMDR.MD[3:0] = 1000	MTUA.TOER.OE3D = 1	
		Complementary PWM mode 1	MTU3.TMDR.MD[3:0] = 1101		
		Complementary PWM mode 2	MTU3.TMDR.MD[3:0] = 1110		
		Complementary PWM mode 3	MTU3.TMDR.MD[3:0] = 1111		
		MTU4	MTIOC4A	Normal operation	MTU4.TMDR.MD[3:0] = 0000
PWM mode 1	MTU4.TMDR.MD[3:0] = 0010			MTUA.TOER.OE4A = 1 MTU4.TIORH.IOA[1:0] = 01/10/11	
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000			MTUA.TOER.OE4A = 1	
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101				
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110				
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111				
	Reset-synchronized PWM mode (linked operation with MTU3)		Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4A = 1
			Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000	
			Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000	
			Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000	

Table 17.43 Settings to Enable Output on the Various MTU Pins (3 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins
MTU4	MTIOC4B	Normal operation	MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4B = 1 MTU4.TIORH.IOB[3] = 0 MTU4.TIORH.IOB[1:0] = 01/10/11
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000	MTUA.TOER.OE4B = 1
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4B = 1
		Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000	
		MTIOC4C	MTIOC4C	Normal operation
PWM mode 1	MTU4.TMDR.MD[3:0] = 0010			MTUA.TOER.OE4C = 1 MTU4.TMDR.BFA = 0 MTU4.TMDR.BFB = 0 MTU4.TIORL.IOC[1:0] = 01/10/11
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000			MTUA.TOER.OE4C = 1 MTU4.TMDR.BFA = 0
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101			
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110			
– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111			
Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000			MTUA.TOER.OE4C = 1 MTU4.TMDR.BFA = 0
Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000			
Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000			
Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000			
MTIOC4D	MTIOC4D	Normal operation	MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4D = 1 MTU4.TMDR.BFB = 0 MTU4.TIORL.IOD[3] = 0 MTU4.TIORL.IOD[1:0] = 01/10/11
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1000	MTUA.TOER.OE4D = 1
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU4.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1000 MTU4.TMDR.MD[3:0] = 0000	MTUA.TOER.OE4D = 1
		Complementary PWM mode 1 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1101 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1110 MTU4.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU3)	MTU3.TMDR.MD[3:0] = 1111 MTU4.TMDR.MD[3:0] = 0000	

Table 17.43 Settings to Enable Output on the Various MTU Pins (4 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins		
MTU6	MTIOC6A	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TIORH.IOA[3] = 0 MTU6.TIORH.IOA[1:0] = 01/10/11		
		PWM mode 1	MTU6.TMDR.MD[3:0] = 0010	MTU6.TIORH.IOA[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	Except MTU6.TCR.CCLR[2:0] = 001 MTU6.TIORH.IOA[1:0] = 01/10/11		
	MTIOC6B	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TIORH.IOB[3] = 0 MTU6.TIORH.IOB[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	Except MTU6.TCR.CCLR[2:0] = 010 MTU6.TIORH.IOB[1:0] = 01/10/11		
	MTIOC6C	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TMDR.BFA = 0 MTU6.TIORL.IOC[3] = 0 MTU6.TIORL.IOC[1:0] = 01/10/11		
		PWM mode 1	MTU6.TMDR.MD[3:0] = 0010	MTU6.TMDR.BFA = 0 MTU6.TMDR.BFB = 0 MTU6.TIORL.IOC[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	MTU6.TMDR.BFA = 0 Except MTU6.TCR.CCLR[2:0] = 101 MTU6.TIORL.IOC[1:0] = 01/10/11		
	MTIOC6D	Normal operation	MTU6.TMDR.MD[3:0] = 0000	MTU6.TMDR.BFB = 0 MTU6.TIORL.IOD[3] = 0 MTU6.TIORL.IOD[1:0] = 01/10/11		
		PWM mode 2	MTU6.TMDR.MD[3:0] = 0011	MTU6.TMDR.BFB = 0 Except MTU6.TCR.CCLR[2:0] = 110 MTU6.TIORL.IOD[1:0] = 01/10/11		
	MTU7	MTIOC7A	Normal operation	MTU7.TMDR.MD[3:0] = 0000	MTU7.TIOR.IOA[3] = 0 MTU7.TIOR.IOA[1:0] = 01/10/11	
			PWM mode 1	MTU7.TMDR.MD[3:0] = 0010	MTU7.TIOR.IOA[1:0] = 01/10/11	
PWM mode 2			MTU7.TMDR.MD[3:0] = 0011	Except MTU7.TCR.CCLR[1:0] = 01 MTU7.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 1			MTU7.TMDR.MD[3:0] = 0100	MTU7.TIOR.IOA[3] = 0 MTU7.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 2			MTU7.TMDR.MD[3:0] = 0101			
Phase count mode 3			MTU7.TMDR.MD[3:0] = 0110			
Phase count mode 4			MTU7.TMDR.MD[3:0] = 0111			
MTIOC7B		Normal operation	MTU7.TMDR.MD[3:0] = 0000	MTU7.TIOR.IOB[3] = 0 MTU7.TIOR.IOB[1:0] = 01/10/11		
		PWM mode 2	MTU7.TMDR.MD[3:0] = 0011	Except MTU7.TCR.CCLR[1:0] = 10 MTU7.TIOR.IOB[1:0] = 01/10/11		
		Phase count mode 1	MTU7.TMDR.MD[3:0] = 0100	MTU7.TIOR.IOB[3] = 0 MTU7.TIOR.IOB[1:0] = 01/10/11		
		Phase count mode 2	MTU7.TMDR.MD[3:0] = 0101			
		Phase count mode 3	MTU7.TMDR.MD[3:0] = 0110			
		Phase count mode 4	MTU7.TMDR.MD[3:0] = 0111			
		MTU8	MTIOC8A	Normal operation	MTU8.TMDR.MD[3:0] = 0000	MTU8.TIOR.IOA[3] = 0 MTU8.TIOR.IOA[1:0] = 01/10/11
				PWM mode 1	MTU8.TMDR.MD[3:0] = 0010	MTU8.TIOR.IOA[1:0] = 01/10/11
PWM mode 2	MTU8.TMDR.MD[3:0] = 0011			Except MTU8.TCR.CCLR[1:0] = 01 MTU8.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 1	MTU8.TMDR.MD[3:0] = 0100			MTU8.TIOR.IOA[3] = 0 MTU8.TIOR.IOA[1:0] = 01/10/11		
Phase count mode 2	MTU8.TMDR.MD[3:0] = 0101					
Phase count mode 3	MTU8.TMDR.MD[3:0] = 0110					
Phase count mode 4	MTU8.TMDR.MD[3:0] = 0111					

Table 17.43 Settings to Enable Output on the Various MTU Pins (5 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins
MTU8	MTIOC8B	Normal operation	MTU8.TMDR.MD[3:0] = 0000	MTU8.TIOR.IOB[3] = 0 MTU8.TIOR.IOB[1:0] = 01/10/11
		PWM mode 2	MTU8.TMDR.MD[3:0] = 0011	Except MTU8.TCR.CCLR[1:0] = 10 MTU8.TIOR.IOB[1:0] = 01/10/11
		Phase count mode 1	MTU8.TMDR.MD[3:0] = 0100	MTU8.TIOR.IOB[3] = 0
		Phase count mode 2	MTU8.TMDR.MD[3:0] = 0101	MTU8.TIOR.IOB[1:0] = 01/10/11
		Phase count mode 3	MTU8.TMDR.MD[3:0] = 0110	
		Phase count mode 4	MTU8.TMDR.MD[3:0] = 0111	
MTU9	MTIOC9A	Normal operation	MTU9.TMDR.MD[3:0] = 0000	MTU9.TIORH.IOA[3] = 0 MTU9.TIORH.IOA[1:0] = 01/10/11
		PWM mode 1	MTU9.TMDR.MD[3:0] = 0010	MTU9.TIORH.IOA[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU9.TMDR.MD[3:0] = 1000	MTUB.TOER1.PSYE = 1
		Complementary PWM mode 1	MTU9.TMDR.MD[3:0] = 1101	
		Complementary PWM mode 2	MTU9.TMDR.MD[3:0] = 1110	
		Complementary PWM mode 3	MTU9.TMDR.MD[3:0] = 1111	
	MTIOC9B	Normal operation	MTU9.TMDR.MD[3:0] = 0000	MTUB.TOER.OE3B = 1 MTU9.TIORH.IOB[3] = 0 MTU9.TIORH.IOB[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU9.TMDR.MD[3:0] = 1000	MTUB.TOER.OE3B = 1
		Complementary PWM mode 1	MTU9.TMDR.MD[3:0] = 1101	
		Complementary PWM mode 2	MTU9.TMDR.MD[3:0] = 1110	
		Complementary PWM mode 3	MTU9.TMDR.MD[3:0] = 1111	
		MTIOC9C	Normal operation	MTU9.TMDR.MD[3:0] = 0000
	PWM mode 1		MTU9.TMDR.MD[3:0] = 0010	MTU9.TMDR.BFA = 0 MTU9.TMDR.BFB = 0 MTU9.TIORL.IOC[1:0] = 01/10/11
	MTIOC9D	Normal operation	MTU9.TMDR.MD[3:0] = 0000	MTUB.TOER.OE3D = 1 MTU9.TMDR.BFB = 0 MTU9.TIORL.IOD[3] = 0 MTU9.TIORL.IOD[1:0] = 01/10/11
		Reset-synchronized PWM mode	MTU9.TMDR.MD[3:0] = 1000	MTUB.TOER.OE3D = 1
Complementary PWM mode 1		MTU9.TMDR.MD[3:0] = 1101		
Complementary PWM mode 2		MTU9.TMDR.MD[3:0] = 1110		
Complementary PWM mode 3		MTU9.TMDR.MD[3:0] = 1111		
MTU10		MTIOC10A	Normal operation	MTU10.TMDR.MD[3:0] = 0000
	PWM mode 1		MTU10.TMDR.MD[3:0] = 0010	MTUB.TOER.OE4A = 1 MTU10.TIORH.IOA[1:0] = 01/10/11
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1000	MTUB.TOER.OE4A = 1
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1101	
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1110	
	– (setting prohibited)		MTU10.TMDR.MD[3:0] = 1111	
	Reset-synchronized PWM mode (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4A = 1
	Complementary PWM mode 1 (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000	
	Complementary PWM mode 2 (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000	
	Complementary PWM mode 3 (linked operation with MTU9)		MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000	

Table 17.43 Settings to Enable Output on the Various MTU Pins (6 / 6)

Channel	Pin Name	Operating Mode	Mode Select Bit	Settings to Enable Output on the Various Pins
MTU10	MTIOC10B	Normal operation	MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4B = 1 MTU10.TIORH.IOB[3] = 0 MTU10.TIORH.IOB[1:0] = 01/10/11
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1000	MTUB.TOER.OE4B = 1
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4B = 1
		Complementary PWM mode 1 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000	
		MTIOC10C	MTIOC10C	Normal operation
PWM mode 1	MTU10.TMDR.MD[3:0] = 0010			MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0 MTU10.TMDR.BFB = 0 MTU10.TIORL.IOC[1:0] = 01/10/11
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1000			MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1101			
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1110			
– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1111			
Reset-synchronized PWM mode (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000			MTUB.TOER.OE4C = 1 MTU10.TMDR.BFA = 0
Complementary PWM mode 1 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000			
Complementary PWM mode 2 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000			
Complementary PWM mode 3 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000			
MTIOC10D	MTIOC10D	Normal operation	MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4D = 1 MTU10.TMDR.BFB = 0 MTU10.TIORL.IOD[3] = 0 MTU10.TIORL.IOD[1:0] = 01/10/11
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1000	MTUB.TOER.OE4D = 1
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1101	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1110	
		– (setting prohibited)	MTU10.TMDR.MD[3:0] = 1111	
		Reset-synchronized PWM mode (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1000 MTU10.TMDR.MD[3:0] = 0000	MTUB.TOER.OE4D = 1
		Complementary PWM mode 1 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1101 MTU10.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 2 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1110 MTU10.TMDR.MD[3:0] = 0000	
		Complementary PWM mode 3 (linked operation with MTU9)	MTU9.TMDR.MD[3:0] = 1111 MTU10.TMDR.MD[3:0] = 0000	

17.4.5 Treatment of Unused Pins

The treatment of unused pins is listed in Table 17.44.

Table 17.44 Treatment of Unused Pins (85-Pin TFLGA)

Pin Name	Treatment
EMLE	Connect this pin to Vss via a pull-down resistor.
MD1, MD0	(Always used as a mode pin)
MDE	(Always used as mode pins)
RES#	Connect this pin to Vcc via a pull-up resistor.
USB0_DP	Leave these pins open.
USB0_DM	
BSCANP	Connect this pin to Vss via a pull-down resistor.
P35/NMI	Connect this pin to Vcc via a pull-up resistor.
EXTAL	(Always used as a clock pin)
XTAL	Leave this pin open.
XCIN	Connect this pin to Vcc via a pull-up resistor or to Vss via a pull-down resistor.
XCOU	Leave this pin open.
Ports 0 to 5, and A to D	<ul style="list-style-type: none"> ▪ Connect these pins to Vcc via a pull-up resistor or to Vss via a pull-down resistor ▪ These pins can be left while PORTn.ICR is in the initial state (the input buffer disabled)*.
VREFH	Connect this pin to AVcc
VREFL	Connect this pin to AVSS

Note: * Do not change the initial value of PORTn.ICR. Changing the initial value may generate shoot-through current.

17.5 Configuration of I/O Port

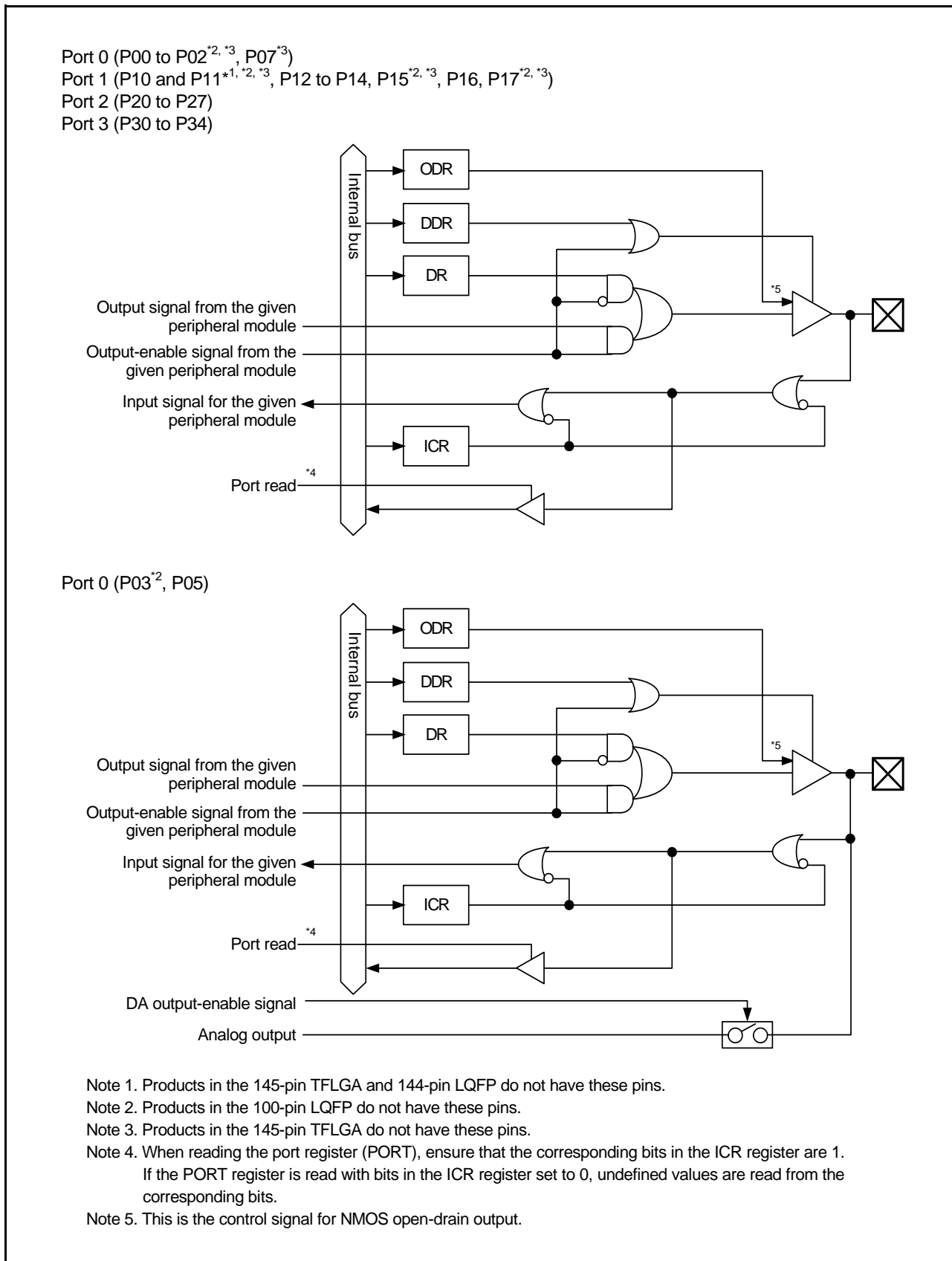


Figure 17.1 Configuration of I/O Port (1)

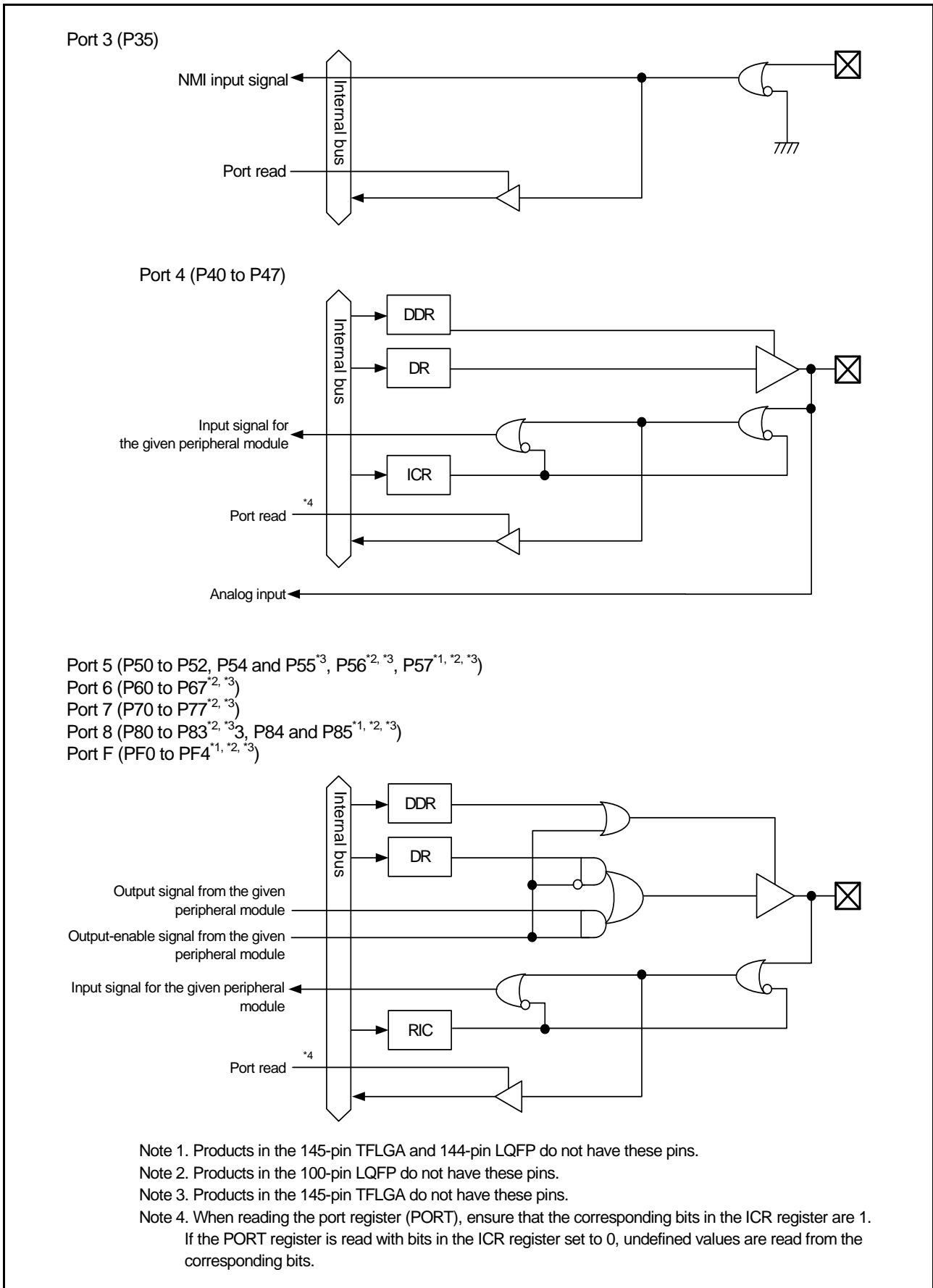


Figure 17.2 Configuration of I/O Port (2)

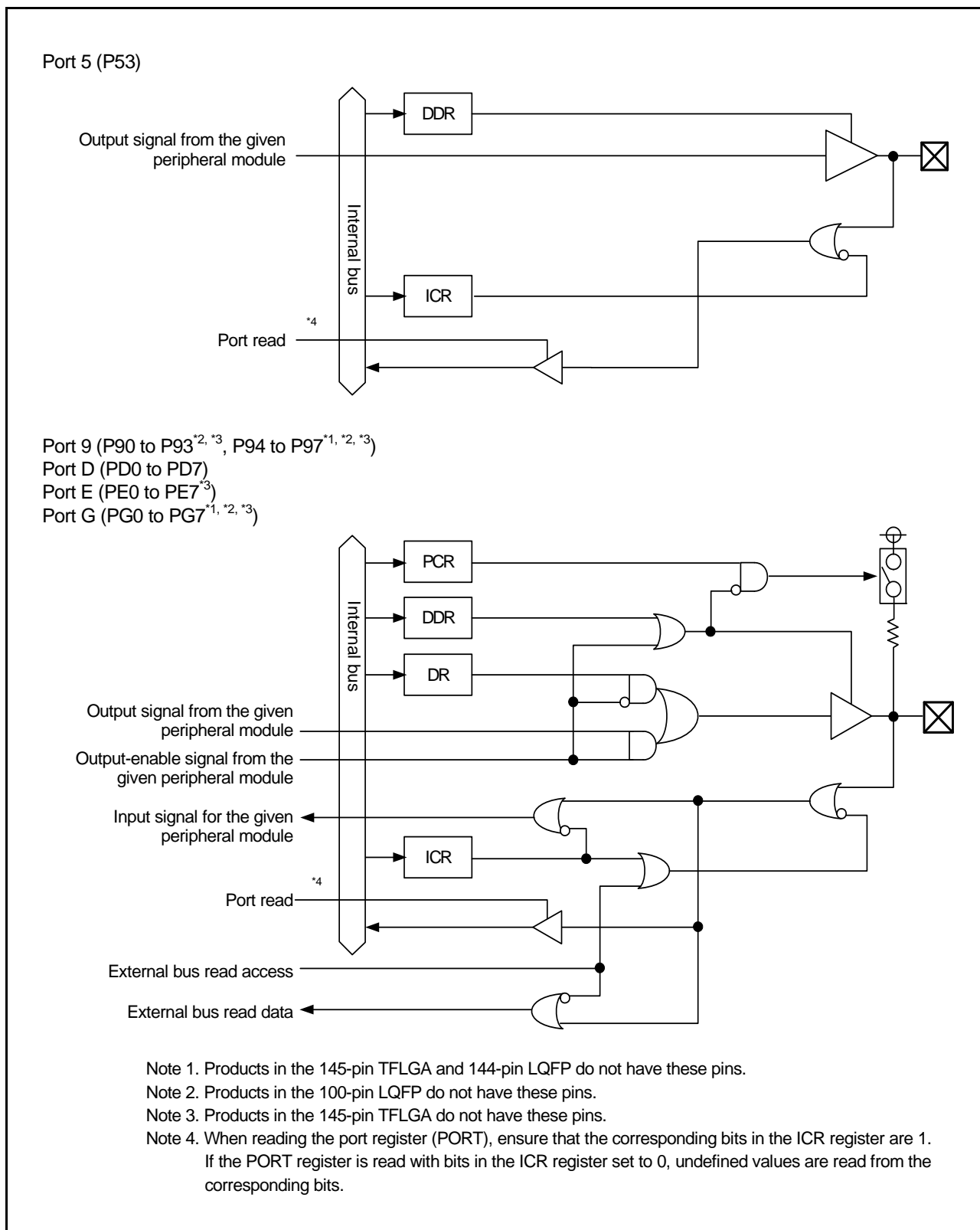


Figure 17.3 Configuration of I/O Port (3)

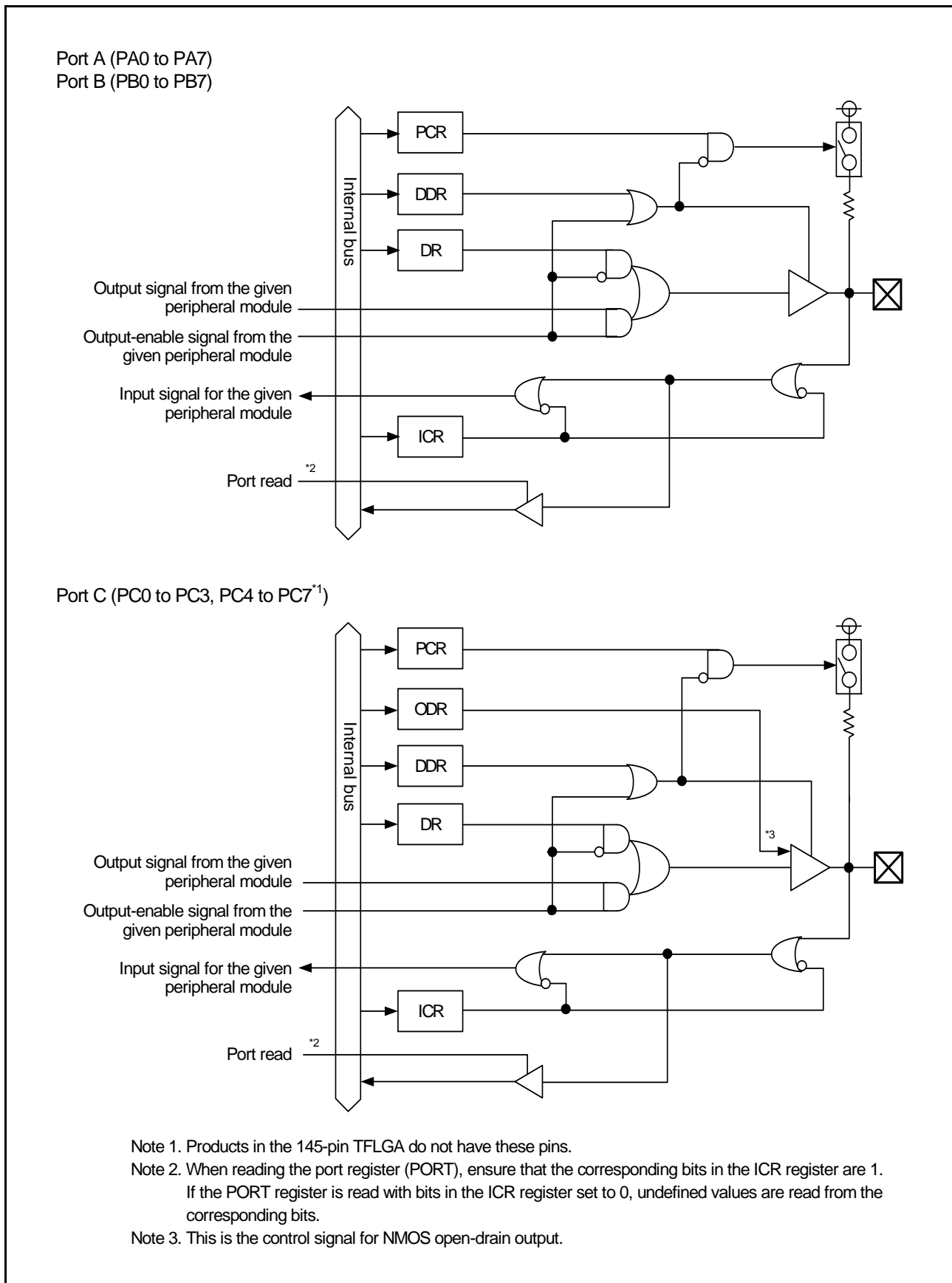


Figure 17.4 Configuration of I/O Port (4)

17.6 Usage Notes

17.6.1 Setting the Input Buffer Control Register (PORTn.ICR)

Changes to PORTn.ICR settings can lead to the generation of internal edges, depending on the setting and the pin states at the time it is made. This may lead to operation that was not intended. Change the setting of a PORTn.ICR while the input signal from the pins is fixed to the high level or, if a peripheral function has been assigned to a pin, while the input function is disabled.

If a PORTn.ICR setting enables an input buffer, and several input functions have been allocated to the corresponding pin, the pin state is reflected in the values of all of the input signals. Thus, even for input functions that are not in use, attention must be paid to the settings of the corresponding peripheral modules.

If a pin is being used as an output pin, the output value is taken in as the pin state when the input buffer is enabled by a PORTn.ICR setting. For pins being used as output pins, disable the input buffer by setting the corresponding bits in the given PORTn.ICR.

17.6.2 Setting the Port Function Control Register (PFCRm)

Each port function control register controls an I/O port. When setting input or output functions for individual pins, select a pin for the input or output and then enable or disable the input or output function.

If the levels for a pin before and after it has been switched to operate as an input differ from each other, an internal edge is generated. This may lead to operation that was not intended. To avoid this, follow the procedure below when switching a pin to input operation.

1. Disable the input in the peripheral module settings that correspond to functions of the pin to be switched.
2. Make the port function control register m setting to select input operation for the pin.
3. Enable the input in the peripheral module settings that correspond to functions of the pin to be switched

If the settings for a pin before and after it has been switched to operate as an output differ from each other, an internal edge is output. This may lead to operation that was not intended. To avoid this, follow the procedure below when switching a pin to output operation.

1. Disable the output in the peripheral module settings that correspond to functions of the pin to be switched.
2. Make the port function control register m setting to select output operation for the pin.
3. Enable the output in the peripheral module settings that correspond to functions of the pin to be switched.

A single pin function may correspond to a pin selection bit for changing a pin for the input or output and a pin enable bit for enabling a pin function. In such cases, set the pin for the input or output and then set the enable bit to enable the pin function.

17.6.3 Changing the Output Enable Settings

Pins are initially in the high-impedance state because the output function of the port is initially disabled. If the output enable setting is changed, the port leaves the high-impedance state and enters the output state (high or low); however, this may lead to generation of internal edges depending on the LSI internal state. To prevent edge generation, 0 or 1 should be previously set to the port data register (DR) using the procedure below.

Changing the pin state from high-impedance to high output:

1. Set 1 (set the internal state of the LSI to high) to the port data register (DR) of the pins.
2. Select the output pins by setting the port function register.
3. Enable the pins for output.

Changing the pin state from high-impedance to low output:

1. Set 0 (set the internal state of the LSI to low) to the port data register (DR) of the pins.
2. Select the output pins by setting the port function register.
3. Enable the pins for output.

17.6.4 Setting Open-Drain Pins with TDO Function

With the RX62N and RX621 Groups, setting the open-drain capability for the pin with TDO function allows the pin to provide open-drain output regardless of whether the pin is being used for TDO, thus disabling on-chip emulator communication.

In the RX62N and RX621 Group products, the TDO function is allocated to pin P26 in 145-pin TFLGA, 144-LQFP, 100-pin LQFP, and 85-pin TFLGA; setting the open-drain capability (PORT2.ODR.B6 = 1) for P26 disables communication using the TDO pin. To use the on-chip emulator with those products, do not set the open-drain capability for P26.

17.6.5 Reading Port Registers (PORT)

Before reading the port register, enable the input buffer of the corresponding pins by setting the corresponding bits in PORTn.ICR to 1.

If the PORTn.PORT register is read with bits in PORTn.ICR set to 0, undefined values are read from the corresponding bits.

18. Multi-Function Timer Pulse Unit 2 (MTU2)

18.1 Overview

The RX62N/RX621 Group has two on-chip multi-function timer pulse units (MTU). Each unit comprises six 16-bit timer channels (unit 0: channels 0 to 5; unit 1: channels 6 to 11) and a total of 12 channels are available.

Table 18.1 shows the specifications of the MTU, Table 18.2 shows the function list of MTU unit 0, and Table 18.3 shows the function list of MUT unit 1. Figure 18.1 shows a block diagram of the MTU.

Units 0 and 1 have the same functions; that is, the functions of channels 0 to 5 are the same as those of channels 6 to 11. The descriptions of unit 0 in this section are also applied to unit 1.

Table 18.1 Specifications of MTU

Item	Description
Pulse input/output	16 lines max.
Pulse input	3 lines
Count clock	Eight clocks or seven clocks for each channel (four clocks for channels 5 and 11)
Available operations	<p>[Channels 0 to 4 and 6 to 10]</p> <ul style="list-style-type: none"> • Waveform output at compare match • Input capture function • Counter clear operation • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous clearing by compare match or input capture • Simultaneous register input/output by synchronous counter operation • A maximum of 12-phase PWM output is available in combination with synchronous operation • Cascade connection operation <p>[Channels 0, 3, 4, 6, 9, and 10]</p> <ul style="list-style-type: none"> • Buffer operation specifiable • 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. <p>[Channels 1, 2, 7, and 8]</p> <ul style="list-style-type: none"> • Phase counting mode specifiable independently <p>[Channels 3, 4, 9, and 10]</p> <ul style="list-style-type: none"> • A total of six-phase waveform output, which includes three phases each for positive and negative complementary PWM or reset PWM output, by interlocking operation <p>[Channels 5 and 11]</p> <ul style="list-style-type: none"> • Dead time compensation counter
Complementary PWM mode	<ul style="list-style-type: none"> • Interrupts at the crest and trough of the counter value • A/D converter start triggers can be skipped
Interrupt sources	28 sources
Buffer operation	Automatic transfer of register data
Trigger generation	<p>Programmable pulse generator (PPG) output trigger can be generated</p> <p>A/D converter start trigger can be generated</p>
Power reduction	Each unit can be independently set to module stop state

Table 18.2 MTU Functions (Unit 0) (1 / 3)

Item	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
Count clock	PCLK/1	PCLK/1	PCLK/1	PCLK/1	PCLK/1	PCLK/1
	PCLK/4	PCLK/4	PCLK/4	PCLK/4	PCLK/4	PCLK/4
	PCLK/16	PCLK/16	PCLK/16	PCLK/16	PCLK/16	PCLK/16
	PCLK/64	PCLK/64	PCLK/64	PCLK/64	PCLK/64	PCLK/64
	MTCLKA	PCLK/256	PCLK/1024	PCLK/256	PCLK/256	
	MTCLKB	MTCLKA	MTCLKA	PCLK/1024	PCLK/1024	
	MTCLKC	MTCLKB	MTCLKB	MTCLKA	MTCLKA	
	MTCLKD		MTCLKC	MTCLKB	MTCLKB	
General registers (TGR)	TGRA	TGRA	TGRA	TGRA	TGRA	TGRU
	TGRB	TGRB	TGRB	TGRB	TGRB	TGRV
	TGRE					TGRW
General registers/ buffer registers	TGRC	—	—	TGRC	TGRC	—
	TGRD			TGRD	TGRD	
	TGRF					
I/O pins	MTIOC0A	MTIOC1A	MTIOC2A	MTIOC3A	MTIOC4A	Input pins
	MTIOC0B	MTIOC1B	MTIOC2B	MTIOC3B	MTIOC4B	MTIC5U
	MTIOC0C			MTIOC3C	MTIOC4C	MTIC5V
	MTIOC0D			MTIOC3D	MTIOC4D	MTIC5W
Counter clear function	TGR	TGR	TGR	TGR	TGR	TGR
	compare	compare	compare	compare	compare	compare
	match or input	match or input	match or input	match or input	match or input	match or input
	capture	capture	capture	capture	capture	capture or input capture
Compa re 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	√	—	—	√	√	—

Table 18.2 MTU Functions (Unit 0) (2 / 3)

Item	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
Dead time compensation counter function	—	—	—	—	—	√
DMACA activation	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	—
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow or underflow	TGR compare match or input capture
A/D converter start trigger	TGRA compare match or input capture TGRB compare match or input capture TGRE compare match TGRF compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture TCNT underflow (trough) in complementary PWM mode	—
PPG trigger	TGRA and TGRB compare match or input capture	TGRA and TGRB compare match or input capture	TGRA and TGRB compare match or input capture	TGRA and TGRB compare match or input capture	—	—

Table 18.2 MTU Functions (Unit 0) (3 / 3)

Item	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
Interrupt sources	7 sources <ul style="list-style-type: none"> • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Compare match 0E • Compare match 0F • Overflow 	4 sources <ul style="list-style-type: none"> • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow 	4 sources <ul style="list-style-type: none"> • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow 	5 sources <ul style="list-style-type: none"> • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow 	5 sources <ul style="list-style-type: none"> • Compare match or input capture 4A • Compare match or input capture 4B • Compare match or input capture 4C • Compare match or input capture 4D • Overflow or underflow 	3 sources <ul style="list-style-type: none"> • Compare match or input capture 5U • Compare match or input capture 5V • Compare match or input capture 5W
A/D converter start request delaying function	—	—	—	—	<ul style="list-style-type: none"> • A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT 	—
Interrupt skipping function	—	—	—	<ul style="list-style-type: none"> • Skips TGRA compare match interrupts 	<ul style="list-style-type: none"> • Skips TCIV interrupts 	—
Module stop function	MSTPCRA.MSTPA9*					

[Legend]

√: Possible

—: Not possible

Note: * For details on the module stop function, see section 9, Low Power Consumption.

Table 18.3 MTU Functions (Unit 1) (1 / 3)

Item	MTU6	MTU7	MTU8	MTU9	MTU10	MTU11	
Count clock	PCLK/1	PCLK/1	PCLK/1	PCLK/1	PCLK/1	PCLK/1	
	PCLK/4	PCLK/4	PCLK/4	PCLK/4	PCLK/4	PCLK/4	
	PCLK/16	PCLK/16	PCLK/16	PCLK/16	PCLK/16	PCLK/16	
	PCLK/64	PCLK/64	PCLK/64	PCLK/64	PCLK/64	PCLK/64	
	MTCLKE	PCLK/256	PCLK/1024	PCLK/256	PCLK/256		
	MTCLKF	MTCLKE	MTCLKE	PCLK/1024	PCLK/1024		
	MTCLKG	MTCLKF	MTCLKF	MTCLKE	MTCLKE		
	MTCLKH		MTCLKG	MTCLKF	MTCLKF		
General registers (TGR)	TGRA	TGRA	TGRA	TGRA	TGRA	TGRU	
	TGRB	TGRB	TGRB	TGRB	TGRB	TGRV	
	TGRE					TGRW	
General registers/ buffer registers	TGRC	—	—	TGRC	TGRC	—	
	TGRD			TGRD	TGRD		
	TGRF						
I/O pins	MTIOC6A	MTIOC7A	MTIOC8A	MTIOC9A	MTIOC10A	Input pins	
	MTIOC6B	MTIOC7B	MTIOC8B	MTIOC9B	MTIOC10B	MTIC11U	
	MTIOC6C			MTIOC9C	MTIOC10C	MTIC11V	
	MTIOC6D			MTIOC9D	MTIOC10D	MTIC11W	
Counter clear function	TGR	TGR	TGR	TGR	TGR	TGR	
	compare	compare	compare	compare	compare	compare	
	match or input capture	match or input capture	match or input capture	match or input capture	match or input capture	match or input capture	
						compare match or input capture	
Compa re 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 mode	—	—	—	√	√	—	
AC synchronous motor drive mode	√	—	—	√	√	—	
Phase counting mode	—	√	√	—	—	—	
Buffer operation	√	—	—	√	√	—	

Table 18.3 MTU Functions (Unit 1) (2 / 3)

Item	MTU6	MTU7	MTU8	MTU9	MTU10	MTU11
Dead time compensation counter function	—	—	—	—	—	√
DMACA activation	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	—
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow or underflow	TGR compare match or input capture
A/D converter start trigger	TGRA compare match or input capture TGRB compare match or input capture TGRE compare match TGRF compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture TCNT underflow (trough) in complementary PWM mode	—
PPG trigger	TGRA and TGRB compare match or input capture	TGRA and TGRB compare match or input capture	TGRA and TGRB compare match or input capture	TGRA and TGRB compare match or input capture	—	—

Table 18.3 MTU Functions (Unit 1) (3 / 3)

Item	MTU6	MTU7	MTU8	MTU9	MTU10	MTU11
Interrupt sources	7 sources <ul style="list-style-type: none"> • Compare match or input capture 6A • Compare match or input capture 6B • Compare match or input capture 6C • Compare match or input capture 6D • Compare match 6E • Compare match 6F • Overflow 	4 sources <ul style="list-style-type: none"> • Compare match or input capture 7A • Compare match or input capture 7B • Overflow • Underflow 	4 sources <ul style="list-style-type: none"> • Compare match or input capture 8A • Compare match or input capture 8B • Overflow • Underflow 	5 sources <ul style="list-style-type: none"> • Compare match or input capture 9A • Compare match or input capture 9B • Compare match or input capture 9C • Compare match or input capture 9D • Overflow 	5 sources <ul style="list-style-type: none"> • Compare match or input capture 10A • Compare match or input capture 10B • Compare match or input capture 10C • Compare match or input capture 10D • Overflow or underflow 	3 sources <ul style="list-style-type: none"> • Compare match or input capture 11U • Compare match or input capture 11V • Compare match or input capture 11W
A/D converter start request delaying function	—	—	—	—	• A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT	—
Interrupt skipping function	—	—	—	• Skips TGRA compare match interrupts	• Skips TCIV interrupts	—
Module stop function	MSTPCRA.MSTPA8*					

[Legend]

√: Possible

—: Not possible

Note: * For details on the module stop function, see section 9., Low Power Consumption.

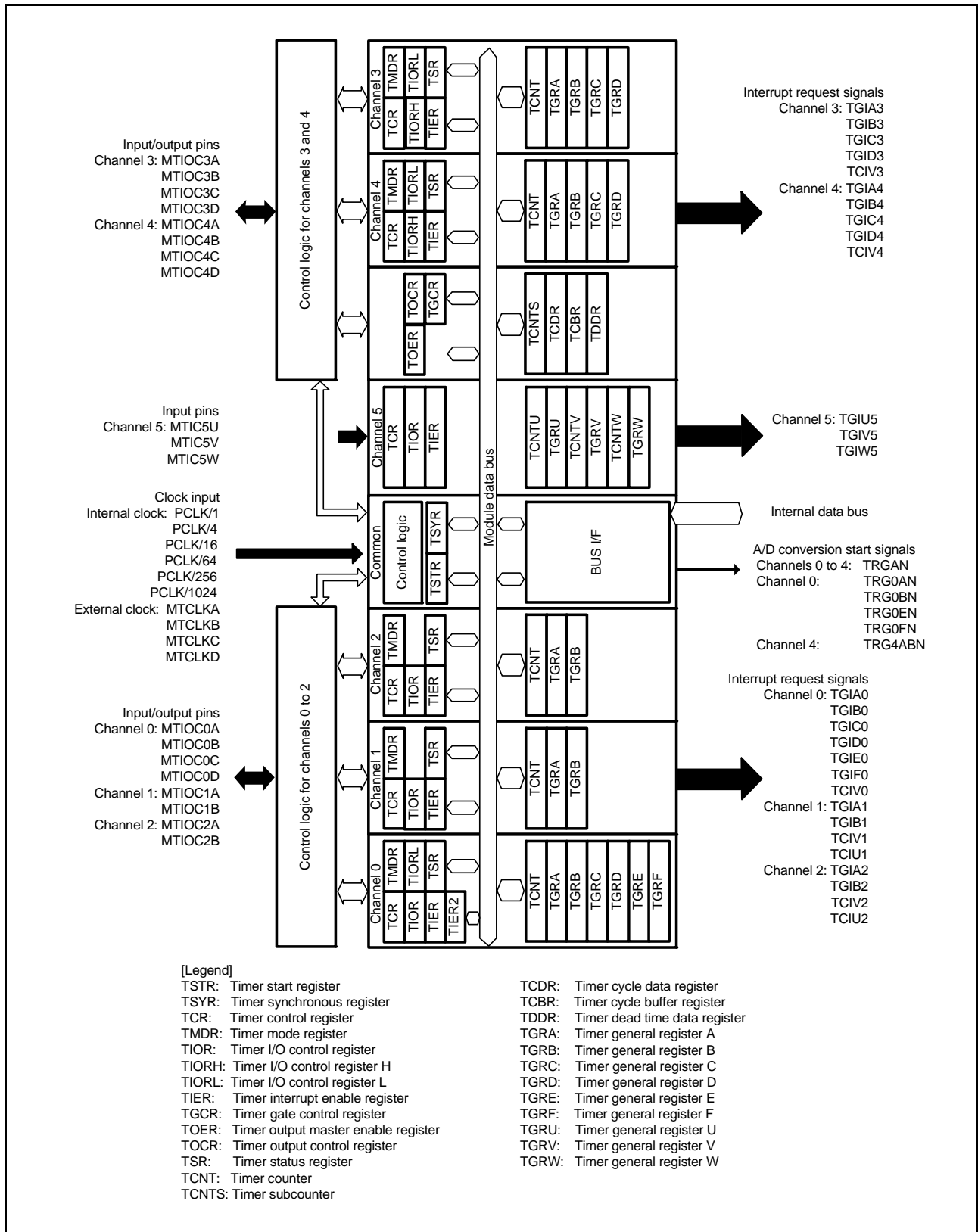


Figure 18.1 Block Diagram of MTU (Unit 0)

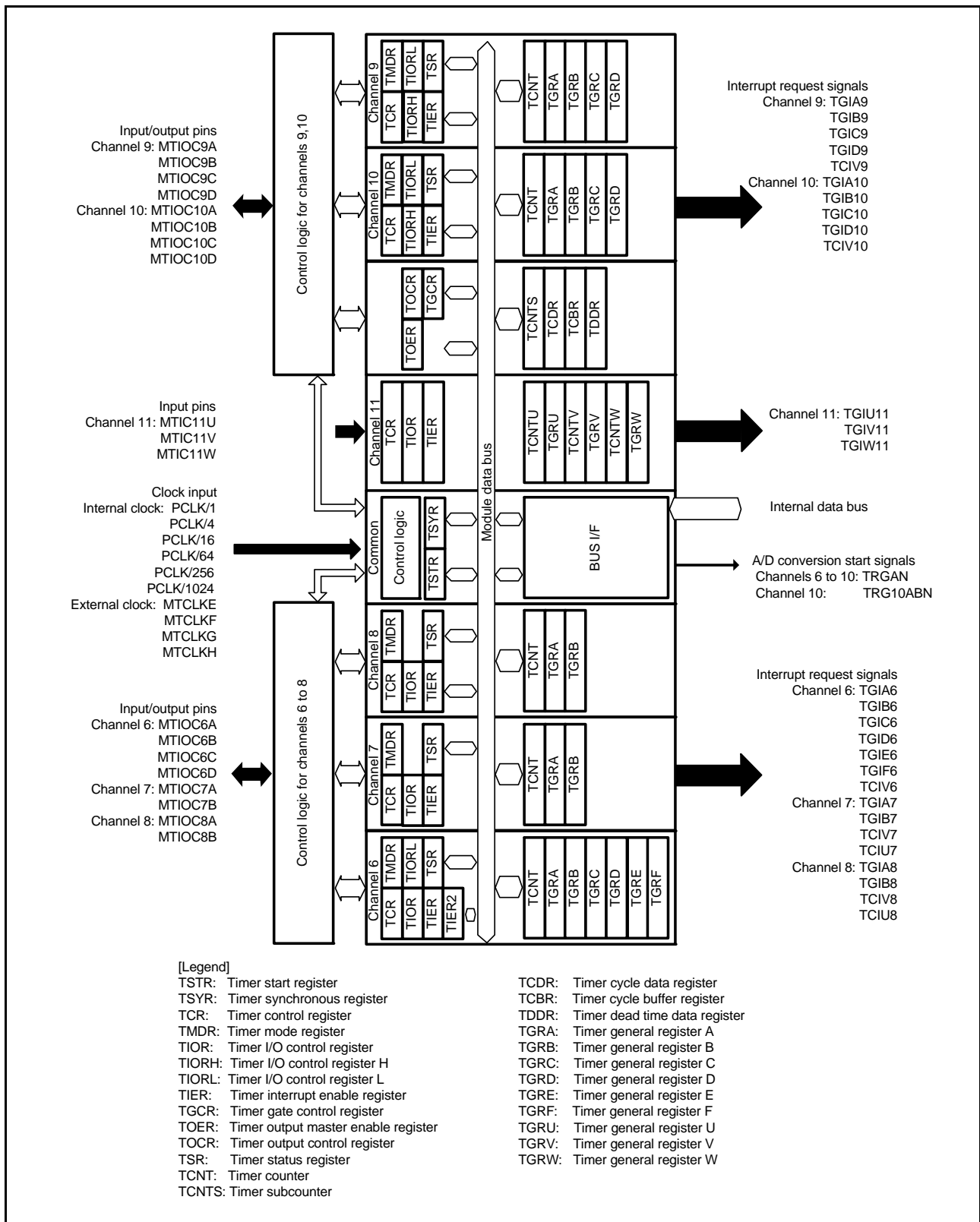


Figure 18.2 Block Diagram of MTU (Unit 1)

Table 18.4 shows the pin configuration of the MTU.

Table 18.4 Pin Configuration (1 / 2)

Unit	Channel	Pin Name	I/O	Function
Unit 0	Common	MTCLKA	Input	External clock A input pin (Channel 1 phase counting mode A phase input)
		MTCLKB	Input	External clock B input pin (Channel 1 phase counting mode B phase input)
		MTCLKC	Input	External clock C input pin (Channel 2 phase counting mode A phase input)
		MTCLKD	Input	External clock D input pin (Channel 2 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

Table 18.4 Pin Configuration (2 / 2)

Unit	Channel	Pin Name	I/O	Function
Unit 1	Common	MTCLKE	Input	External clock E input pin (channel 7 phase counting mode A phase input)
		MTCLKF	Input	External clock F input pin (channel 7 phase counting mode B phase input)
		MTCLKG	Input	External clock G input pin (channel 8 phase counting mode A phase input)
		MTCLKH	Input	External clock H input pin (channel 8 phase counting mode B phase input)
	MTU6	MTIOC6A	I/O	TGRA6 input capture input/output compare output/PWM output pin
		MTIOC6B	I/O	TGRB6 input capture input/output compare output/PWM output pin
		MTIOC6C	I/O	TGRC6 input capture input/output compare output/PWM output pin
		MTIOC6D	I/O	TGRD6 input capture input/output compare output/PWM output pin
	MTU7	MTIOC7A	I/O	TGRA7 input capture input/output compare output/PWM output pin
		MTIOC7B	I/O	TGRB7 input capture input/output compare output/PWM output pin
	MTU8	MTIOC8A	I/O	TGRA8 input capture input/output compare output/PWM output pin
		MTIOC8B	I/O	TGRB8 input capture input/output compare output/PWM output pin
	MTU9	MTIOC9A	I/O	TGRA9 input capture input/output compare output/PWM output pin
		MTIOC9B	I/O	TGRB9 input capture input/output compare output/PWM output pin
		MTIOC9C	I/O	TGRC9 input capture input/output compare output/PWM output pin
		MTIOC9D	I/O	TGRD9 input capture input/output compare output/PWM output pin
MTU10	MTIOC10A	I/O	TGRA10 input capture input/output compare output/PWM output pin	
	MTIOC10B	I/O	TGRB10 input capture input/output compare output/PWM output pin	
	MTIOC10C	I/O	TGRC10 input capture input/output compare output/PWM output pin	
	MTIOC10D	I/O	TGRD10 input capture input/output compare output/PWM output pin	
MTU11	MTIC11U	Input	TGRU11 input capture input/external pulse input pin	
	MTIC11V	Input	TGRV11 input capture input/external pulse input pin	
	MTIC11W	Input	TGRW11 input capture input/external pulse input pin	

18.2 Register Descriptions

Table 18.5 and Table 18.6 list the registers of the MTU.

Table 18.5 Registers of MTU (1 / 5)

Unit	Channel	Register Name	Symbol	Value after Reset	Address	Access Size		
Unit 0	MTU0	Timer control register	TCR	00h	0008 8700h	8		
		Timer mode register	TMDR	00h	0008 8701h	8		
		Timer I/O control register H	TIORH	00h	0008 8702h	8		
		Timer I/O control register L	TIORL	00h	0008 8703h	8		
		Timer interrupt enable register	TIER	00h	0008 8704h	8		
		Timer status register	TSR	C0h	0008 8705h	8		
		Timer counter	TCNT	0000h	0008 8706h	16		
		Timer general register A	TGRA	FFFFh	0008 8708h	16		
		Timer general register B	TGRB	FFFFh	0008 870Ah	16		
		Timer general register C	TGRC	FFFFh	0008 870Ch	16		
		Timer general register D	TGRD	FFFFh	0008 870Eh	16		
		Timer general register E	TGRE	FFFFh	0008 8720h	16		
		Timer general register F	TGRF	FFFFh	0008 8722h	16		
		Timer interrupt enable register 2	TIER2	00h	0008 8724h	8		
		Timer buffer operation transfer mode register	TBTM	00h	0008 8726h	8		
		MTU1	MTU1	Timer control register	TCR	00h	0008 8780h	8
				Timer mode register	TMDR	00h	0008 8781h	8
Timer I/O control register	TIOR			00h	0008 8782h	8		
Timer interrupt enable register	TIER			00h	0008 8784h	8		
Timer status register	TSR			C0h	0008 8785h	8		
Timer counter	TCNT			0000h	0008 8786h	16		
Timer general register A	TGRA			FFFFh	0008 8788h	16		
Timer general register B	TGRB			FFFFh	0008 878Ah	16		
Timer input capture control register	TICCR			00h	0008 8790h	8		
MTU2	MTU2	Timer control register	TCR	00h	0008 8800h	8		
		Timer mode register	TMDR	00h	0008 8801h	8		
		Timer I/O control register	TIOR	00h	0008 8802h	8		
		Timer interrupt enable register	TIER	00h	0008 8804h	8		
		Timer status register	TSR	C0h	0008 8805h	8		
		Timer counter	TCNT	0000h	0008 8806h	16		
		Timer general register A	TGRA	FFFFh	0008 8808h	16		
Timer general register B	TGRB	FFFFh	0008 880Ah	16				

Table 18.5 Registers of MTU (2 / 5)

Unit	Channel	Register Name	Symbol	Value after Reset	Address	Access Size
Unit 0	MTU3	Timer control register	TCR	00h	0008 8600h	8
		Timer mode register	TMDR	00h	0008 8602h	8
		Timer I/O control register H	TIORH	00h	0008 8604h	8
		Timer I/O control register L	TIORL	00h	0008 8605h	8
		Timer interrupt enable register	TIER	00h	0008 8608h	8
		Timer counter	TCNT	0000h	0008 8610h	16
		Timer general register A	TGRA	FFFFh	0008 8618h	16
		Timer general register B	TGRB	FFFFh	0008 861Ah	16
		Timer general register C	TGRC	FFFFh	0008 8624h	16
		Timer general register D	TGRD	FFFFh	0008 8626h	16
		Timer status register	TSR	C0h	0008 862Ch	8
		Timer buffer operation transfer mode register	TBTM	00h	0008 8638h	8
		MTU4	MTU4	Timer control register	TCR	00h
Timer mode register	TMDR			00h	0008 8603h	8
Timer I/O control register H	TIORH			00h	0008 8606h	8
Timer I/O control register L	TIORL			00h	0008 8607h	8
Timer interrupt enable register	TIER			00h	0008 8609h	8
Timer counter	TCNT			0000h	0008 8612h	16
Timer general register A	TGRA			FFFFh	0008 861Ch	16
Timer general register B	TGRB			FFFFh	0008 861Eh	16
Timer general register C	TGRC			FFFFh	0008 8628h	16
Timer general register D	TGRD			FFFFh	0008 862Ah	16
Timer status register	TSR			C0h	0008 862Dh	8
Timer buffer operation transfer mode register	TBTM			00h	0008 8639h	8
Timer A/D converter start request control register	TADCR			0000h	0008 8640h	16
Timer A/D converter start request cycle set register A	TADCORA			FFFFh	0008 8644h	16
Timer A/D converter start request cycle set register B	TADCORB			FFFFh	0008 8646h	16
Timer A/D converter start request cycle set buffer register A	TADCOBRA			FFFFh	0008 8648h	16
Timer A/D converter start request cycle set buffer register B	TADCOBRB			FFFFh	0008 864Ah	16
MTU5	MTU5	Timer counter U	TCNTU	0000h	0008 8880h	16
		Timer general register U	TGRU	FFFFh	0008 8882h	16
		Timer control register U	TCRU	00h	0008 8884h	8
		Timer I/O control register U	TIORU	00h	0008 8886h	8
		Timer counter V	TCNTV	0000h	0008 8890h	16
		Timer general register V	TGRV	FFFFh	0008 8892h	16
		Timer control register V	TCRV	00h	0008 8894h	8
		Timer I/O control register V	TIORV	00h	0008 8896h	8
		Timer counter W	TCNTW	0000h	0008 88A0h	16
		Timer general register W	TGRW	FFFFh	0008 88A2h	16
		Timer control register W	TCRW	00h	0008 88A4h	8
		Timer I/O control register W	TIORW	00h	0008 88A6h	8
		Timer interrupt enable register	TIER	00h	0008 88B2h	8
		Timer start register	TSTR	00h	0008 88B4h	8
Timer compare match clear register	TCNTCMPCLR	00h	0008 88B6h	8		

Table 18.5 Registers of MTU (3 / 5)

Unit	Channel	Register Name	Symbol	Value after Reset	Address	Access Size		
Unit 0	MTUA	Timer output master enable register	TOER	C0h	0008 860Ah	8		
		Timer gate control register	TGCR	80h	0008 860Dh	8		
		Timer output control register 1	TOCR1	00h	0008 860Eh	8		
		Timer output control register 2	TOCR2	00h	0008 860Fh	8		
		Timer cycle data register	TCDR	FFFFh	0008 8614h	16		
		Timer dead time data register	TDDR	FFFFh	0008 8616h	16		
		Timer subcounter	TCNTS	0000h	0008 8620h	16		
		Timer cycle buffer register	TCBR	FFFFh	0008 8622h	16		
		Timer interrupt skipping set register	TITCR	00h	0008 8630h	8		
		Timer interrupt skipping counter	TITCNT	00h	0008 8631h	8		
		Timer buffer transfer set register	TBTER	00h	0008 8632h	8		
		Timer dead time enable register	TDER	01h	0008 8634h	8		
		Timer output level buffer register	TOLBR	00h	0008 8636h	8		
		Timer waveform control register	TWCR	00h	0008 8660h	8		
		Timer start register	TSTR	00h	0008 8680h	8		
Timer synchronous register	TSYR	00h	0008 8681h	8				
Timer read/write enable register	TRWER	01h	0008 8684h	8				
Unit 1	MTU6	Timer control register	TCR	00h	0008 8B00h	8		
		Timer mode register	TMDR	00h	0008 8B01h	8		
		Timer I/O control register H	TIORH	00h	0008 8B02h	8		
		Timer I/O control register L	TIORL	00h	0008 8B03h	8		
		Timer interrupt enable register	TIER	00h	0008 8B04h	8		
		Timer status register	TSR	C0h	0008 8B05h	8		
		Timer counter	TCNT	0000h	0008 8B06h	16		
		Timer general register A	TGRA	FFFFh	0008 8B08h	16		
		Timer general register B	TGRB	FFFFh	0008 8B0Ah	16		
		Timer general register C	TGRC	FFFFh	0008 8B0Ch	16		
		Timer general register D	TGRD	FFFFh	0008 8B0Eh	16		
		Timer general register E	TGRE	FFFFh	0008 8B20h	16		
		Timer general register F	TGRF	FFFFh	0008 8B22h	16		
		Timer interrupt enable register 2	TIER2	00h	0008 8B24h	8		
		Timer buffer operation transfer mode register	TBTM	00h	0008 8B26h	8		
		MTU7	MTU7	Timer control register	TCR	00h	0008 8B80h	8
				Timer mode register	TMDR	00h	0008 8B81h	8
				Timer I/O control register	TIOR	00h	0008 8B82h	8
Timer interrupt enable register	TIER			00h	0008 8B84h	8		
Timer status register	TSR			C0h	0008 8B85h	8		
Timer counter	TCNT			0000h	0008 8B86h	16		
Timer general register A	TGRA			FFFFh	0008 8B88h	16		
Timer general register B	TGRB			FFFFh	0008 8B8Ah	16		
Timer input capture control register	TICCR			00h	0008 8B90h	8		

Table 18.5 Registers of MTU (4 / 5)

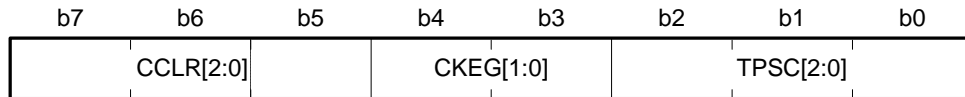
Unit	Channel	Register Name	Symbol	Value after Reset	Address	Access Size
Unit 1	MTU8	Timer control register	TCR	00h	0008 8C00h	8
		Timer mode register	TMDR	00h	0008 8C01h	8
		Timer I/O control register	TIOR	00h	0008 8C02h	8
		Timer interrupt enable register	TIER	00h	0008 8C04h	8
		Timer status register	TSR	C0h	0008 8C05h	8
		Timer counter	TCNT	0000h	0008 8C06h	16
		Timer general register A	TGRA	FFFFh	0008 8C08h	16
		Timer general register B	TGRB	FFFFh	0008 8C0Ah	16
MTU9	MTU9	Timer control register	TCR	00h	0008 8A00h	8
		Timer mode register	TMDR	00h	0008 8A02h	8
		Timer I/O control register H	TIORH	00h	0008 8A04h	8
		Timer I/O control register L	TIORL	00h	0008 8A05h	8
		Timer interrupt enable register	TIER	00h	0008 8A08h	8
		Timer counter	TCNT	0000h	0008 8A10h	16
		Timer general register A	TGRA	FFFFh	0008 8A18h	16
		Timer general register B	TGRB	FFFFh	0008 8A1Ah	16
		Timer general register C	TGRC	FFFFh	0008 8A24h	16
		Timer general register D	TGRD	FFFFh	0008 8A26h	16
		Timer status register	TSR	C0h	0008 8A2Ch	8
		Timer buffer operation transfer mode register	TBTM	00h	0008 8A38h	8
MTU10	MTU10	Timer control register	TCR	00h	0008 8A01h	8
		Timer mode register	TMDR	00h	0008 8A03h	8
		Timer I/O control register H	TIORH	00h	0008 8A06h	8
		Timer I/O control register L	TIORL	00h	0008 8A07h	8
		Timer interrupt enable register	TIER	00h	0008 8A09h	8
		Timer counter	TCNT	0000h	0008 8A12h	16
		Timer general register A	TGRA	FFFFh	0008 8A1Ch	16
		Timer general register B	TGRB	FFFFh	0008 8A1Eh	16
		Timer general register C	TGRC	FFFFh	0008 8A28h	16
		Timer general register D	TGRD	FFFFh	0008 8A2Ah	16
		Timer status register	TSR	C0h	0008 8A2Dh	8
		Timer buffer operation transfer mode register	TBTM	00h	0008 8A39h	8
		Timer A/D converter start request control register	TADCR	0000h	0008 8A40h	16
		Timer A/D converter start request cycle set register A	TADCORA	FFFFh	0008 8A44h	16
		Timer A/D converter start request cycle set register B	TADCORB	FFFFh	0008 8A46h	16
		Timer A/D converter start request cycle set buffer register A	TADCOBRA	FFFFh	0008 8A48h	16
		Timer A/D converter start request cycle set buffer register B	TADCOBRB	FFFFh	0008 8A4Ah	16

Table 18.5 Registers of MTU (5 / 5)

Unit	Channel	Register Name	Symbol	Value after Reset	Address	Access Size	
Unit 1	MTU11	Timer counter U	TCNTU	0000h	0008 8C80h	16	
		Timer general register U	TGRU	FFFFh	0008 8C82h	16	
		Timer control register U	TCRU	00h	0008 8C84h	8	
		Timer I/O control register U	TIORU	00h	0008 8C86h	8	
		Timer counter V	TCNTV	0000h	0008 8C90h	16	
		Timer general register V	TGRV	FFFFh	0008 8C92h	16	
		Timer control register V	TCRV	00h	0008 8C94h	8	
		Timer I/O control register V	TIORV	00h	0008 8C96h	8	
		Timer counter W	TCNTW	0000h	0008 8CA0h	16	
		Timer general register W	TGRW	FFFFh	0008 8CA2h	16	
		Timer control register W	TCRW	00h	0008 8CA4h	8	
		Timer I/O control register W	TIORW	00h	0008 8CA6h	8	
		Timer interrupt enable register	TIER	00h	0008 8CB2h	8	
		Timer start register	TSTR	00h	0008 8CB4h	8	
		Timer compare match clear register	TCNTCMPCLR	00h	0008 8CB6h	8	
		MTUB	Timer output master enable register	TOER	C0h	0008 8A0Ah	8
			Timer gate control register	TGCR	80h	0008 8A0Dh	8
			Timer output control register 1	TOCR1	00h	0008 8A0Eh	8
			Timer output control register 2	TOCR2	00h	0008 8A0Fh	8
			Timer cycle data register	TCDR	FFFFh	0008 8A14h	16
Timer dead time data register	TDDR		FFFFh	0008 8A16h	16		
Timer subcounter	TCNTS		0000h	0008 8A20h	16		
Timer cycle buffer register	TCBR		FFFFh	0008 8A22h	16		
Timer interrupt skipping set register	TITCR		00h	0008 8A30h	8		
Timer interrupt skipping counter	TITCNT		00h	0008 8A31h	8		
Timer buffer transfer set register	TBTER		00h	0008 8A32h	8		
Timer dead time enable register	TDER		01h	0008 8A34h	8		
Timer output level buffer register	TOLBR		00h	0008 8A36h	8		
Timer waveform control register	TWCR		00h	0008 8A60h	8		
Timer start register	TSTR		00h	0008 8A80h	8		
Timer synchronous register	TSYR		00h	0008 8A81h	8		
Timer read/write enable register	TRWER		01h	0008 8A84h	8		

18.2.1 Timer Control Register (TCR)

Addresses: 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, MTU6.TCR 0008 8B00h, MTU7.TCR 0008 8B80h, MTU8.TCR 0008 8C00h, MTU9.TCR 0008 8A00h, MTU10.TCR 0008 8A01h, MTU11.TCRU 0008 8C84h, MTU11.TCRV 0008 8C94h, MTU11.TCRW 0008 8CA4h



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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Time Prescaler Select	See Table 18.8 to Table 18.12.	R/W
b4, b3	CKEG[1:0] *1	Clock Edge Select	b4 b3 0 0: Count at rising edge 0 1: Count at falling edge 1 x: Count at both edges	R/W
b7 to b5	CCLR[2:0]	Counter Clear Source Select	See Table 18.6 and Table 18.7.	R/W

[Legend]

x: Don't care

Note 1. In channels 5 and 11, bits 4 and 3 are reserved.

TCR is an 8-bit readable/writable register that controls the TCNT operation for each channel. The MTU has a total of 16 TCR registers, one each for channels 0 to 4 and 6 to 10 and three (TCRU, TCRV, and TCRW) for channels 5 and 11. TCR values should be specified only while TCNT operation is stopped.

TPSC[2:0] Bits (Time Prescaler Select)

These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See Table 18.8 to Table 18.12 for details.

CKEG[1: 0] Bits (Clock Edge Select)

These bits select the input clock edge. When the input clock is counted at both edges, the input clock period is halved (e.g. PCLK/4 at both edges = PCLK/2 at rising edge). If phase counting mode is used on channels 1 and 2, the setting of these bits is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is PCLK/4 or slower. When PCLK/1 or the overflow/underflow in another channel is selected for the input clock, a value can be written to these bits but counter operation compiles with the initial value.

CCLR[2:0] Bits (Counter Clear Source Select)

These bits select the TCNT counter clearing source. See Table 18.6 and Table 18.7 for details.

Table 18.6 CCLR[2:0] (Channels 0, 3, 4, 6, 9, and 10)

Channel	Bit 7	Bit 6	Bit 5	Description
	CCLR2	CCLR1	CCLR0	
0, 3, 4, 6, 9, 10	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare match/input capture *2
	1	1	0	TCNT cleared by TGRD compare match/input capture *2
	1	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYR.SYNC bit to 1.

Note 2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority and compare match/input capture does not occur.

Table 18.7 CCLR[2:0] (Channels 1, 2, 7, and 8)

Channel	Bit 7	Bit 6	Bit 5	Description
	Reserved *2	CCLR1	CCLR0	
1, 2, 7, 8	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation *1

Note 1. Synchronous operation is selected by setting the TSYR.SYNC bit to 1.

Note 2. Bit 7 is reserved in channels 1, 2, 7, and 8. It is always read as 0. The write value should be 0.

Table 18.8 TPSC[2:0] (Channels 0 and 6)

Channel	Bit 2	Bit 1	Bit 0	Description
	TPSC2	TPSC1	TPSC0	
0, 6	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on MTCLKn pin input (n = A or E)
	1	0	1	External clock: counts on MTCLKn pin input (n = B or F)
	1	1	0	External clock: counts on MTCLKn pin input (n = C or G)
	1	1	1	External clock: counts on MTCLKn pin input (n = D or H)

Table 18.9 TPSC[2:0] (Channels 1 and 7)

Channel	Bit 2	Bit 1	Bit 0	Description
	TPSC2	TPSC1	TPSC0	
1, 7	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on MTCLKn pin input (n = A or E)
	1	0	1	External clock: counts on MTCLKn pin input (n = B or F)
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Counts on TCNTn overflow/underflow (n = 2 or 8)

Note: This setting is ignored when channel 1 or 7 is in phase counting mode.

Table 18.10 TPSC[2:0] (Channels 2 and 8)

Channel	Bit 2	Bit 1	Bit 0	Description
	TPSC2	TPSC1	TPSC0	
2, 8	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on MTCLKn pin input (n = A or E)
	1	0	1	External clock: counts on MTCLKn pin input (n = B or F)
	1	1	0	External clock: counts on MTCLKn pin input (n = C or G)
	1	1	1	Internal clock: counts on PCLK/1024

Note: This setting is ignored when channel 2 or 8 is in phase counting mode.

Table 18.11 TPSC[2:0] (Channels 3, 4, 9, and 10)

Channel	Bit 2	Bit 1	Bit 0	Description
	TPSC2	TPSC1	TPSC0	
3, 4, 9, 10	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	Internal clock: counts on PCLK/256
	1	0	1	Internal clock: counts on PCLK/1024
	1	1	0	External clock: counts on MTCLKA/MTCLKE pin input
	1	1	1	External clock: counts on MTCLKB/MTCLKF pin input

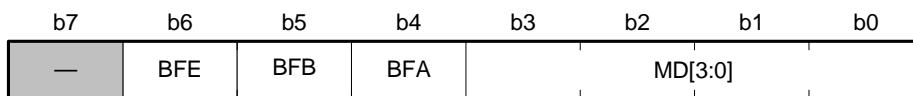
Table 18.12 TPSC[2:0] (Channels 5 and 11)

Channel	Bit 1	Bit 0	Description
	TPSC1	TPSC0	
5, 11	0	0	Internal clock: counts on PCLK/1
	0	1	Internal clock: counts on PCLK/4
	1	0	Internal clock: counts on PCLK/16
	1	1	Internal clock: counts on PCLK/64

Note: Bit 2 is reserved in channels 5 and 11. These bits are always read as 0. The write value should be 0.

18.2.2 Timer Mode Register (TMDR)

Addresses: MTU0.TMDR 0008 8701h, MTU1.TMDR 0008 8781h, MTU2.TMDR 0008 8801h, MTU3.TMDR 0008 8602h, MTU4.TMDR 0008 8603h, MTU6.TMDR 0008 8B01h, MTU7.TMDR 0008 8B81h, MTU8.TMDR 0008 8C01h, MTU9.TMDR 0008 8A02h, MTU10.TMDR 0008 8A03h



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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	These bits specify the timer operating mode. See Table 18.13 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: MTUn.TGRE and MTUn.TGRF operate normally 1: MTUn.TGRE and MTUn.TGRF used together for buffer operation (n = 0 or 6)	R/W
b7	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W

TMDR is an 8-bit readable/writable register that specifies the operating mode of each channel. The MTU has a total of ten TMDR registers, one each for channels 0 to 4 and 6 to 10. TMDR values should be specified only while TCNT operation is stopped.

Table 18.13 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 operation
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

[Legend]

x: Don't care

Note 1. PWM mode 2 cannot be set for channels 3, 4, 9, and 10.

Note 2. Phase counting mode cannot be set for channels 0, 3, 4, 6, 9, and 10.

Note 3. Reset-synchronized PWM mode and complementary PWM mode can only be set for channels 3 and 9.

When channel 3 or 9 is set to reset-synchronized PWM mode or complementary PWM mode, the channel 4 or 10 settings become ineffective and automatically conform to the channel 3 or 9 setting, respectively. Do not set channels 4 and 10 to reset-synchronized PWM mode or complementary PWM mode. Set the initial values in channels 4 and 10.

Reset-synchronized PWM mode and complementary PWM mode cannot be set for channels 0, 1, 2, 6, 7, and 8.

BFA Bit (Buffer Operation A)

This bit specifies whether to operate TGRA in the normal way or to use TGRA and TGRC together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRC occurs in complementary PWM mode. If a compare match occurs on channel 4 or 10 in the Tb interval in complementary PWM mode, the TGIEC bit in timer interrupt enable register (MTU4.TIER or MTU10.TIER) should be cleared to 0.

When channel 3, 4, 9, or 10 is set to reset-synchronized PWM mode or complementary PWM mode, the buffer operation conforms to the channel 3 or 9 setting. Set the TMDR.BFA bit of channels 4 and 10 to 0.

In channels 1, 2, 7, and 8, which have no TGRC, this bit is reserved. It is always read as 0. The write value should be 0.

Refer to figure 17.41 for an illustration of the Tb interval in complementary PWM mode.

BFB Bit (Buffer Operation B)

This bit specifies whether to operate TGRB in the normal way or to use TGRB and TGRD together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRD occurs in complementary PWM mode. If a compare match occurs in the T_b interval in complementary PWM mode, the TGIED bit in timer interrupt enable register 3, 4, 9, or 10 (MTU3.TIER, MTU4.TIER, MTU9.TIER, or MTU10.TIER) should be cleared to 0.

When channel 3, 4, 9, or 10 is set to reset-synchronized PWM mode or complementary PWM mode, the buffer operation conforms to the channel 3 or 9 setting. Set the TMDR.BFB bit of channels 4 and 10 to 0.

In channels 1, 2, 7, and 8, which have no TGRD, this bit is reserved. It is always read as 0. The write value should be 0. Refer to figure 18.42 for an illustration of the T_b interval in complementary PWM mode.

BFE Bit (Buffer Operation E)

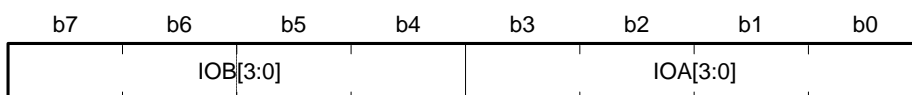
This bit specifies whether to operate MTU0.TGRE and MTU0.TGRF in the normal way or to use them together for buffer operation. Compare match with TGRF occurs even when TGRF is used as a buffer register. The state is the same in unit 1.

In channels 1 to 4 and 7 to 10, this bit is reserved. It is always read as 0. The write value should be 0.

18.2.3 Timer I/O Control Register (TIOR)

MTU0.TIORH, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU4.TIORH,
 MTU6.TIORH, MTU7.TIOR, MTU8.TIOR, MTU9.TIORH, MTU10.TIORH

Addresses: MTU0.TIORH 0008 8702h, MTU1.TIOR 0008 8782h, MTU2.TIOR 0008 8802h, MTU3.TIORH 0008 8604h,
 MTU4.TIORH 0008 8606h, MTU6.TIORH 0008 8B02h, MTU7.TIOR 0008 8B82h, MTU8.TIOR 0008 8C02h,
 MTU9.TIORH 0008 8A04h, MTU10.TIORH 0008 8A06h



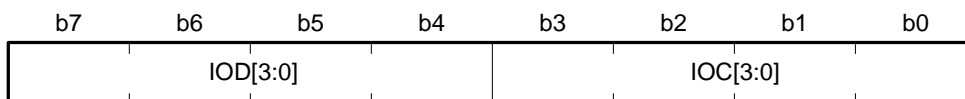
Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	I/O Control A	See the following tables. MTU0.TIORH: Table 18.22 MTU1.TIOR: Table 18.24 MTU2.TIOR: Table 18.25 MTU3.TIORH: Table 18.26 MTU4.TIORH: Table 18.28	R/W
b7 to b4	IOB[3:0]	I/O Control B	See the following tables. MTU0.TIORH: Table 18.14 MTU1.TIOR: Table 18.16 MTU2.TIOR: Table 18.17 MTU3.TIORH: Table 18.18 MTU4.TIORH: Table 18.20	R/W

Note : The above describes only unit 0 because channels 6 to 11 (unit 1) have the same functions as those of channels 0 to 5 (unit 0), respectively.

- MTU0.TIORL, MTU3.TIORL, MTU4.TIORL, MTU6.TIORL, MTU9.TIORL, MTU10.TIORL

Addresses: MTU0.TIORL 0008 8703h, MTU3.TIORL 0008 8605h, MTU4.TIORL 0008 8607h,
 MTU6.TIORL 0008 8B03h, MTU9.TIORL 0008 8A05h, MTU10.TIORL 0008 8A07h



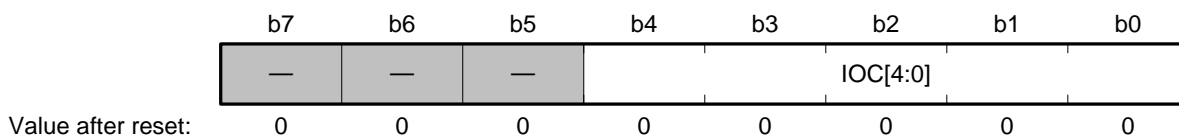
Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	I/O Control C	See the following tables. MTU0.TIORL: Table 18.23 MTU3.TIORL: Table 18.27 MTU4.TIORL: Table 18.29	R/W
b7 to b4	IOD[3:0]	I/O Control D	See the following tables. MTU0.TIORL: Table 18.15 MTU3.TIORL: Table 18.19 MTU4.TIORL: Table 18.21	R/W

Note : The above describes only unit 0 because channels 6, 9, and 10 (unit 1) have the same functions as those of channels 0, 3, and 4 (unit 0), respectively.

- MTU5.TIORU, MTU5.TIORV, MTU5.TIORW, MTU11.TIORU, MTU11.TIORV, MTU11.TIORW

Addresses: MTU5.TIORU 0008 8886h, MTU5.TIORV 0008 8896h, MTU5.TIORW 0008 88A6h,
 MTU11.TIORU 0008 8C86h, MTU11.TIORV 0008 8C96h, MTU11.TIORW 0008 8CA6h



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 18.30	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note : The above describes only unit 0 because channel 11 (unit 1) has the same functions as those of channel 5 (unit 0).

TIOR is an 8-bit readable/writable register that controls TGR. The MTU has a total of 22 TIOR registers, two each for channels 0, 3, 4, 6, 9, and 10, one each for channels 1, 2, 7, and 8, and six (MTU5.TIORU/V/W and MTU11.TIORU/V/W) each for channels 5 and 11.

TIOR should be set when TMDR is set to select normal operation, PWM mode, or phase counting mode.

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Table 18.14 TIORH (MTU0 and MTU6)

Bit 7	Bit 6	Bit 5	Bit 4	Description		
IOB3	IOB2	IOB1	IOB0	MTUn.TGRB Function	MTIOCnB 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 channel 1. Input capture at MTUm.TCNT up-count/down-count. (m = 1 or 7)		

[Legend]

x: Don't care

n = 0 or 6

Table 18.15 TIORL (MTU0 and MTU6)

Bit 7	Bit 6	Bit 5	Bit 4	Description		
IOD3	IOD2	IOD1	IOD0	MTUn.TGRD Function	MTIOCnD 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 channel m. Input capture at MTUm.TCNT up-count/down-count. (m = 1 or 7)		

[Legend]

x: Don't care

n = 0 or 6

Note : * When the TMDRn.BFB is set to 1 and MTUn.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.16 TIOR (MTU1 and MTU7)

Bit 7	Bit 6	Bit 5	Bit 4	Description		
IOB3	IOB2	IOB1	IOB0	MTUm.TGRB Function	MTIOCNB 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 MTUm.TGRC compare match/input capture.. (m = 0 or 6)		

[Legend]

x: Don't care

n = 1 or 7

Table 18.17 TIOR (MTU2 and MTU8)

Bit 7	Bit 6	Bit 5	Bit 4	Description		
IOB3	IOB2	IOB1	IOB0	MTUn.TGRB Function	MTIOCNB 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.		

[Legend]

x: Don't care

n = 2 or 8

Table 18.18 TIORH (MTU3 and MTU9)

Bit 7	Bit 6	Bit 5	Bit 4	Description		
IOB3	IOB2	IOB1	IOB0	MTUn.TGRB Function	MTIOCNB 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.		

[Legend]

x: Don't care

n = 3 or 9

Table 18.19 TIORL (MTU3 and MTU9)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD3	IOD2	IOD1	IOD0	MTUn.TGRD Function	MTIOCnD 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.

[Legend]

x: Don't care

n = 3 or 9

Note : * When the BFB bit in MTUn.TMDR is set to 1 and MTUn.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.20 TIORH (MTU4 and MTU10)

Bit 7	Bit 6	Bit 5	Bit 4	Description		
IOB3	IOB2	IOB1	IOB0	MTUn.TGRB Function	MTIOcNB 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.		

[Legend]

x: Don't care

n = 4 or 10

Table 18.21 TIORL (MTU4 and MTU10)

Bit 7	Bit 6	Bit 5	Bit 4	Description		
IOD3	IOD2	IOD1	IOD0	MTUn.TGRD Function	MTIOCnD 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.		

[Legend]

x: Don't care

n = 4 or 10

Note : * When the BFB bit in MTUn.TMDR is set to 1 and MTUn.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.22 TIORH (MTU0 and MTU6)

Bit 3	Bit 2	Bit 1	Bit 0	Description		
IOA3	IOA2	IOA1	IOA0	MTUn.TGRA Function	MTIOCnA 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 channel 1 or 7. Input capture at MTU1/7.TCNT up-count/down-count.		

[Legend]

x: Don't care

n = 0 or 6

Table 18.23 TIORL (MTU0 and MTU6)

Bit 3	Bit 2	Bit 1	Bit 0	Description		
IOC3	IOC2	IOC1	IOC0	MTUn.TGRC Function	MTIOCNc 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 channel 1 or 7. Input capture at MTU1/7.TCNT up-count/down-count.		

[Legend]

x: Don't care

n = 0 or 6

Note : * When the BFA bit in MTUn.TMDR is set to 1 and MTUn.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.24 TIOR (MTU1 and MTU7)

Bit 3	Bit 2	Bit 1	Bit 0	Description		
IOA3	IOA2	IOA1	IOA0	MTUm.TGRA Function	MTIOCnA 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 MTUm.TGRC compare match/input capture. (m = 0 or 6)		

[Legend]

x: Don't care

n = 1 or 7

Table 18.25 TIOR (MTU2 and MTU8)

Bit 3	Bit 2	Bit 1	Bit 0	Description		
IOA3	IOA2	IOA1	IOA0	MTUn.TGRA Function	MTIOCnA 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.		

[Legend]

x: Don't care

n = 2 or 8

Table 18.26 TIORH (MTU3 and MTU9)

Bit 3	Bit 2	Bit 1	Bit 0	Description		
IOA3	IOA2	IOA1	IOA0	MTUn.TGRA Function	MTIOcNA 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.		

[Legend]

x: Don't care

n = 3 or 9

Table 18.27 TIORL (MTU3 and MTU9)

Bit 3	Bit 2	Bit 1	Bit 0	Description		
IOC3	IOC2	IOC1	IOC0	MTUn.TGRC Function	MTIOCNc 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.		

[Legend]

x: Don't care

n = 3 or 9

Note : * When the BFA bit in MTUn.TMDR is set to 1 and MTUn.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.28 TIORH (MTU4 and MTU10)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	IOA2	IOA1	IOA0	MTUn.TGRA Function	MTIOCnA 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.

[Legend]

x: Don't care

n = 4 or 10

Table 18.29 TIORL (MTU4 and MTU10)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC3	IOC2	IOC1	IOC0	MTUn.TGRC Function	MTIOcNc 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.

[Legend]

x: Don't care

n = 4 or 10

Note : * When the BFA bit in MTUn.TMDR is set to 1 and MTUn.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.30 TIORU, TIORV, and TIORW (MTU5 and MTU11)

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC4	IOC3	IOC2	IOC1	IOC0	MTUn.TGRU, MTUn.TGRV, MTUn.TGRW Function	
					MTICnU, MTICnV, MTICnW 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 crest of complementary PWM mode.
1	1	0	1	1		Measurement of low pulse width of external input signal. Capture at crest and trough of complementary PWM mode.
1	1	1	0	0		Setting prohibited
1	1	1	0	1		Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	1	1	0		Measurement of high pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	1	1	1		Measurement of high pulse width of external input signal. Capture at crest and trough of complementary PWM mode.

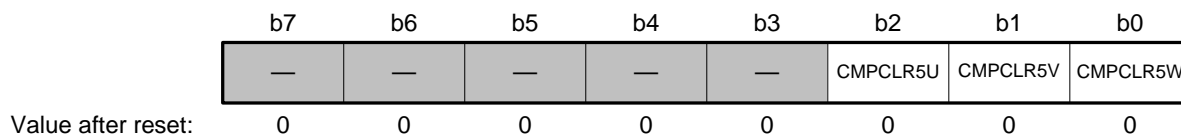
[Legend]

x: Don't care

n = 5 or 11

18.2.4 Timer Compare Match Clear Register (TCNTCMPCLR)

Addresses: MTU5.TCNTCMPCLR 0008 88B6h, MTU11.TCNTCMPCLR 0008 8CB6h



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 always read as 0. The write value should be 0.	R/W

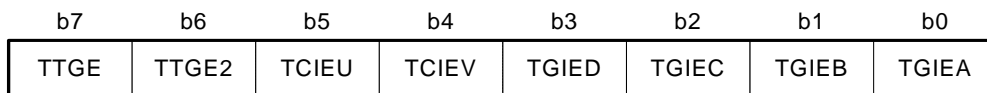
Note : Only unit 0 is described here because channel 11 (unit 1) has the same functions as those of channel 5 (unit 0).

TCNTCMPCLR is an 8-bit readable/writable register that specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW. The MTU has one TCNTCMPCLR each for channels 5 and 11.

18.2.5 Timer Interrupt Enable Register (TIER)

- TIER (MTU0 to MTU4, MTU6 to MTU10)

Addresses: MTU0.TIER 0008 8704h, MTU1.TIER 0008 8784h, MTU2.TIER 0008 8804h, MTU3.TIER 0008 8608h, MTU4.TIER 0008 8609h, MTU6.TIER 0008 8B04h, MTU7.TIER 0008 8B84h, MTU8.TIER 0008 8C04h, MTU9.TIER 0008 8A08h, MTU10.TIER 0008 8A09h



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 MTUn.TCNT underflow (trough) disabled 1: A/D converter start request generation by MTUn.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

[Legend] n = 4 or 10

TIER is an 8-bit readable/writable register that enables or disables interrupt requests in each channel. The MTU has a total of 14 TIER registers, two each for channels 0 and 6 and one each for channels 1 to 5 and 7 to 11.

TGIEA and TGIEB Bits (TGR Interrupt Enable A and B)

Each bit enables or disables interrupt requests (TGIn) (n = A or B).

TGIEC and TGIED Bits (TGR Interrupt Enable C and D)

Each bit enables or disables interrupt requests (TGIn) in channels 0, 3, 4, 6, 9, and 10 (n = C or D).

In channels 1, 2, 7, and 8, these bits are reserved. They are always 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 channels 1, 2, 7, and 8.

In channels 0, 3, 4, 6, 9, and 10, this bit is reserved. It is always 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 MTU4/10.TCNT underflow (trough) in complementary PWM mode.

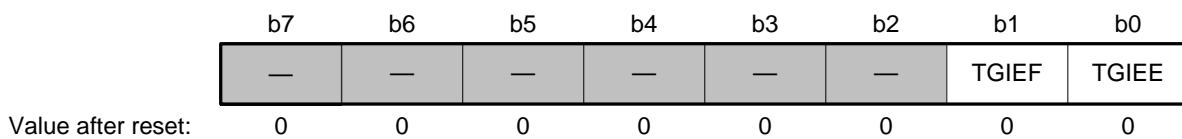
In channels 0 to 3 and 6 to 9, this bit is reserved. It is always read as 0. The write value should be 0.

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, MTU6)

Addresses: MTU0.TIER2 0008 8724h, MTU6.TIER2 0008 8B24h



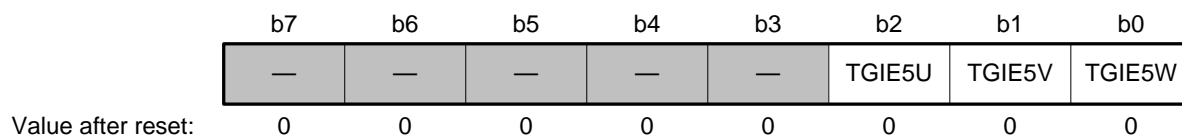
Bit	Symbol	Bit Name	Description	R/W
b0	TGIEE	TGR Interrupt Enable E	0: Interrupt requests (TGIE) disabled 1: Interrupt requests (TGIE) enabled	R/W
b1	TGIEF	TGR Interrupt Enable F	0: Interrupt requests (TGIF) disabled 1: Interrupt requests (TGIF) enabled	R/W
b7 to b2	—	Reserved	These bits are always read 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 MTUn.TCNT and MTUn.TGRm (n = 0 or 6, m = E or F).

- TIER (MTU5, MTU11)

Addresses: MTU5.TIER 0008 88B2h, MTU11.TIER 0008 8CB2h



Bit	Symbol	Bit Name	Description	R/W
b0	TGIE5W	TGR Interrupt Enable 5W	0: Interrupt requests TGInW disabled 1: Interrupt requests TGInW enabled	R/W
b1	TGIE5V	TGR Interrupt Enable 5V	0: Interrupt requests TGInV disabled 1: Interrupt requests TGInV enabled	R/W
b2	TGIE5U	TGR Interrupt Enable 5U	0: Interrupt requests TGInU disabled 1: Interrupt requests TGInU enabled	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note : Only unit 0 is described here because channel 11 (unit 1) has the same functions as those of channel 5 (unit 0).

[Legend] n = 5 or 11

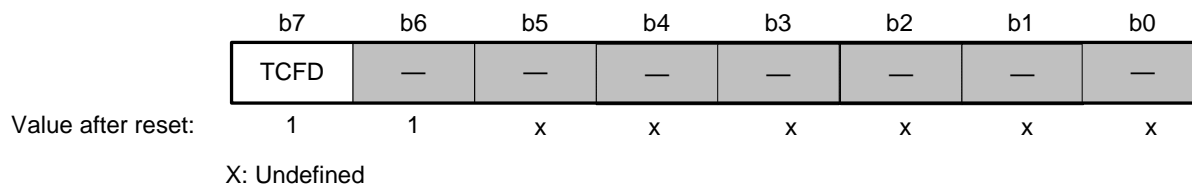
TGIE5W, TGIE5V, and TGIE5U Bits (TGR Interrupt Enable 5m)

Each bit enables or disables interrupt requests (TGIm) (m = U, V, or W).

18.2.6 Timer Status Register (TSR)

- TSR (MTU0 to MTU4, MTU6 to MTU10)

Addresses: MTU0.TSR 0008 8705h, MTU1.TSR 0008 8785h, MTU2.TSR 0008 8805h, MTU3.TSR 0008 862Ch, MTU4.TSR 0008 862Dh, MTU6.TSR 0008 8B05h, MTU7.TSR 0008 8B85h, MTU8.TSR 0008 8C05h, MTU9.TSR 0008 8A2Ch, MTU10.TSR 0008 8A2Dh



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are always read as undefined. The write value should be 1.	R/W
b6	—	Reserved	This bit is always read as 1. The write value should be 1.	R/W
b7	TCFD	Count Direction Flag	0: TCNT counts down 1: TCNT counts up	R

TSR is an 8-bit readable/writable register that indicates the status of each channel. The MTU has a total of 10 TSR registers, two each for channels 0 and 6 and one each for channels 1 to 4 and 7 to 10.

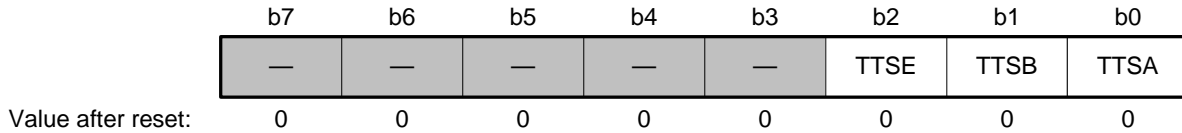
TCFD Flag (Count Direction Flag)

Status flag that shows the direction in which TCNT counts in channels 1 to 4 and 7 to 10.

In channels 0 and 6, this bit is reserved. It is always read as 1. The write value should be 1.

18.2.7 Timer Buffer Operation Transfer Mode Register (TBTM)

Addresses: MTU0.TBTM 0008 8726h, MTU3.TBTM 0008 8638h, MTU4.TBTM 0008 8639h,
MTU6.TBTM 0008 8B26h, MTU9.TBTM 0008 8A38h, MTU10.TBTM 0008 8A39h



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 channel 0 or 6, data is transferred from MTUn.TGRF to MTUn.TGRE 1: When MTUn.TCNT is cleared in channel 0 or 6, data is transferred from MTUn.TGRF to MTUn.TGRE (n = 0 or 6)	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

TBTM is an 8-bit readable/writable register that specifies the timing for transferring data from the buffer register to the timer general register in PWM mode. The MTU has a total of six TBTM registers, one each for channels 0, 3, 4, 6, 9, and 10.

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 MTUn.TGRF to MTUn.TGRE when they are used together for buffer operation.

In channels 3, 4, 9, and 10, this bit is reserved. It is always read as 0 and the write value should be 0. When channel 0 or 6 is not set to PWM mode, do not set the TTSE bit to 1.

(n = 0 or 6)

18.2.8 Timer Input Capture Control Register (TICCR)

Addresses: MTU1.TICCR 0008 8790h, MTU7.TICCR 0008 8B90h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	I2BE	I2AE	I1BE	I1AE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	I1AE	Input Capture Enable	0: Does not include the MTIOCjA pin in the MTU2.TGRA input capture conditions 1: Includes the MTIOCjA pin in the MTU2.TGRA input capture conditions	R/W
b1	I1BE	Input Capture Enable	0: Does not include the MTIOCjB pin in the MTU2.TGRB input capture conditions 1: Includes the MTIOCjB pin in the MTU2.TGRB input capture conditions	R/W
b2	I2AE	Input Capture Enable	0: Does not include the MTIOcKA pin in the MTU1.TGRA input capture conditions 1: Includes the MTIOcKA pin in the MTU1.TGRA input capture conditions	R/W
b3	I2BE	Input Capture Enable	0: Does not include the MTIOcKB pin in the MTU1.TGRB input capture conditions 1: Includes the MTIOcKB pin in the MTU1.TGRB input capture conditions	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

[Legend] j = 1 or 7, k = 2 or 8

Note: Only unit 0 is described here because channels 7 and 8 (unit 1) have the same functions as those of channels 1 and 2 (unit 0), respectively.

MTICCR specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded. The MTU has one MTICCR each for channels 1 and 7.

18.2.9 Timer A/D Converter Start Request Control Register (TADCR)

Address: MTU4.TADCR 0008 8640h, MTU10.TADCR 0008 8A40h

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*

Note: * Do not set bits 6, and 4 to 0 to 1 when complementary PWM mode is not selected.

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
b1	ITB3AE	TGIA3 Interrupt Skipping Link Enable	0: TGI3A interrupt skipping is not linked 1: TGI3A interrupt skipping is linked	R/W
b2	ITA4VE	TCIV4 Interrupt Skipping Link Enable	0: TCIV4 interrupt skipping is not linked 1: TCIV4 interrupt skipping is linked	R/W
b3	ITA3AE	TGIA3 Interrupt Skipping Link Enable	0: TGI3A interrupt skipping is not linked 1: TGI3A interrupt skipping is linked	R/W
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
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
b7	UT4AE	Up-Count TRG4AN Enable	0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT up-count operation 1: A/D converter up requests (TRG4AN) enabled during MTU4.TCNT down-count operation	R/W
b13 to b8	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b15, b14	BF[1:0]	MTU4.TADCOBRA/B Transfer Timing Select	See Table 18.31 for details.	R/W

Note 1. Since channels 4 and 10 have the same functionality, the explanation here is only for unit 0.

Note 2. Access to TADCR in eight-bit units is prohibited. Always access this register in 16-bit units.

Note 3. While interrupt skipping is prohibited, i.e. while the T3AEN and T4VEN bits and the skipping count setting bits (T3ACOR and T4VCOR) in the timer interrupt skipping set register (TITCR) are set to "0", do not set up interlocking with interrupt skipping, i.e. set the ITA3AE, ITA4VE, ITB3AE, or ITB4VE bits in the timer A/D conversion start request register (TADCR) to "0".

Note 4. Requests to start A/D conversion are not issued if the setting for interlocking with interrupt skipping is made while interrupt skipping is prohibited.

TADCR is a 16-bit readable/writable register that enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping operation. The MTU has one TADCR each for channels 4 and 10.

Table 18.31 Setting of Transfer Timing by BF1 and BF0 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 MTUn.TCNT count. *1
1	0	Transfers data from the cycle set buffer register to the cycle set register at the trough of the MTUn.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 MTUn.TCNT count. *2

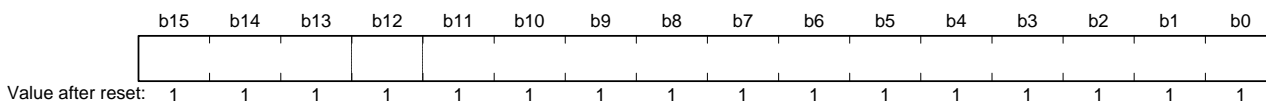
[Legend] n = 4 or 10, m = 3 or 9

Note 1. Data is transferred from the cycle set buffer register to the cycle set register when the crest of the MTUn.TCNT count is reached in complementary PWM mode, when a compare match occurs between MTUm.TCMT and MTUm.TGRA in reset-synchronized PWM mode, or when a compare match occurs between MTUn.TCNT and MTUn.TGRA in PWM mode 1 or normal operation mode.

Note 2. These settings are prohibited when complementary PWM mode is not selected.

18.2.10 Timer A/D Converter Start Request Cycle Set Register (TADCORA, TADCORB)

Addresses: MTU4.TADCORA 0008 8644h, MTU4.TADCORB 0008 8646h, MTU10.TADCORA 0008 8A44h, MTU10.TADCORB 0008 8A46h



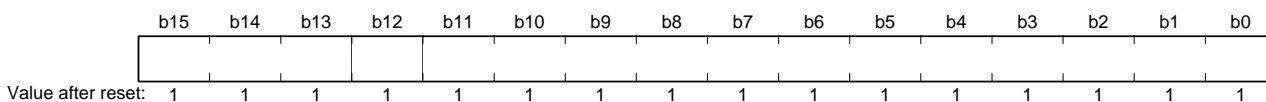
Note: TADCORA/B must not be accessed in eight bits; they should always be accessed in 16 bits.

TADCORA and TADCORB are 16-bit readable/writable registers. When the MTUn.TCNT (n = 4 or 10) count reaches the value in TADCORA or TADCORB, a corresponding A/D converter start request will be issued.

The TADCORA and TADCORB values after reset are FFFFh.

18.2.11 Timer A/D Converter Start Request Cycle Set Buffer Register (TADCOBRA, TADCOBRB)

Addresses: MTU4.TADCOBRA 0008 8648h, MTU4.TADCOBRB 0008 864Ah, MTU10.TADCOBRA 0008 8A48h, MTU10.TADCOBRB 0008 8A4Ah



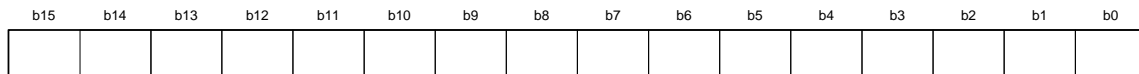
Note: TADCORA_4 and TADCORB_4 must not be accessed in eight bits; they should always be accessed in 16 bits.

TADCOBRA and TADCOBRB are 16-bit readable/writable registers. When the crest or trough of the MTU4/10.TCNT count is reached, these register values are transferred to TADCORA and TADCORB, respectively.

The TADCOBRA and TADCOBRB values after reset are FFFFh.

18.2.12 Timer Counter (TCNT)

Addresses: 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, MTU6.TCNT 0008 8B06h, MTU7.TCNT 0008 8B86h, MTU8.TCNT 0008 8C06h, MTU9.TCNT 0008 8A10h, MTU10.TCNT 0008 8A12h, MTU11.TCNTU 0008 8C80h, MTU11.TCNTV 0008 8C90h, MTU11.TCNTW 0008 8CA0h



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

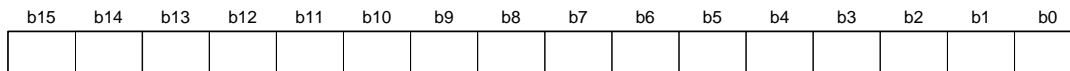
Note: The TCNT counters must not be accessed in eight bits; they should always be accessed in 16 bits.

TCNT is a 16-bit readable/writable counter. The MTU has a total of 16 TCNT counters, one each for channels 0 to 4 and 6 to 10 and three (MTUn.TCNTU, TCNTUV, and TCNTUW) for channels 5 and 11 (n = 5 or 11).

TCNT is initialized to 0000h by a reset.

18.2.13 Timer General Register (TGR)

Addresses: 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, MTU6.TGRA 0008 8B08h, MTU6.TGRB 0008 8B0Ah, MTU6.TGRC 0008 8B0Ch, MTU6.TGRD 0008 8B0Eh, MTU6.TGRE 0008 8B20h, MTU6.TGRF 0008 8B22h, MTU7.TGRA 0008 8B88h, MTU7.TGRB 0008 8B8Ah, MTU8.TGRA 0008 8C08h, MTU8.TGRB 0008 8C0Ah, MTU9.TGRA 0008 8A18h, MTU9.TGRB 0008 8A1Ah, MTU9.TGRC 0008 8A24h, MTU9.TGRD 0008 8A26h, MTU10.TGRA 0008 8A1Ch, MTU10.TGRB 0008 8A1Eh, MTU10.TGRC 0008 8A28h, MTU10.TGRD 0008 8A2Ah, MTU11.TGRU 0008 8C82h, MTU11.TGRV 0008 8C92h, MTU11.TGRW 0008 8CA2h



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

Note: The TGR registers must not be accessed in eight bits; they should always be accessed in 16 bits. TGR registers are initialized to FFFFh.

TGR is a 16-bit readable/writable register. The MTU has a total of 42 TGR registers, six each for channels 0 and 6, two each for channels 1, 2, 7, and 8, four each for channels 3, 4, 9, and 10, and three for channels 5 and 11.

TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture registers. TGRC and TGRD for channels 0, 3, 4, 6, 9, and 10 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.

MTUn.TGRE and MTUn.TGRF function as compare match registers. When the MTUn.TCNT count matches the MTUn.TGRE value, an A/D converter start request can be issued (n = 0). TGRF can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF.

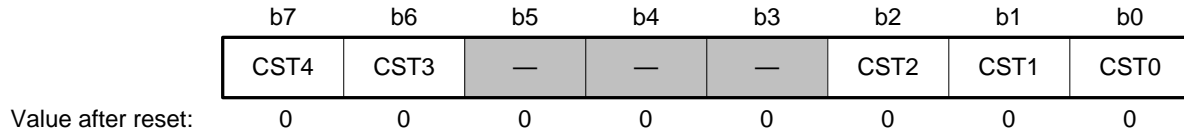
MTUm.TGRU, MTUm.TGRV, and MTUm.TGRW function as compare match, input capture, or external pulse width measurement registers (m = 5 or 11).

Furthermore, the interval for issuing requests to start A/D conversion can be set in this register. Refer to section 34, 12-Bit A/D Converter (S12AD), and section 35, 10-Bit A/D Converter (ADa), for details of the conditions for starting A/D conversion.

18.2.14 Timer Start Register (TSTR)

- MTUA.TSTR (MTU0 to MTU4), MTUB.TSTR (MTU6 to MTU10)

Addresses: MTUA.TSTR 0008 8680h, MTUB.TSTR 0008 8A80h



Bit	Symbol	Bit Name	Description	R/W
b0	CST0	Counter Start 0	0: MTU0.TCNT count operation is stopped 1: MTU0.TCNT performs count operation	R/W
b1	CST1	Counter Start 1	0: MTU1.TCNT count operation is stopped 1: MTU1.TCNT performs count operation	R/W
b2	CST2	Counter Start 2	0: MTU2.TCNT count operation is stopped 1: MTU2.TCNT performs count operation	R/W
b5 to b3	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b6	CST3	Counter Start 3	0: MTU3.TCNT count operation is stopped 1: MTU3.TCNT performs count operation	R/W
b7	CST4	Counter Start 4	0: MTU4.TCNT count operation is stopped 1: MTU4.TCNT performs count operation	R/W

Note : Only unit 0 is described here because channels 6 to 10 (unit 1) have the same functions as those of channels 0 to 4 (unit 0), respectively.

TSTRA starts or stops TCNT operation in MTU0 to MTU4.

TSTRB starts or stops TCNT operation in MTU6 to MTU10.

Before setting the operating mode in TMDR or setting the TCNT count clock in TCR, be sure to stop the TCNT counter.

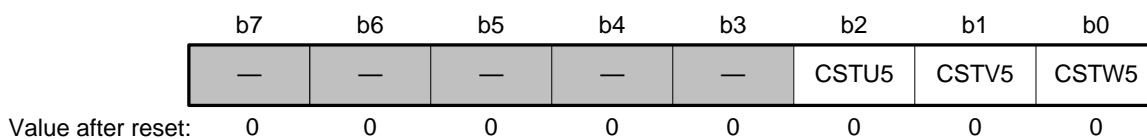
CST Bits (Counter Start)

Each bit starts or stops TCNT in the corresponding channel.

If 0 is written to the CST 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 CST bit is 0, the pin output level will be changed to the specified initial output value.

- MTUA.TSTR (MTU5), MTUB.TSTR (MTU11)

Addresses: MTUA.TSTR 0008 88B4h, MTUB.TSTR 0008 8CB4h

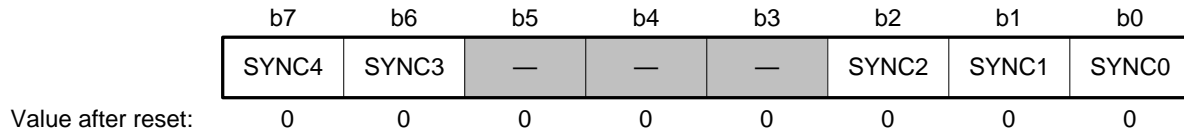


Bit	Symbol	Bit Name	Description	R/W
b0	CSTW5	Counter Start W5	0: MTU5.TCNTW count operation is stopped 1: MTU5.TCNTW performs count operation	R/W
b1	CSTV5	Counter Start V5	0: MTU5.TCNTV count operation is stopped 1: MTU5.TCNTV performs count operation	R/W
b2	CSTU5	Counter Start U5	0: MTU5.TCNTU count operation is stopped 1: MTU5.TCNTU performs count operation	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note : Only unit 0 is described here because channel 11 (unit 1) has the same functions as those of channel 5 (unit 0).

18.2.15 Timer Synchronous Register (TSYR)

Addresses: MTUA.TSYR 0008 8681h, MTUB.TSYR 0008 8A81h



Bit	Symbol	Bit Name	Description	R/W
b0	SYNC0	Timer Synchronous Operation 0	0: MTU0.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU0.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b1	SYNC1	Timer Synchronous Operation 1	0: MTU1.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU1.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b2	SYNC2	Timer Synchronous Operation 2	0: MTU2.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU2.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b5 to b3	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b6	SYNC3	Timer Synchronous Operation 3	0: MTU3.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b7	SYNC4	Timer Synchronous Operation 4	0: MTU4.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W

Note : Only unit 0 is described here.

TSYRA selects independent operation or synchronous operation of TCNT in MTU0 to MTU4.

TSYRB selects independent operation or synchronous operation of TCNT in MTU6 to MTU10.

A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

SYNC Bits (Timer Synchronous Operation)

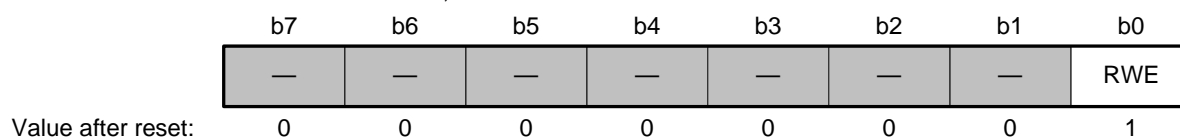
Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous presetting of multiple channels and synchronous clearing by counter clearing on another channel are possible.

To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR[2:0] in TCR.

18.2.16 Timer Read/Write Enable Register (TRWER)

Addresses: MTUA.TRWER 0008 8684h, MTUB.TRWER 0008 8A84h



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 always read as 0. The write value should be 0.	R/W

TRWERA enables or disables access to the registers and counters that have write-protection capability against accidental modification in channels 3 and 4.

TRWERB enables or disables access to the registers and counters that have write-protection capability against accidental modification in channels 9 and 10.

RWE Bit (Read/Write Enable)

This bit enables or disables access to the registers that have write-protection capability against accidental modification.

[Clearing condition]

- When 0 is written to the RWE bit after reading RWE = 1

- Registers and Counters having Write-Protection Capability against Accidental Modification

44 registers: MTUn.TCR, MTUn.TMDR, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTUj.TOER, MTUj.TOCR1, MTUj.TOCR2, MTUj.TGCR, MTUj.TCDR, MTUj.TDDR, and MTUn.TCNT (n = 3, 4, 9, or 10, j = A or B)

18.2.17 Timer Output Master Enable Register (TOER)

Addresses: MTUA.TOER 0008 860Ah, MTUB.TOER 0008 8A0Ah

	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 always read as 1. The write value should be 1.	R/W

Note 1. When making the “disabled” setting for MTU output, if the non-active level is to be output on all pins, be sure to set the data direction register (DDR) and data register (DR) of the I/O port for output of the non-active level in advance. For details, see section 17, I/O Ports.

TOERA enables or disables output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

TOERB enables or disables output settings for output pins MTIOC10D, MTIOC10C, MTIOC9D, MTIOC10B, MTIOC10A, and MTIOC9B.

These pins do not output correctly if the TOER bits have not been set. In channels 3, 4, 9, and 10, set TOER prior to setting TIOR.

Set the MTUA.TOER register after setting the CST3 and CST4 bits in the MTUA.TSTR register. Set the MTUB.TOER register after setting the CST0 and CST1 bits in the MTUB.TSTR register (see figure 18.36 and figure 18.39).

18.2.18 Timer Output Control Register 1 (TOCR1)

Addresses: MTUA.TOCR1 0008 860Eh, MTUB.TOCR1 0008 8A0Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
Value after reset:	0	0	0	0	0*	0	0	0

Note: * 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.

Bit	Symbol	Bit Name	Description	R/W
b0	OLSP	Output Level Select P ^{*2, *3}	See Table 18.32.	R/W
b1	OLSN	Output Level Select N ^{*2, *3}	See Table 18.33.	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
b5, b4	—	Reserved	These bits are always 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 always 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 inverse-phase output will be the exact inverse of the positive-phase output. In this case, only the OLSP bit is valid.

TOCR1 are 8-bit readable/writable registers that enable or disable PWM-synchronized toggle output in complementary PWM mode and reset-synchronized PWM mode, and control inversion of PWM output level.

Use MTUA.TOCR1 for unit 0 and MTUB.TOCR1 for unit 1.

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 18.32 Output Level Select Function

Bit 0	Function			
	OLSP	Initial Output	Active Level	Compare Match Output
				Compare Match Output
			Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 18.33 Output Level Select Function

Bit 1	Function			
	OLSN	Initial Output	Active Level	Compare Match Output
				Compare Match Output
			Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Figure 18.3 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

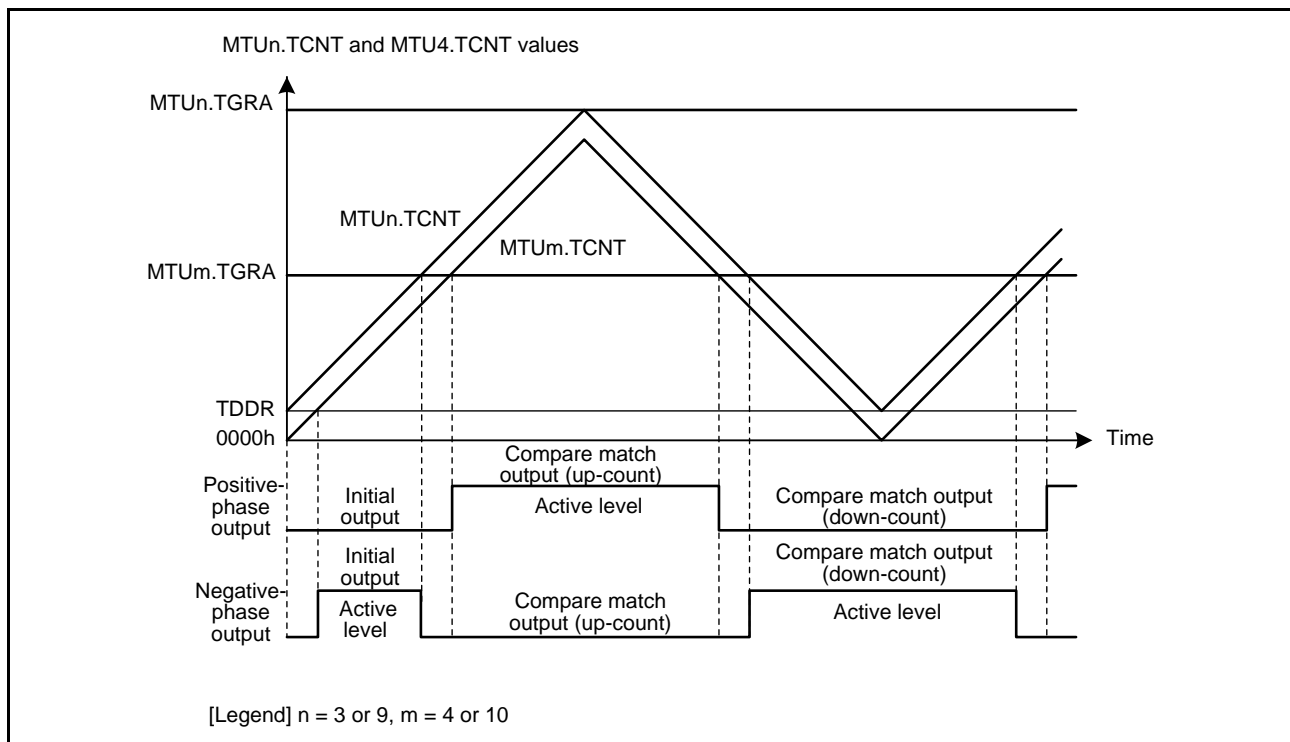
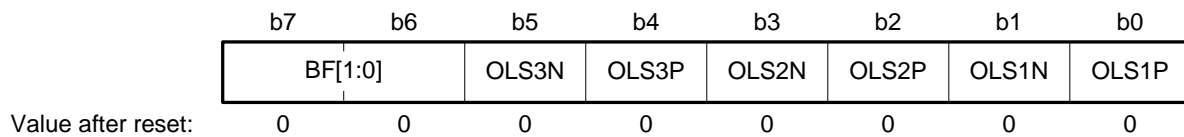


Figure 18.3 Example of Output in Complementary PWM Mode

18.2.19 Timer Output Control Register 2 (TOCR2)

Addresses: MTUA.TOCR2 0008 860Fh, MTUB.TOCR2 0008 8A0Fh



Bit	Symbol	Bit Name	Description	R/W
b0	OLS1P	Output Level Select 1P ^{*1, *2}	This bit selects the output level on MTIOC3B in reset-synchronized PWM mode and complementary PWM mode. See Table 18.34.	R/W
b1	OLS1N	Output Level Select 1N ^{*1, *2}	This bit selects the output level on MTIOC3D in reset-synchronized PWM mode and complementary PWM mode. See Table 18.35.	R/W
b2	OLS2P	Output Level Select 2P ^{*1, *2}	This bit selects the output level on MTIOC4A in reset-synchronized PWM mode and complementary PWM mode. See Table 18.36.	R/W
b3	OLS2N	Output Level Select 2N ^{*1, *2}	This bit selects the output level on MTIOC4C in reset-synchronized PWM mode and complementary PWM mode. See Table 18.37.	R/W
b4	OLS3P	Output Level Select 3P ^{*1, *2}	This bit selects the output level on MTIOC4B in reset-synchronized PWM mode and complementary PWM mode. See Table 18.38.	R/W
b5	OLS3N	Output Level Select 3N ^{*1, *2}	This bit selects the output level on MTIOC4D in reset-synchronized PWM mode and complementary PWM mode. See Table 18.39.	R/W
b7, b6	BF[1:0]	TOLBR Buffer Transfer Timing Select	These bits select the timing for transferring data from TOLBRj to TOCR2. See Table 18.40 for details.	R/W

Note 1. Since channels 3 and 9 and channels 4 and 10 have the same functions, so they are explained here by taking unit 0 as an example. This setting is valid when the TOCR1.TOCS bit is set to 1.

Note 2. If dead time is not generated, the inverse-phase output will be the exact inverse of the positive-phase output. In this case, only the OLSiP (i = 1, 2, or 3) bit is valid.

TOCR2 control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Use MTUA.TOCR2 for unit 0 and MTUB.TOCR2 for unit 1.

Table 18.34 MTIOcM_B Output Level Select Function

Bit 0	Function		Compare Match Output	
			Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

[Legend] m = 3 or 9

Table 18.35 MTIOcM Output Level Select Function

Bit 1	Function		Compare Match Output	
			Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

[Legend] m = 3 or 9

Note : The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 18.36 MTIOcN Output Level Select Function

Bit 2	Function		Compare Match Output	
			Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

[Legend] n = 4 or 10

Table 18.37 MTIOcC Output Level Select Function

Bit 3	Function		Compare Match Output	
			Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

[Legend] n = 4 or 10

Note : The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 18.38 MTIOcM Output Level Select Function

Bit 4	Function		Compare Match Output	
			Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

[Legend] m = 3 or 9

Table 18.39 MTIOcMD Output Level Select Function

Bit 5	Function			
	OLS3N	Initial Output	Active Level	Compare Match Output
Up-Counting				Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

[Legend] m = 3 or 9

Note : The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

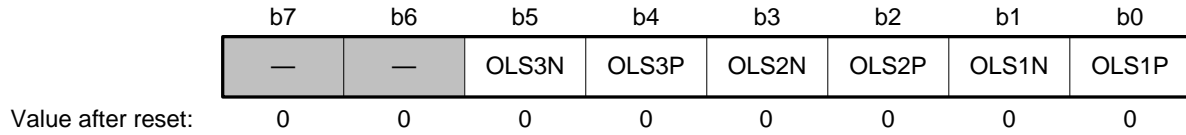
Table 18.40 Setting of TOCR2.BF[1:0] Bits

Bit 7	Bit 6	Description	
BF1	BF0	Complementary PWM Mode	Reset-Synchronized PWM Mode
0	0	Does not transfer data from the buffer register (TOLBRj) to TOCR2.	Does not transfer data from the buffer register (TOLBR) to TOCR2.
0	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest of the MTUn.TCNT count.	Transfers data from the buffer register (TOLBR) to TOCR2 when MTUn.TCNT or MTUm.TCNT is cleared.
1	0	Transfers data from the buffer register (TOLBR) to TOCR2 at the trough of the MTUn.TCNT count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest and trough of the MTUn.TCNT count.	Setting prohibited

[Legend] n = 4 or 10, m = 3 or 9

18.2.20 Timer Output Level Buffer Register (TOLBR)

Addresses: MTUA.TOLBR 0008 8636h, MTUB.TOLBR 0008 8A36h



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 always read as 0. The write value should be 0.	R/W

[Legend] Only unit 0 is described here.

TOLBR are 8-bit readable/writable registers that function as buffer registers for TOCR2 and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 18.4 shows an example of the PWM output level setting procedure in buffer operation.

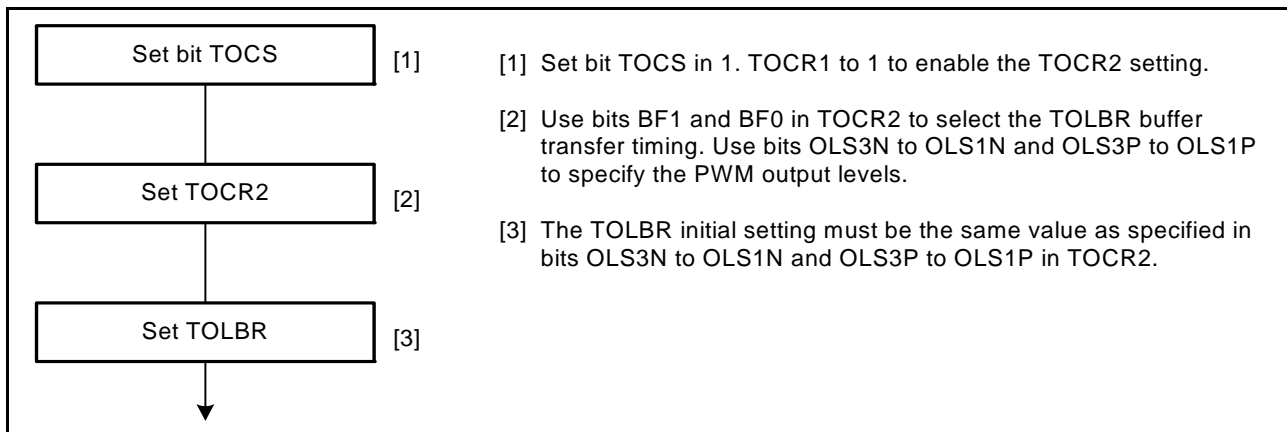


Figure 18.4 Example of PWM Output Level Setting Procedure in Buffer Operation

18.2.21 Timer Gate Control Register (TGCR)

Addresses: MTUA.TGCR 0008 860Dh, MTUB.TGCR 0008 8A0Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BDC	N	P	FB	WF	VF	UF
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	UF	Output Phase Switch	These bits turn on or off the positive-phase/negative-phase output. The setting of these bits is valid only when the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. See Table 18.41.	R/W
b1	VF			R/W
b2	WF			R/W
b3	FB	External Feedback Signal Enable	0: Output is switched by external input (input sources are TGRA, TGRB, and TGRC input capture signals in channel n (n = 0 or 6)) 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 always read as 1. The write value should be 1.	R/W

TGCR control the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. These register settings are ineffective for anything other than complementary PWM mode and reset-synchronized PWM mode.

UF, VF, and WF Bits (Output Phase Switch)

The setting of these bits is valid only when the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. See Table 18.41.

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 MTUn (n = 0 or 6) 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 (MTIOcNB, MTIOcMA, and MTIOcMB pins). (n = 3 or 9, m = 4 or 10)

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 (MTIOcND, MTIOcMC, and MTIOcMD pins). (n = 3 or 9, m = 4 or 10)

BDC Bit (Brushless DC Motor)

This bit selects whether to make the functions of TGCRj (j = A or B) effective or ineffective.

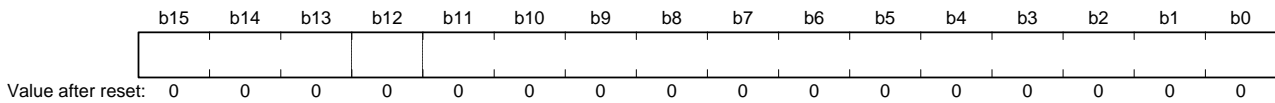
Table 18.41 Output Level Select Function

Bit 2	Bit 1	Bit 0	Function					
			MTIOCnB	MTIOCmA	MTIOCmB	MTIOCnD	MTIOCmC	MTIOCmD
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

[Legend] n = 3 or 9, m = 4 or 10

18.2.22 Timer Subcounter (TCNTS)

Addresses: MTUA.TCNTS 0008 8620h, MTUB.TCNTS 0008 8A20h



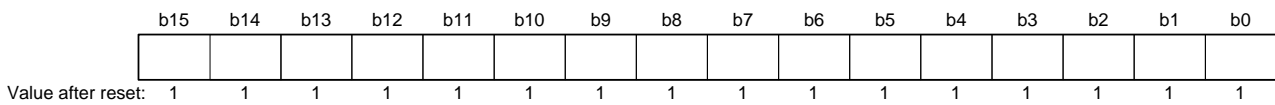
Note: Accessing the TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

TCNTS are 16-bit read-only counters that are used only in complementary PWM mode.

The TCNTS value after reset is 0000h.

18.2.23 Timer Dead Time Data Register (TDDR)

Addresses: MTUA.TDDR 0008 8616h, MTUB.TDDR 0008 8A16h

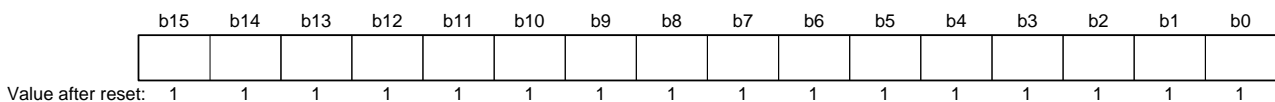


Note: Accessing the TDDR in 8-bit units is prohibited. Always access in 16-bit units.

TDDR are 16-bit registers, used only in complementary PWM mode, that specify the MTU3/9.TCNT and MTU4/10.TCNT counter offset value. In complementary PWM mode, when the MTU3/9.TCNT and MTU4/10.TCNT counters are cleared and then restarted, the TDDR value is loaded into the MTU3/9.TCNT counter and the count operation starts. The TDDR value after reset is FFFFh.

18.2.24 Timer Cycle Data Register (TCDR)

Addresses: MTUA.TCDR 0008 8614h, MTUB.TCDR 0008 8A14h



Note: Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

TCDR are 16-bit registers used only in complementary PWM mode. Set half the PWM carrier cycle as the TCDR value. TCDR is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (down-count to up-count). The TCDR value after reset is FFFFh.

18.2.25 Timer Cycle Buffer Register (TCBR)

Addresses: MTUA.TCBR 0008 8622h, MTUB.TCBR 0008 8A22h

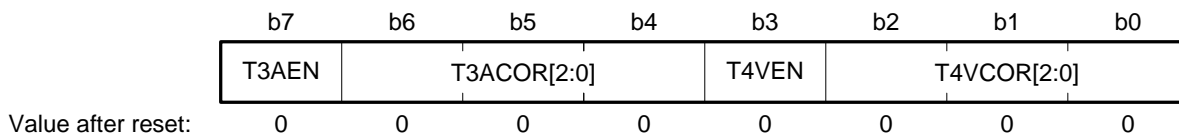


Note: Accessing the TCBR in 8-bit units is prohibited. Always access in 16-bit units.

TCBR are 16-bit registers, used only in complementary PWM mode, that function as buffer registers for TCDR. The TCBR value is transferred to TCDR with the transfer timing set in TMDR. The TCBR value after reset is FFFFh.

18.2.26 Timer Interrupt Skipping Set Register (TITCR)

Addresses: MTUA.TITCR 0008 8630h, MTUB.TITCR 0008 8A30h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCOR[2:0]	TCIV4 Interrupt Skipping Count Setting	These bits specify the TCIV4 interrupt skipping count within the range from 0 to 7. *1 For details, see Table 18.42.	R/W
b3	T4VEN	T4VEN	0: TCIV4 interrupt skipping disabled 1: TCIV4 interrupt skipping enabled	R/W
b6 to b4	T3ACOR[2:0]	TGIA3 Interrupt Skipping Count Setting	These bits specify the TGIA3 interrupt skipping count within the range from 0 to 7. *1,*2 For details, see Table 18.43.	R/W
b7	T3AEN	T3AEN	0: TGIA3 interrupt skipping disabled 1: TGIA3 interrupt skipping enabled	R/W

Note : * Only unit 0 is described here because channels 9 and 10 (unit 1) have the same functions as those of channels 3 and 4 (unit 0), respectively.

Note 1. Interrupt skipping will not be performed if 0 is specified as the interrupt skipping count.

Note 2. Before changing the interrupt skipping count, be sure to clear the skipping counter (TITCNT) by clearing the TITCR.T3AEN and TITCR.T4VEN bits to 0.

TITCR enable or disable interrupt skipping and specify the interrupt skipping count. The MTU has two TITCR registers.

T4VCOR[2:0] Bits (TCIV4 Interrupt Skipping Count Setting)

T3ACOR[2:0] Bits (TGIA3 Interrupt Skipping Count Setting)

These bits specify the TCIVm and TGIAn interrupt skipping count within the range from 0 to 7. For details, see Table 18.42 and Table 18.43.

Table 18.42 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 skip TCIVm interrupts.
0	0	1	Sets the TCIVm interrupt skipping count to 1.
0	1	0	Sets the TCIVm interrupt skipping count to 2.
0	1	1	Sets the TCIVm interrupt skipping count to 3.
1	0	0	Sets the TCIVm interrupt skipping count to 4.
1	0	1	Sets the TCIVm interrupt skipping count to 5.
1	1	0	Sets the TCIVm interrupt skipping count to 6.
1	1	1	Sets the TCIVm interrupt skipping count to 7.

[Legend] m = 4 or 10

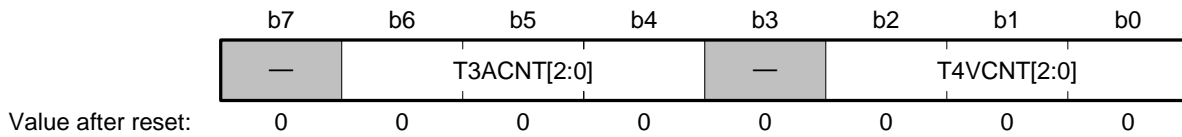
Table 18.43 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits

Bit 6	Bit 5	Bit 4	Description
T3ACOR2	T3ACOR1	T3ACOR0	
0	0	0	Does not skip TGIAn interrupts.
0	0	1	Sets the TGIAn interrupt skipping count to 1.
0	1	0	Sets the TGIAn interrupt skipping count to 2.
0	1	1	Sets the TGIAn interrupt skipping count to 3.
1	0	0	Sets the TGIAn interrupt skipping count to 4.
1	0	1	Sets the TGIAn interrupt skipping count to 5.
1	1	0	Sets the TGIAn interrupt skipping count to 6.
1	1	1	Sets the TGIAn interrupt skipping count to 7.

[Legend] n = 3 or 9

18.2.27 Timer Interrupt Skipping Counter (TITCNT)

Addresses: MTUA.TITCNT 0008 8631h, MTUB.TITCNT 0008 8A31h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCNT[2:0]	TCIV4 Interrupt Counter	While the T4VEN bit in TITCR is set to 1, the count in these bits is incremented every time a TCIV4 interrupt occurs.	R
b3	—	Reserved	This bit is always read as 0. Writing to this bit has no effect.	R
b6 to b4	T3ACNT[2:0]	TGIA3 Interrupt Counter	While the T3AEN bit in TITCR is set to 1, the count in these bits is incremented every time a TGIA3 interrupt occurs.	R
b7	—	Reserved	This bit is always read as 0. Writing to this bit has no effect.	R

Note : Only unit 0 is described here because channels 9 and 10 (unit 1) have the same functions as those of channels 3 and 4 (unit 0), respectively.
To clear the TITCNT, clear the T3AEN and T4VEN bits in TITCR to 0.

TITCNT are 8-bit readable/writable counters. The MTU has two TITCNT counters. TITCNT retain their values even after stopping the count operation of MTUn.TCNT and MTUm.TCNT. (n = 3 or 9, m = 4 or 10)

T4VCNT[2:0] Bits (TCIV4 Interrupt Counter)

[Clearing conditions]

- When the T4VCNT[2:0] bits in TITCNT match the T4VCR[2:0] bits in TITCR
- When the T4VEN bit in TITCR is cleared to 0
- When the T4VCOR[2:0] bits in TITCR are cleared to 000b

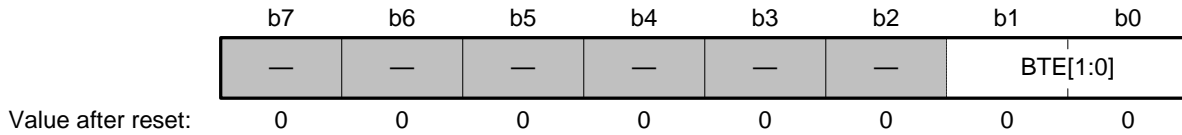
T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)

[Clearing conditions]

- When the T3ACNT[2:0] bits in TITCNT match the T3ACOR[2:0] bits in TITCR
- When the T3AEN bit in TITCR is cleared to 0
- When the T3ACOR2 to T3ACOR0 bits in TITCR are cleared to 000b

18.2.28 Timer Buffer Transfer Set Register (TBTER)

Address: MTUA.TBTER 0008 8632h, MTUB.TBTER 0008 8A32h



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 18.44 for details.	R/W
b7 to b2	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note : * Applicable buffer registers: MTU3.TGRC, MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, and MTUj.TCBR (j = A, B)

TBTER are 8-bit readable/writable registers that enable or disable transfer from the buffer registers used in complementary PWM mode to the temporary registers and specify whether to link the transfer with interrupt skipping operation. The MTU has two TBTER registers.

Table 18.44 Setting of TBTER.BTE[1:0] Bits

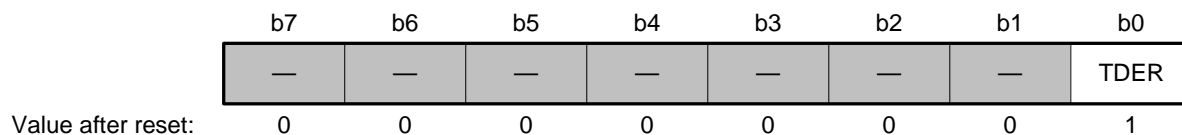
Bit 1	Bit 0	Description
BTE1	BTE0	
0	0	Enables transfer from the buffer registers to the temporary registers* ¹ and does not link the transfer with interrupt skipping operation.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping operation.* ²
1	1	Setting prohibited

Note 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For details, refer to section 18.3.8, Complementary PWM Mode .

Note 2. When interrupt skipping is disabled (the T3AEN and T4VEN bits are cleared to 0 in the timer interrupt skipping set register (TITCR) or the skipping count set bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), disable link of buffer transfer with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

18.2.29 Timer Dead Time Enable Register (TDER)

Addresses: MTUA.TDER 0008 8634h, MTUB.TDER 0008 8A34h



Bit	Symbol	Bit Name	Description	R/W
b0	TDER	Dead Time Enable	0: No dead time is generated 1: Dead time is generated *	R/(W)
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note : * TDDR must be set to 1 or a larger value.

TDER are 8-bit readable/writable registers that control dead time generation in complementary PWM mode. The MTU has two TDER registers; MTUA.TDER for channel 3 and MTUB.TDER for channel 9. TDER should be modified only while TCNT stops.

- **TDER Bit (Dead Time Enable)**

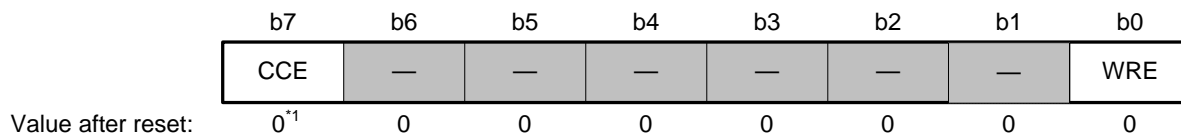
This bit specifies whether to generate dead time.

[Clearing condition]

- When 0 is written to TDER after reading TDER = 1

18.2.30 Timer Waveform Control Register (TWCR)

Addresses: MTUA.TWCR 0008 8660h MTUB.TWCR 0008 8A60h



Note 1. Do not set to 1 when complementary PWM mode is not selected.

Bit	Symbol	Bit Name	Description	R/W
b0	WRE	Initial Output Suppression Enable	0: Initial value specified in TOCR is output 1: Initial output is suppressed.	R/(W)
b6 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b7	CCE	Compare Match Clear Enable	0: Counters are not cleared at MTUn.TGRA compare match 1: Counters are cleared at MTUn.TGRA compare match	R/(W)

[Legend] n = 3 or 9

TWCR are 8-bit readable/writable registers. MTUA.TWCR controls the output waveform when synchronous counter clearing occurs in MTU3.TNCT and MTU4.TNCT in complementary PWM mode and specifies whether to clear the counters at MTU3.TGRA compare match. MTUB.TWCR controls the output waveform when synchronous counter clearing occurs in MTU9.TNCT and MTU10.TNCT in complementary PWM mode and specifies whether to clear the counters at MTU9.TGRA compare match.

The CCE bit and WRE bit in TWCR should be modified only while TCNT stops.

WRE Bit (Initial Output Suppression Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.

The initial output is only suppressed if synchronous clearing occurs within the Tb interval of the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial value specified in TOCR is output regardless of the TWCR.WRE bit setting. The initial value specified in TOCR is also output when synchronous clearing occurs in the Tb interval at the trough immediately after MTUn.TCNT and MTUm.TCNT start operation. (n = 3 or 9, m = 4 or 10)

For the Tb interval at the trough in complementary PWM mode, see Figure 18.42.

[Setting condition]

- When 1 is written to TWCR.WRE after reading TWCR.WRE = 0

CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at TGRAn compare match in complementary PWM mode. (n = 3 or 9)

[Setting condition]

- When 1 is written to CCE after reading CCE = 0

18.2.31 Bus Master Interface

The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer cycle buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register (TCDR), timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set register (TADCOR), and timer A/D converter start request cycle set buffer register (TADCOBR) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/write access. 8-bit read/write is not allowed. Always access the registers in 16-bit units.

All registers other than the above registers are 8-bit registers, so read/write access should be performed in 8-bit units.

(j = A or B)

18.3 Operation

18.3.1 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 MTUn.TSTR is set to 1, TCNT for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example. (n = 5 or 11)

(a) Example of Count Operation Setting Procedure

Figure 18.5 shows an example of the count operation setting procedure.

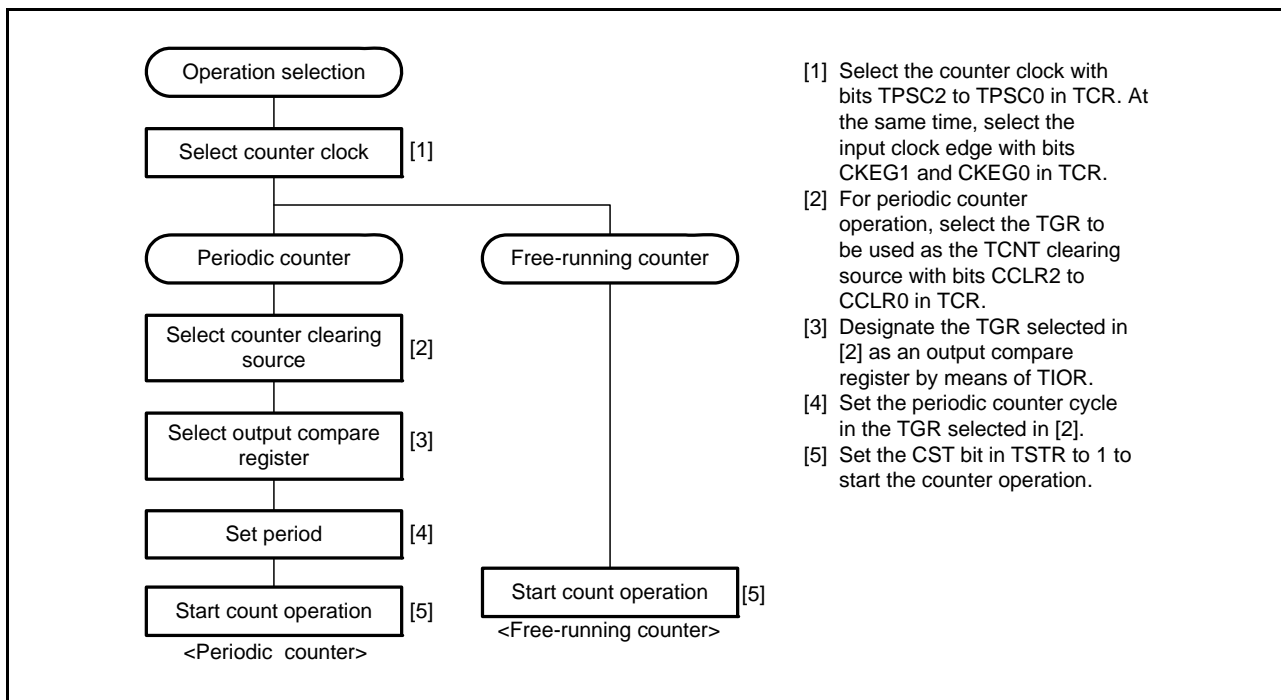


Figure 18.5 Example of Counter Operation Setting Procedure

(b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the MTU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1, the corresponding TCNT counter starts up-count operation as a free-running counter.

When TCNT overflows (from FFFFh to 0000h), the MTU requests an interrupt if the corresponding TCIEV bit in TIER is 1. After an overflow, TCNT starts counting up again from 0000h.

Figure 18.6 illustrates free-running counter operation.

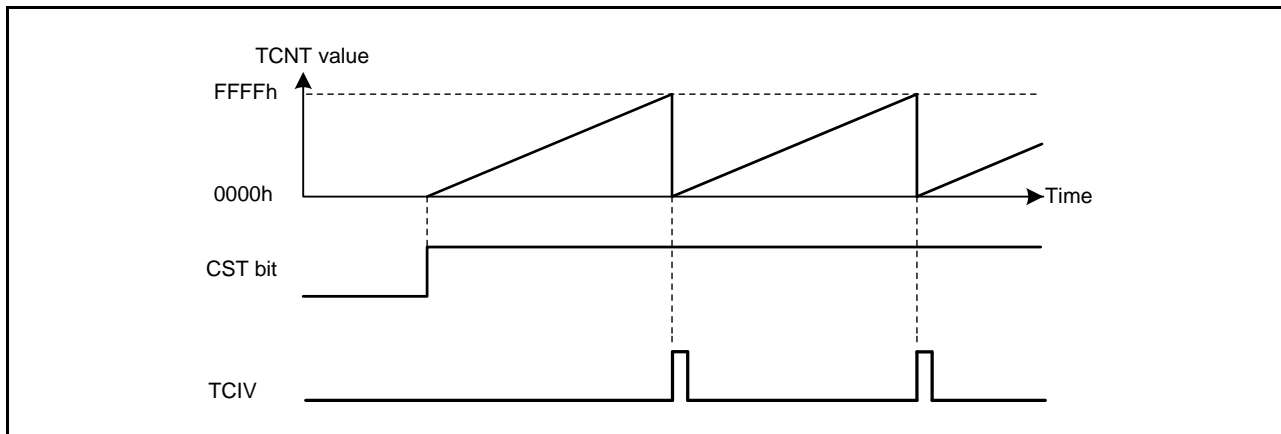


Figure 18.6 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the cycle is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR[2:0] in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count matches the value in TGR, TCNT is cleared to 0000h.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the MTU requests an interrupt. After a compare match, TCNT starts counting up again from 0000h.

Figure 18.7 illustrates periodic counter operation.

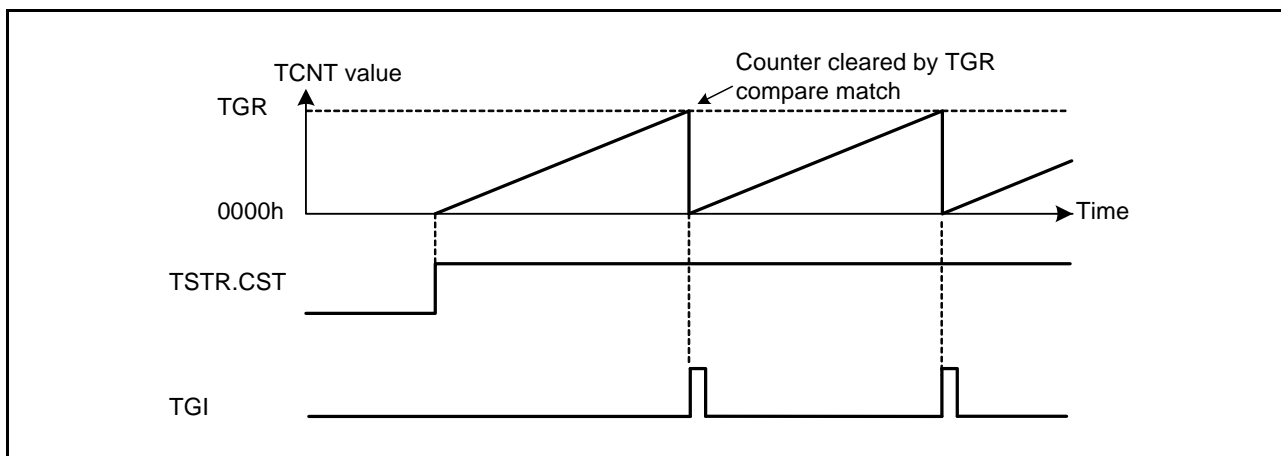


Figure 18.7 Periodic Counter Operation

(2) Waveform Output by Compare Match

The MTU can low output or high output or toggles output from the corresponding output pin using compare match.

(a) Example of Procedure for Setting Waveform Output by Compare Match

Figure 18.8 shows an example of the procedure for setting waveform output by compare match

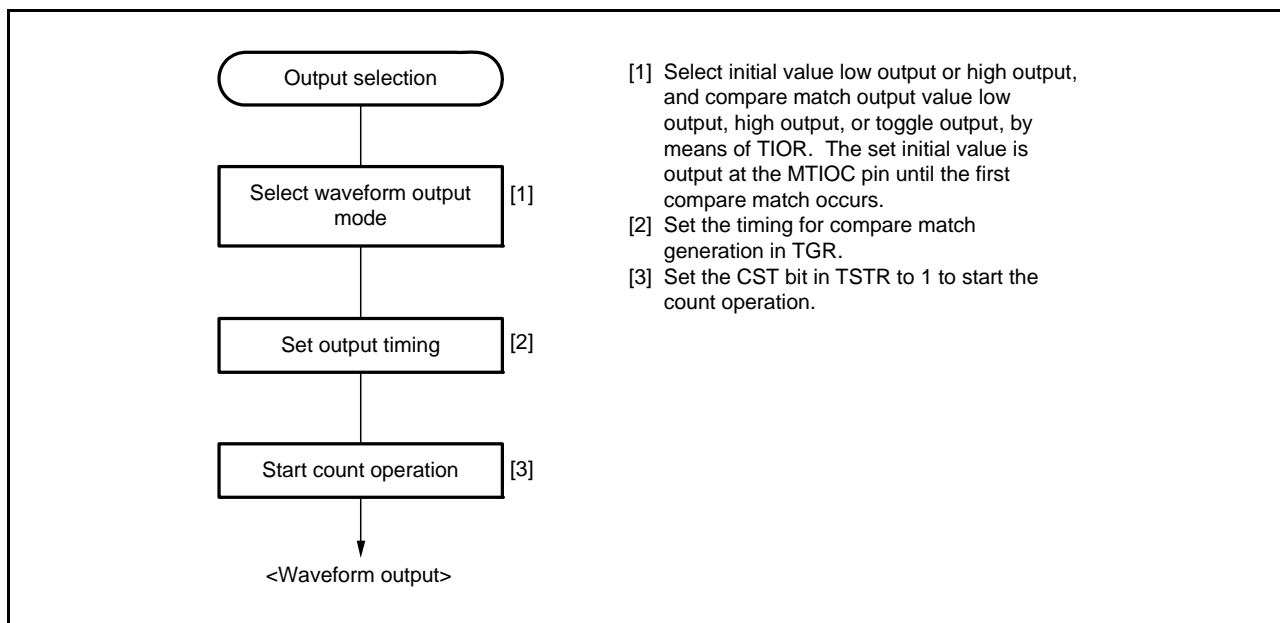


Figure 18.8 Example of Procedure for Setting Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 18.9 shows an example of low output and high output.

In this example, TCNT has been designated as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the pin level is the same as the specified level, the pin level does not change.

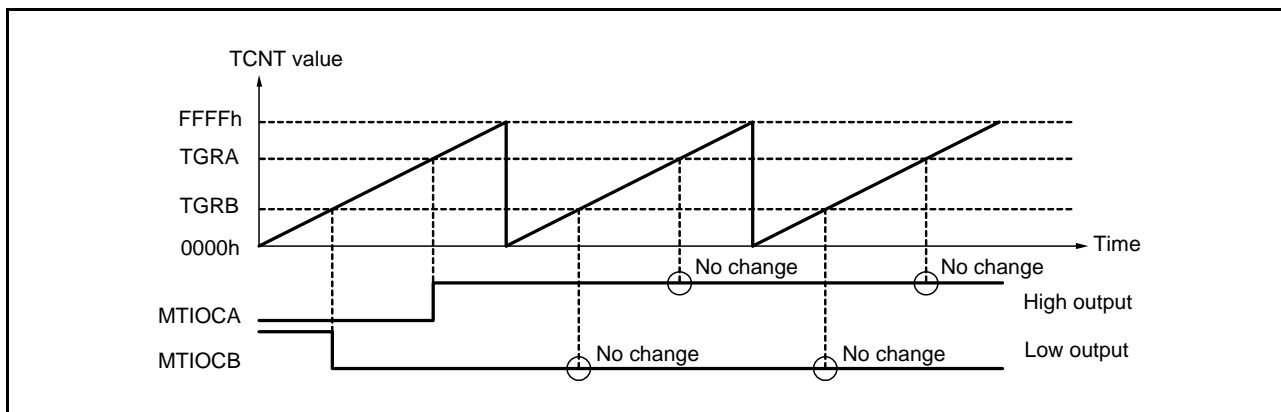


Figure 18.9 Example of Low Output and High Output Operation

Figure 18.10 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made so that the output is toggled by both compare match A and compare match B.

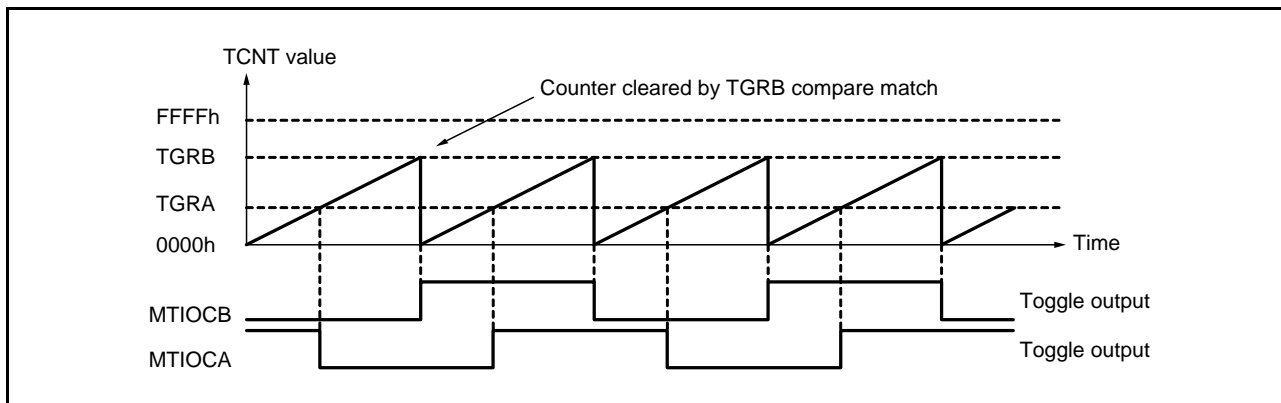


Figure 18.10 Example of Toggle Output Operation

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the MTIOC pin input edge.

The rising edge, falling edge, or both edges can be selected as the detection edge. For channels 0, 1, 6, and 7, 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 channels 0, 1, 6, and 7, PCLK/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if PCLK/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 18.11 shows an example of the input capture operation setting procedure.

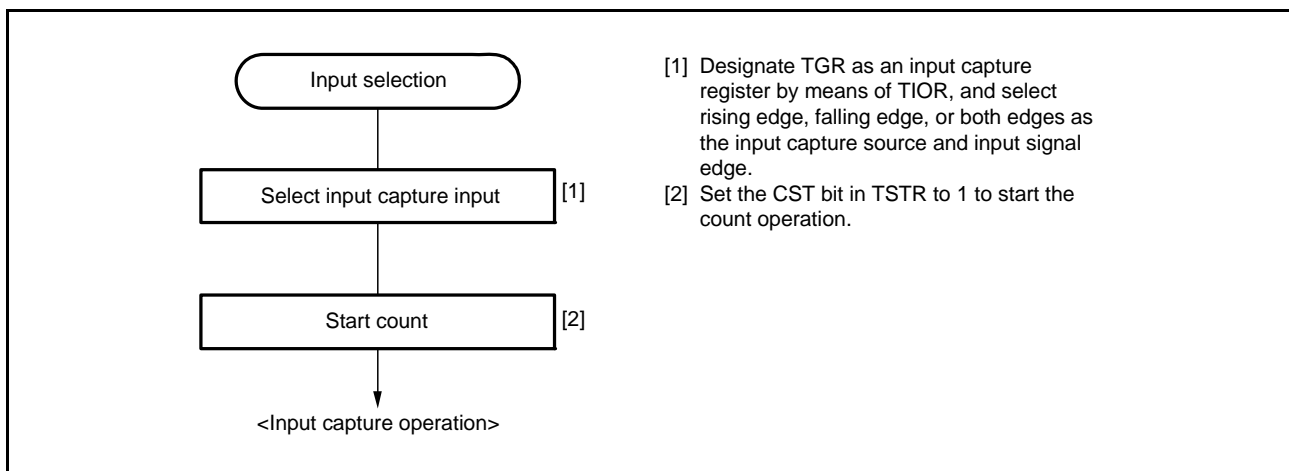


Figure 18.11 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 18.12 shows an example of input capture operation.

In this example, both rising and falling edges have been selected as the MTIOCA pin input capture input edge, the falling edge has been selected as the MTIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

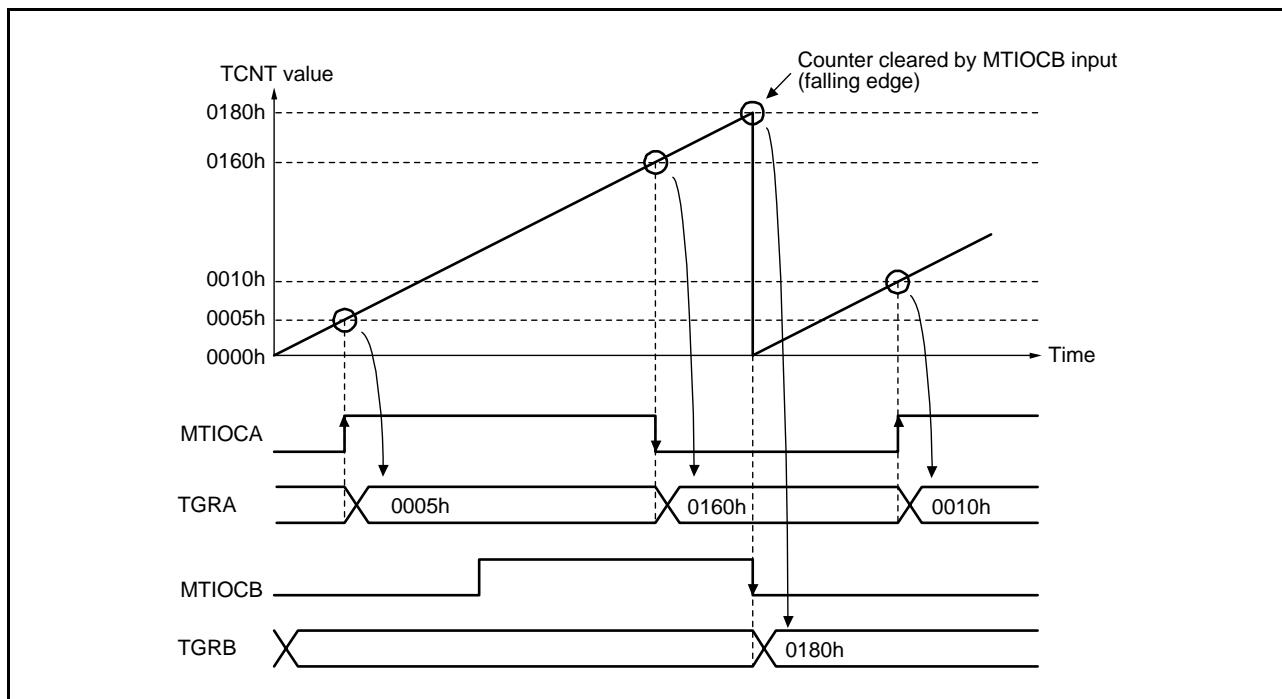


Figure 18.12 Example of Input Capture Operation

18.3.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous presetting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.

Synchronous operation increases the number of TGR registers assigned to a single time base.

Channels 0 to 4 and 6 to 10 can all be designated for synchronous operation. Channels 5 and 11 cannot be used for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 18.13 shows an example of the synchronous operation setting procedure.

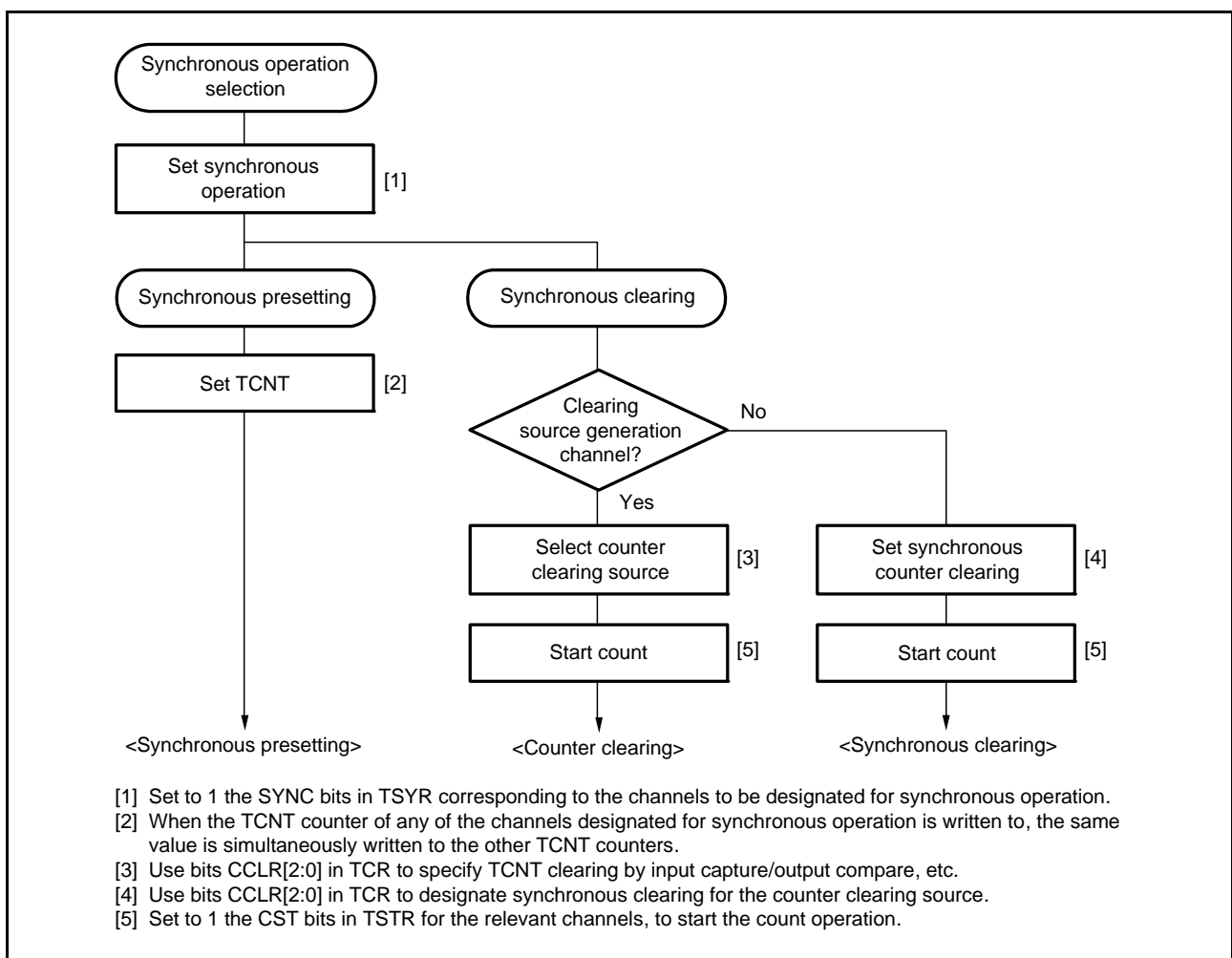


Figure 18.13 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 18.14 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, MTU0.TGRB compare match has been set as the counter clearing source in channel 0, and synchronous clearing has been set for the counter clearing source in channels 1 and 2.

Three-phase PWM waveforms are output from pins MTIOC0A, MTIOC1A, and MTIOC2A. At this time, synchronous presetting and synchronous clearing by MTU0.TGRB compare match are performed for the TCNT counters in channels 0 to 2, and the data set in MTU0.TGRB is used as the PWM cycle.

For details of PWM modes, see section 18.3.5, PWM Modes .

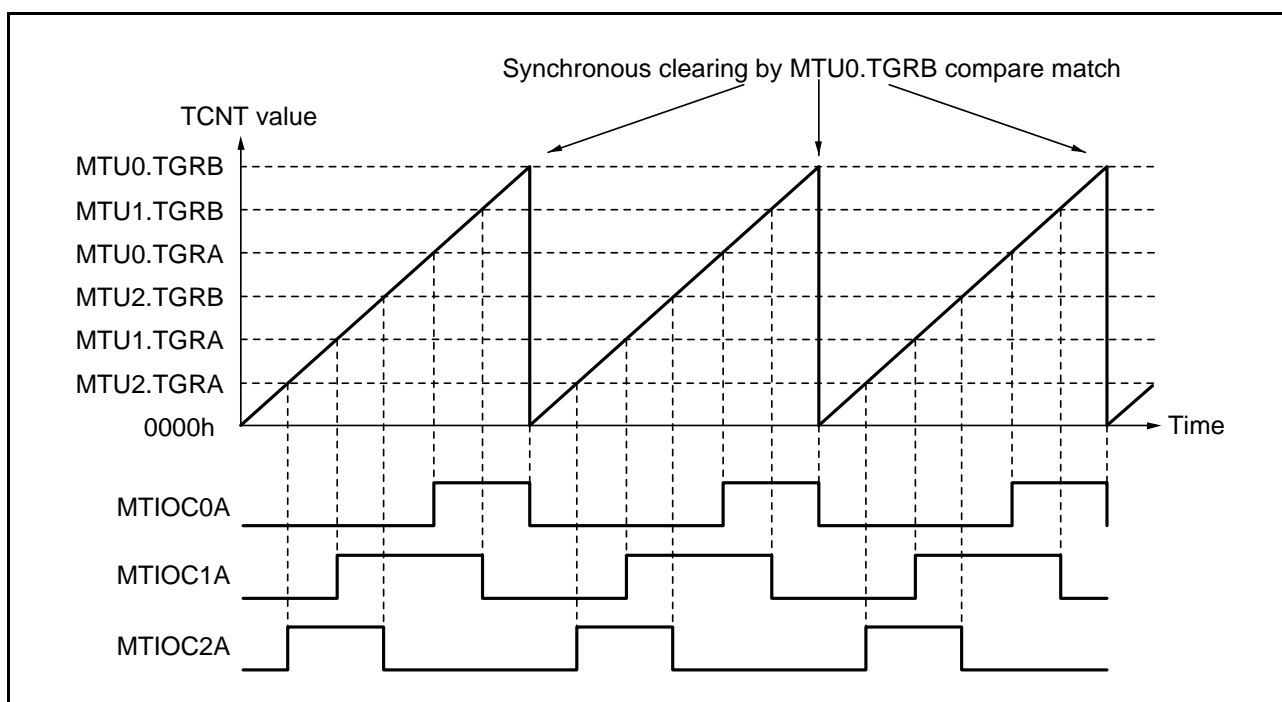


Figure 18.14 Example of Synchronous Operation

18.3.3 Buffer Operation

Buffer operation, provided for channels 0, 3, 4, 6, 9, and 10, enables TGRC and TGRD to be used as buffer registers. In channels 0 and 6, 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: MTUn.TGRE cannot be designated as an input capture register and can only operate as a compare match register. (n = 0 or 6)

Table 18.45 shows the register combinations used in buffer operation.

Table 18.45 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
MTU0, MTU6	TGRA	TGRC
	TGRB	TGRD
	TGRE	TGRF
MTU3, MTU9	TGRA	TGRC
	TGRB	TGRD
MTU4, MTU10	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 18.15.

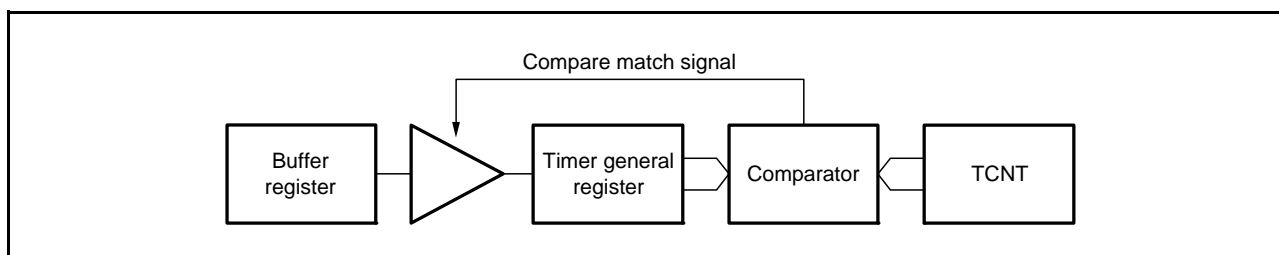


Figure 18.15 Compare Match Buffer Operation

- 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 18.16.

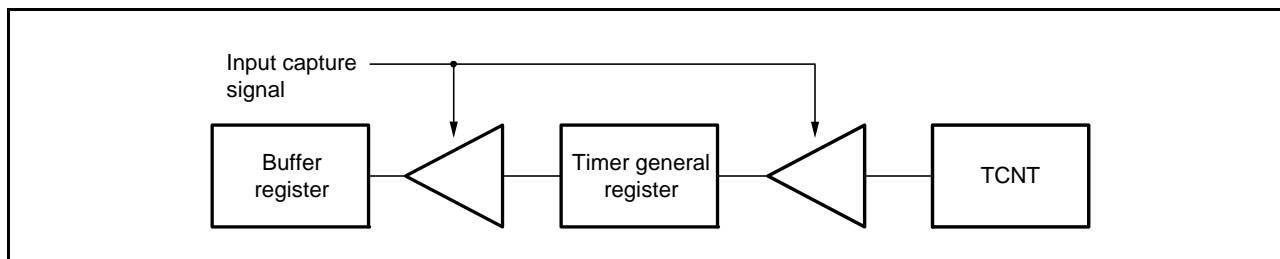


Figure 18.16 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 18.17 shows an example of the buffer operation setting procedure.

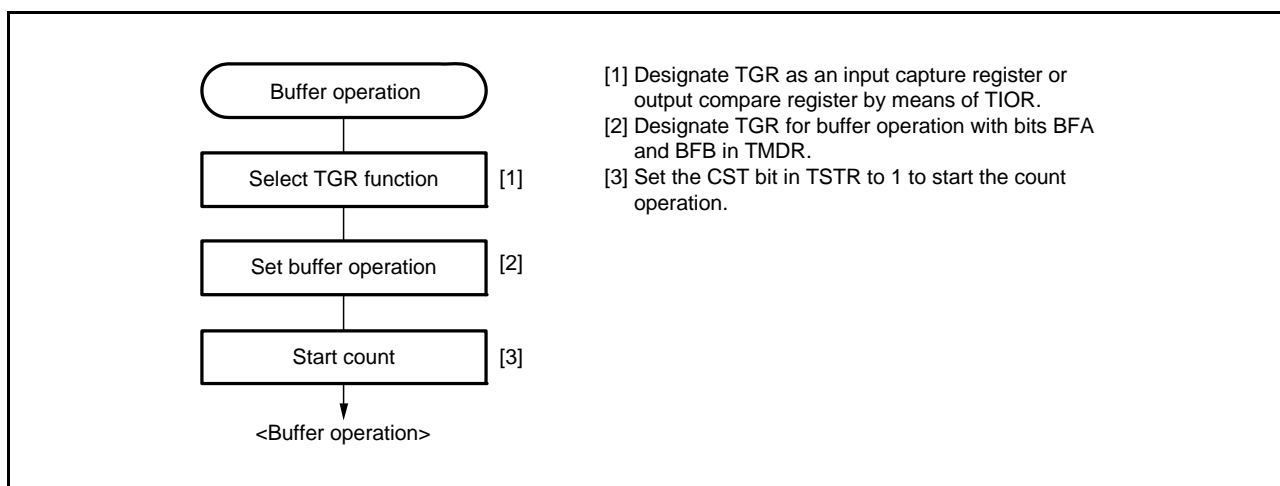


Figure 18.17 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an Output Compare Register

Figure 18.18 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. In this example, the TTSA bit in TBTM is cleared to 0.

As buffer operation has been set, when compare match A occurs, the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

For details of PWM modes, see section 18.3.5, PWM Modes .

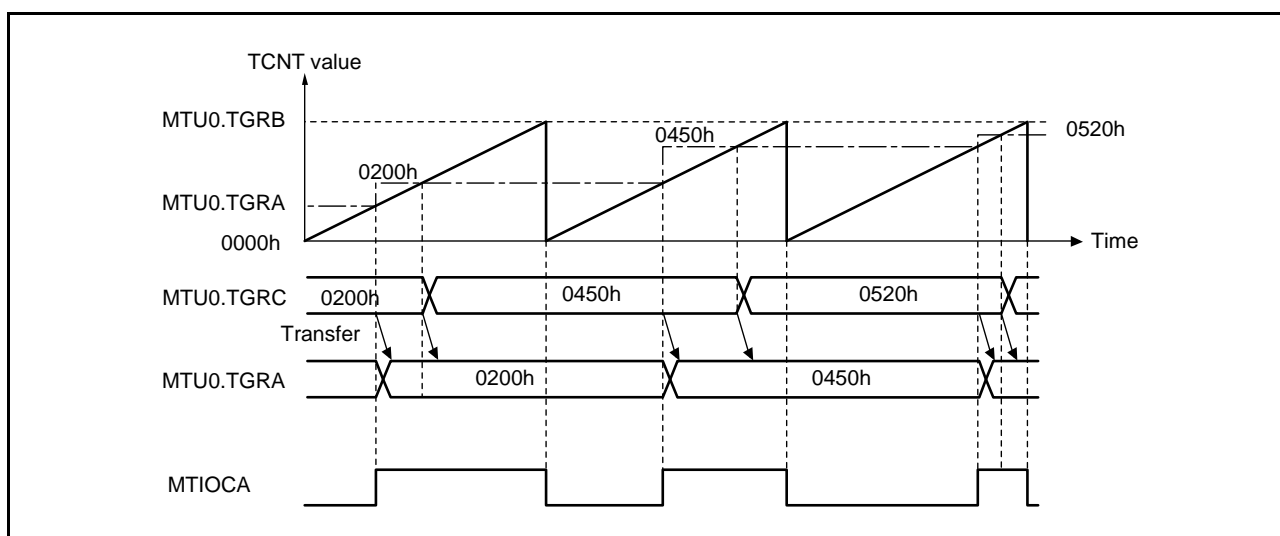


Figure 18.18 Example of Buffer Operation (1)

(b) When TGR is an Input Capture Register

Figure 18.19 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 MTIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

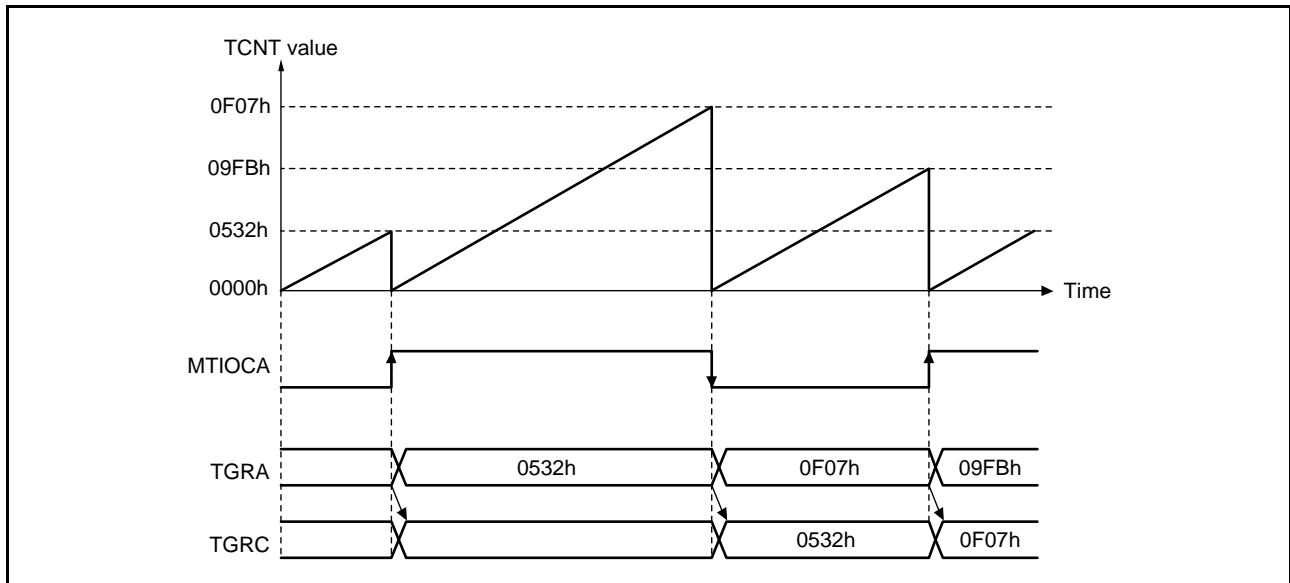


Figure 18.19 Example of Buffer Operation (2)

(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 channels 0 and 6 or in PWM mode 1 for channels 3, 4, 9, and 10 by setting the buffer operation transfer mode registers (MTU_n.TBTM, MTU_m.TBTM, and MTU_i.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. (n = 0 or 6, m = 3 or 9, i = 4 or 10)

- When TCNT overflows (FFFFh to 0000h)
- When 0000h is written to TCNT during counting
- When TCNT is cleared to 0000h under the condition specified in the CCLR[2:0] bits in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 18.20 shows an operation example in which PWM mode 1 is designated for channel 0 and buffer operation is designated for MTU0.TGRA and MTU0.TGRC. The settings used in this example are MTU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. The TTSA bit in MTU0.TBTM is set to 1.

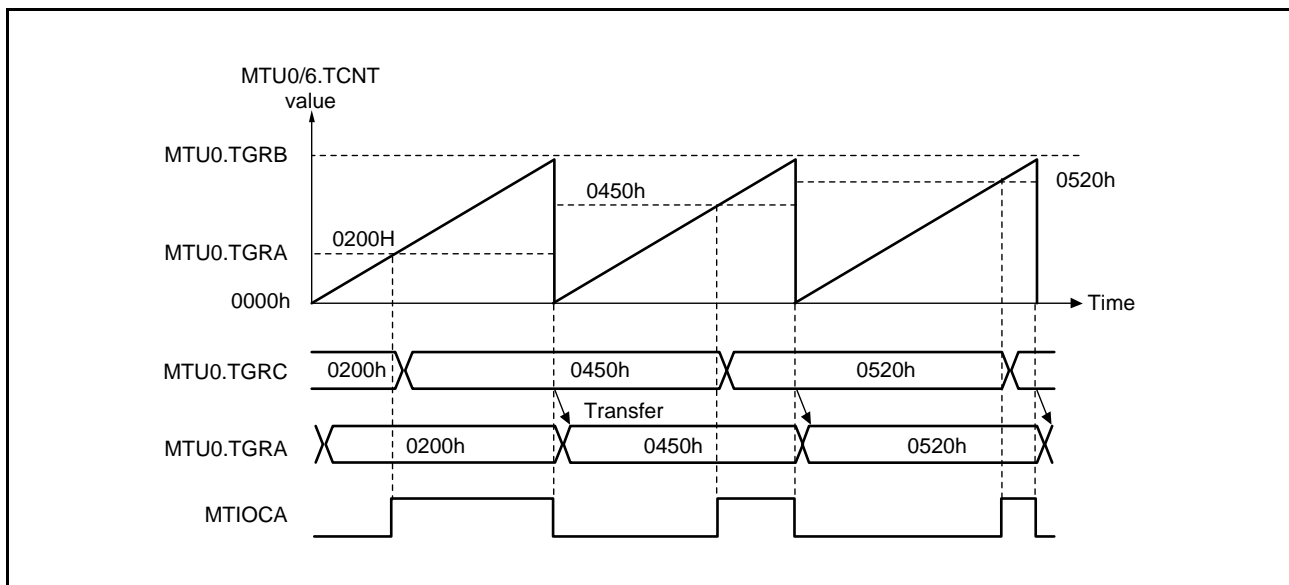


Figure 18.20 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC-to-MTU0.TGRA Transfer Timing

18.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters in different channels are used together as a 32-bit counter.

This function works when overflow/underflow of MTU1.TCNT or MTU7.TCNT is selected as the counter clock for channel 1 or 7 through the TPSC[2:0] bits in TCR.

Underflow occurs only when the lower 16 bits of TCNT is in phase counting mode.

Table 18.46 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 7, the counter clock setting is invalid and the counters operate independently in phase counting mode.

Table 18.46 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2 (Channels 7 and 8)	MTU1.TCNT (MTU7.TCNT)	MTU2.TCNT (MTU8.TCNT)

For simultaneous input capture of MTUn.TCNT and MTUm.TCNT during cascaded operation, additional input capture input pins can be specified by the input capture control register (MTICCR). Edge detection for the input capture condition is of edges in the signal produced by taking the logical OR of the level input on the original input pin and the level being input on the added input pin. Therefore, when either of these is driven high, an edge in the other will not be detected. For details, see the description under (4)Cascaded Operation Example (c). For input capture in cascade connection, refer to section 17.6.20, Simultaneous Input Capture in MTUn.TCNT and MTUm.TCNT in Cascade Connection. (n = 1 or 7, m = 2 or 8)

Table 18.47 shows the MTICCR setting and input capture input pins.

Table 18.47 MTICCR Setting and Input Capture Input Pins

Target Input Capture	MTICCR Setting	Input Capture Input Pin
Input capture from MTUn.TCNT to MTUn.TGRA	I2AE bit = 0 (initial value)	MTIOCnA
	I2AE bit = 1	MTIOCnA, MTIOCmA
Input capture from MTUn.TCNT to MTUn.TGRB	I2BE bit = 0 (initial value)	MTIOCnB
	I2BE bit = 1	MTIOCnB, MTIOCmB
Input capture from MTUm.TCNT to MTUm.TGRA	I1AE bit = 0 (initial value)	MTIOCmA
	I1AE bit = 1	MTIOCmA, MTIOCnA
Input capture from MTUm.TCNT to MTUm.TGRB	I1BE bit = 0 (initial value)	MTIOCmB
	I1BE bit = 1	MTIOCmB, MTIOCnB

[Legend] (n = 1 or 7, m = 2 or 8)

(1) Example of Cascaded Operation Setting Procedure

Figure 18.21 shows an example of the cascaded operation setting procedure.

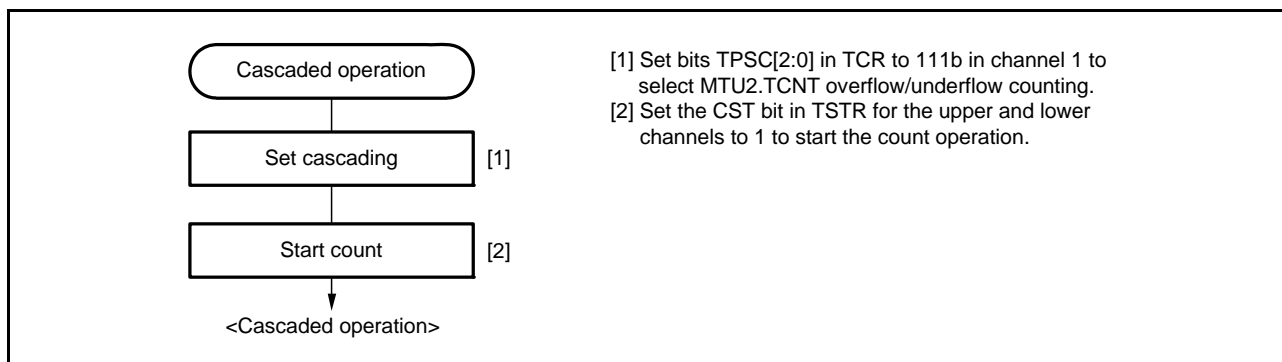


Figure 18.21 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 18.22 illustrates the operation when MTU2.TCNT overflow/underflow counting has been set for MTU1.TCNT and phase counting mode has been designated for channel 2.

MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

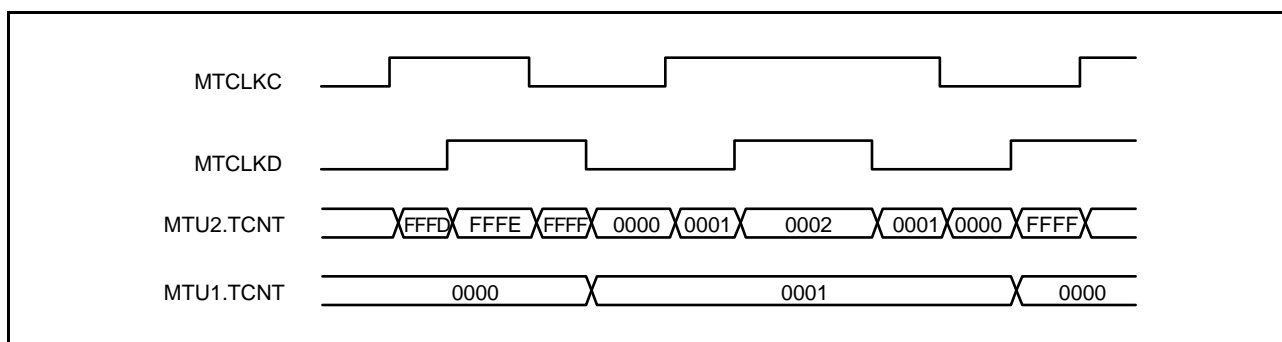


Figure 18.22 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 18.23 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE bit in MTICCR has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA3 to IOA0 bits in MTU1.TIOR have selected the MTIOC1A rising edge for the input capture timing while the IOA3 to IOA0 bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, the rising edge of both MTIOC1A and MTIOC2A is used for the MTU1.TGRA input capture condition. For the MTU2.TGRA input capture condition, the MTIOC2A rising edge is used.

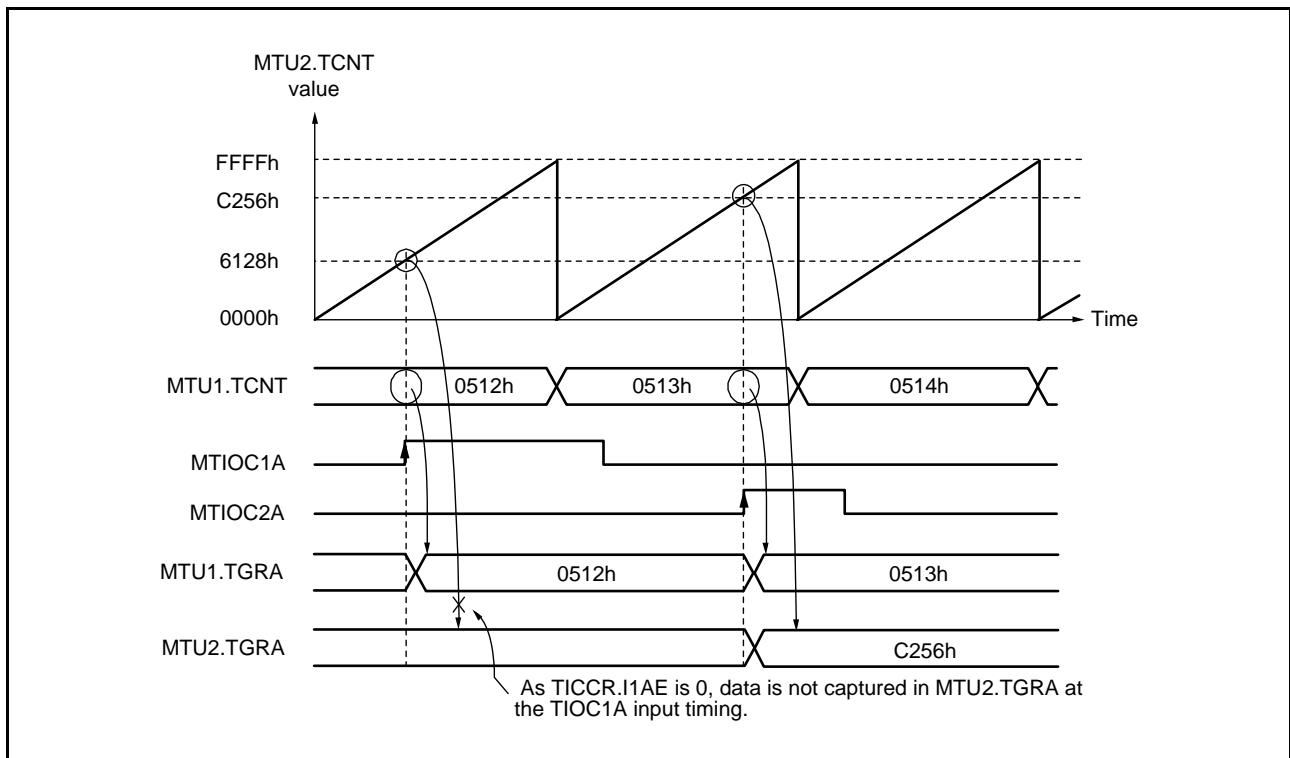


Figure 18.23 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 18.24 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE and I1AE bits in MTICCR have been set to 1 to include the MTIOC2A and MTIOC1A pins in the MTU1.TGRA and MTU2.TGRA input capture conditions, respectively. In this example, the IOA3 to IOA0 bits in both MTU1.TIOR and MTU2.TIOR have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.

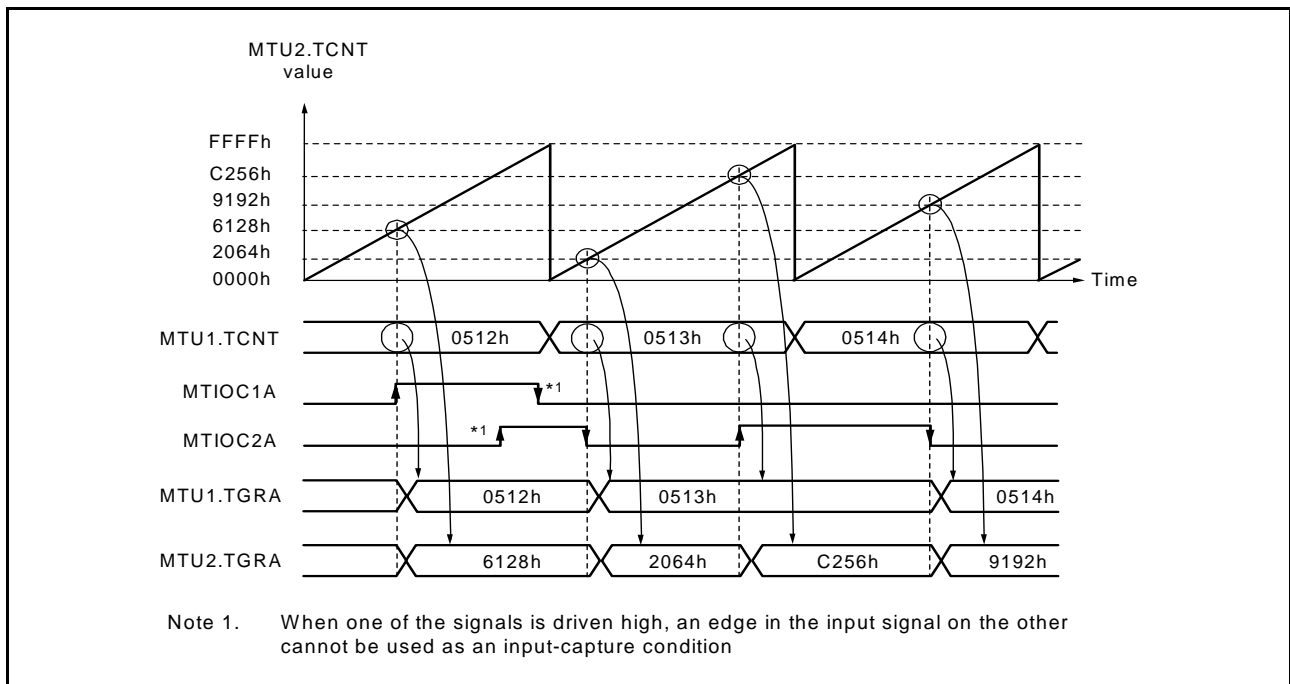


Figure 18.24 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 18.25 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE bit in MTICCR has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA3 to IOA0 bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA3 to IOA0 bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, as MTU1.TIOR has selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing, the MTIOC2A edge is not used for MTU1.TGRA input capture condition although the I2AE bit in MTICCR has been set to 1.

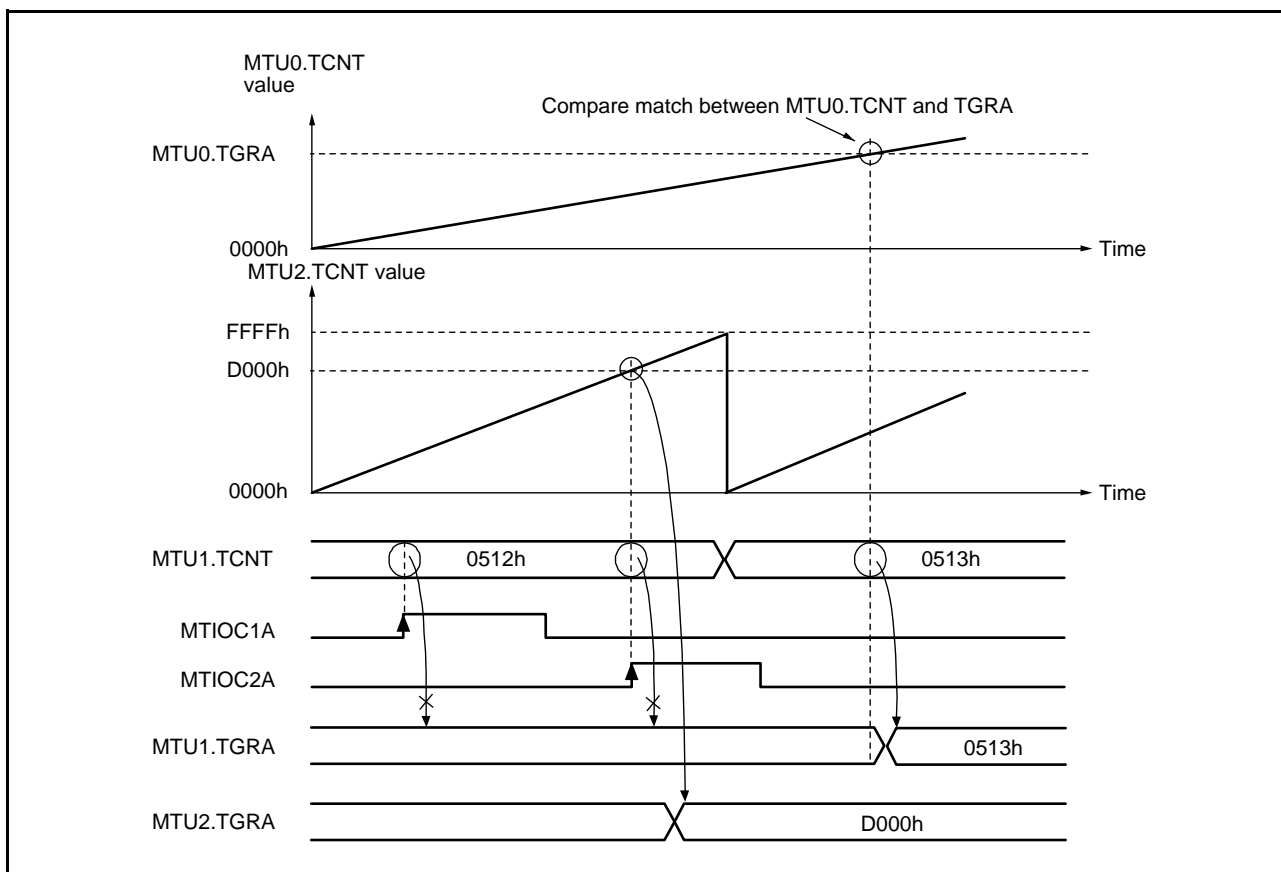


Figure 18.25 Cascaded Operation Example (d)

18.3.5 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. Every channel can be set to PWM mode independently. Synchronous operation is also possible.

There are two PWM modes as described below.

(a) PWM Mode 1

PWM waveforms are output from the MTIOCA and MTIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The levels specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR are output from the MTIOCA and MTIOCC pins at compare matches A and C, and the level specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR are output at compare matches B and D. The initial output value is set in TGRA or TGRC. If the values set in paired TGRs are identical, the output value does not change even when a compare match occurs.

In PWM mode 1, up to eight phases of PWM waveforms can be output.

(b) PWM Mode 2

PWM waveforms output is generated using one TGR as the cycle register and the others as duty registers. The level specified in TIOR is output at compare matches. Upon counter clearing by a synchronized register compare match, the initial value set in TIOR is output from each pin. If the values set in the cycle and duty registers are identical, the output value does not change even when a compare match occurs.

In PWM mode 2, up to eight phases of PWM waveforms can be output when using synchronous operation in combination.

The correspondence between PWM output pins and registers is shown in Table 18.48.

Table 18.48 PWM Output Registers and Output Pins

Channel	Register	Output Pins	
		PWM Mode 1	PWM Mode 2
MTU0, MTU6	MTUn.TGRA (n = 0 or 6)	MTIOC0A MTIOC6A	MTIOC0A MTIOC6A
	MTUn.TGRB (n = 0 or 6)		MTIOC0B MTIOC6B
	MTUn.TGRC (n = 0 or 6)	MTIOC0C MTIOC6C	MTIOC0C MTIOC6C
	MTUn.TGRD (n = 0 or 6)		MTIOC0D MTIOC6D
MTU1, MTU7	MTUn.TGRA (n = 1 or 7)	MTIOC1A MTIOC7A	MTIOC1A MTIOC7A
	MTUn.TGRB (n = 1 or 7)		MTIOC1B MTIOC6B
MTU2, MTU8	MTUn.TGRA (n = 2 or 8)	MTIOC2A MTIOC8A	MTIOC2A MTIOC8A
	MTUn.TGRB (n = 2 or 8)		MTIOC2B MTIOC8B
MTU3, MTU9	MTUn.TGRA (n = 3 or 9)	MTIOC3A	Setting prohibited
	MTUn.TGRB (n = 3 or 9)	MTIOC9A	
	MTUn.TGRC (n = 3 or 9)	MTIOC3C	
	MTUn.TGRD (n = 3 or 9)	MTIOC9C	
MTU4, MTU10	MTUn.TGRA (n = 4 or 10)	MTIOC4A	
	MTUn.TGRB (n = 4 or 10)	MTIOC10A	
	MTUn.TGRC (n = 4 or 10)	MTIOC4C	
	MTUn.TGRD (n = 4 or 10)	MTIOC10C	

Note: In PWM mode 2, PWM waveforms output is not possible for the TGR register in which the PWM cycle is set.

(1) Example of PWM Mode Setting Procedure

Figure 18.26 shows an example of the PWM mode setting procedure.

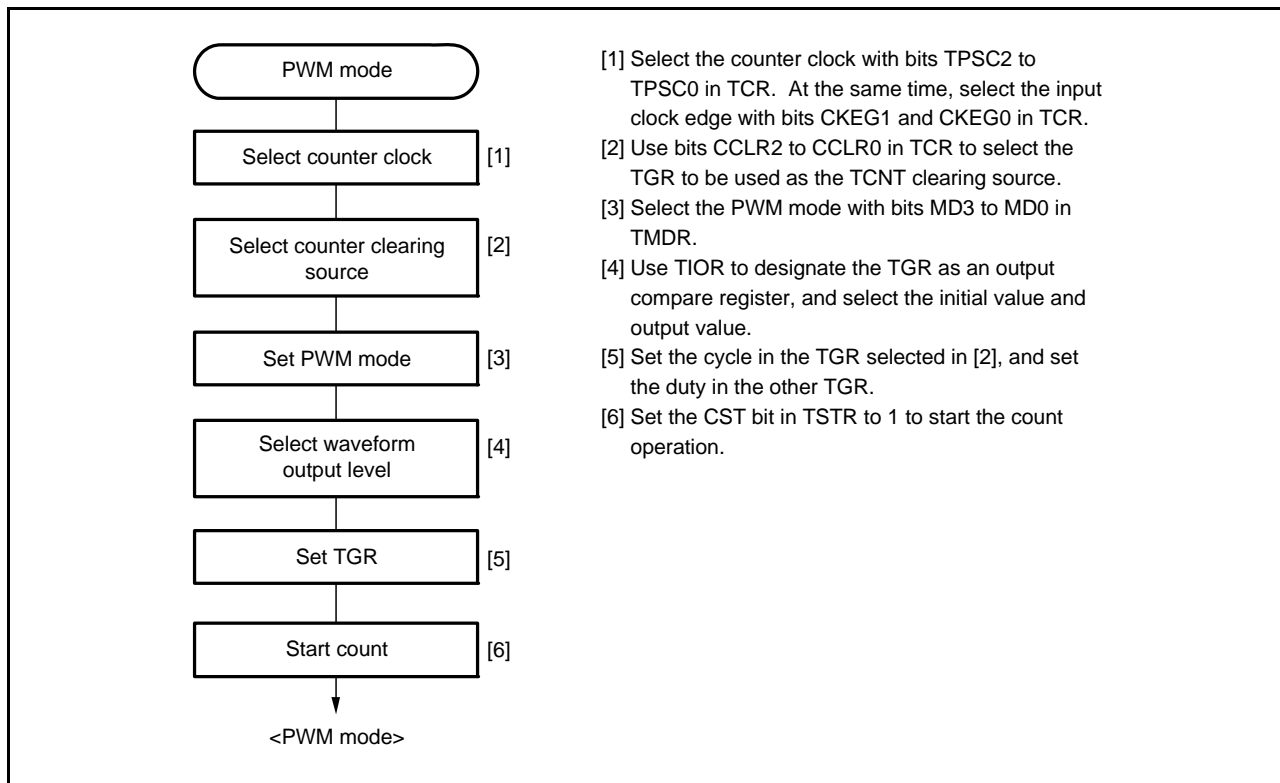


Figure 18.26 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 18.27 shows an example of operation in PWM mode 1.

In this example, TGRA compare match is set as the TCNT clearing source, "low" is set as the initial output value and output value for TGRA, and "high" 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 specifies the duty.

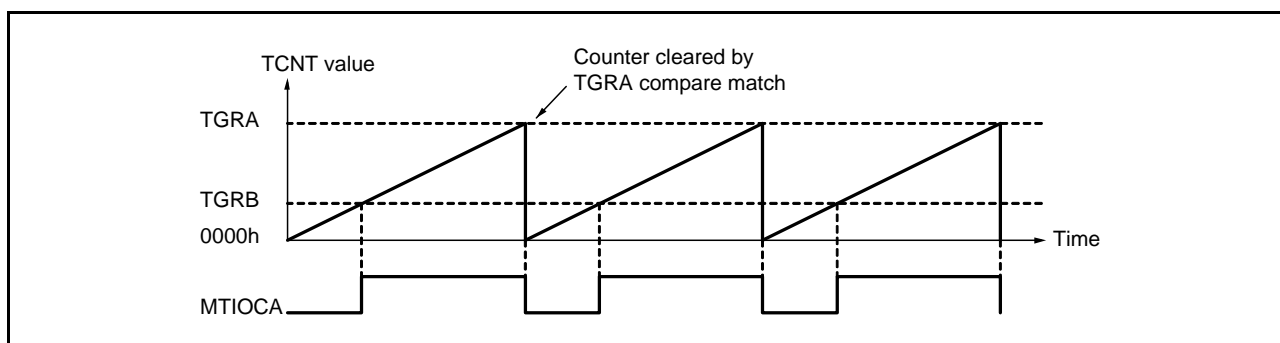


Figure 18.27 Example of PWM Mode 1 Operation

Figure 18.28 shows an example of operation in PWM mode 2.

In this example, synchronous operation is designated for channels 0 and 1, MTU1.TGRB compare match is set as the TCNT clearing source, and "low" is set as the initial output value and "high" as the output value for the other TGR registers (MTU0.TGRA to MTU0.TGRD and MTU1.TGRA), outputting 5-phase PWM waveforms.

In this case, the value set in TGR1B is used as the cycle, and the values set in the other TGRs (MTU0.TGRA to MTU0.TGRD, MTU1.TGRA) are used as the respective duty cycles.

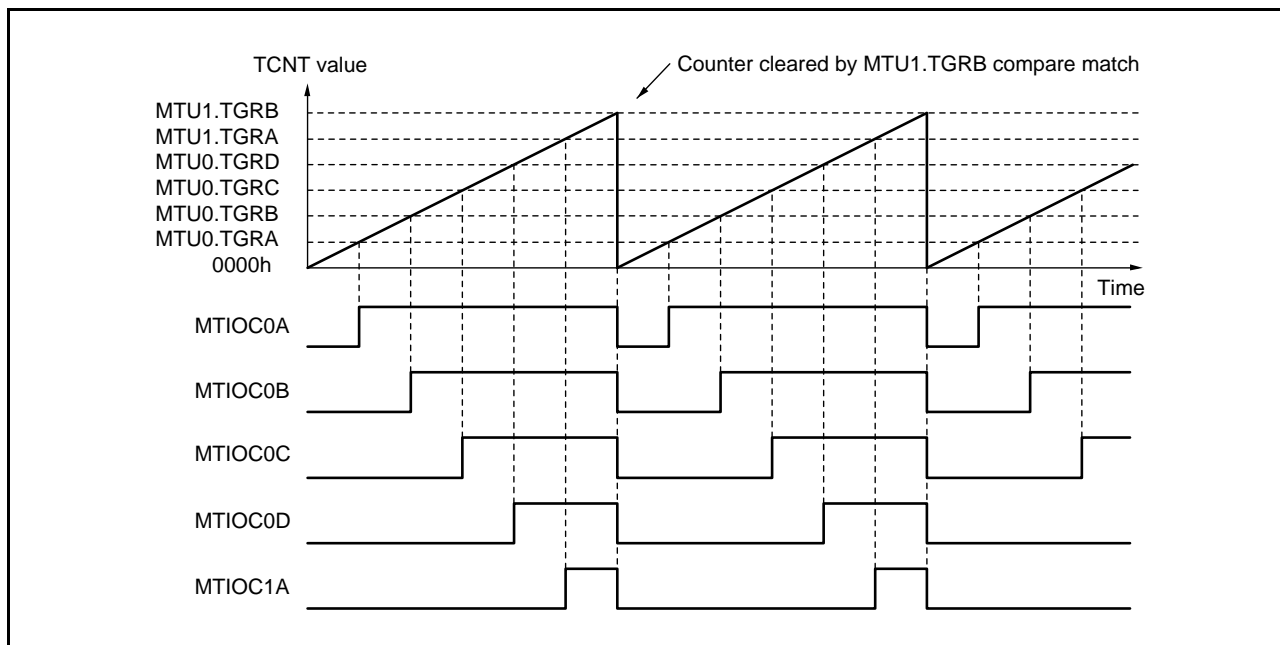


Figure 18.28 Example of PWM Mode 2 Operation

Figure 18.29 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

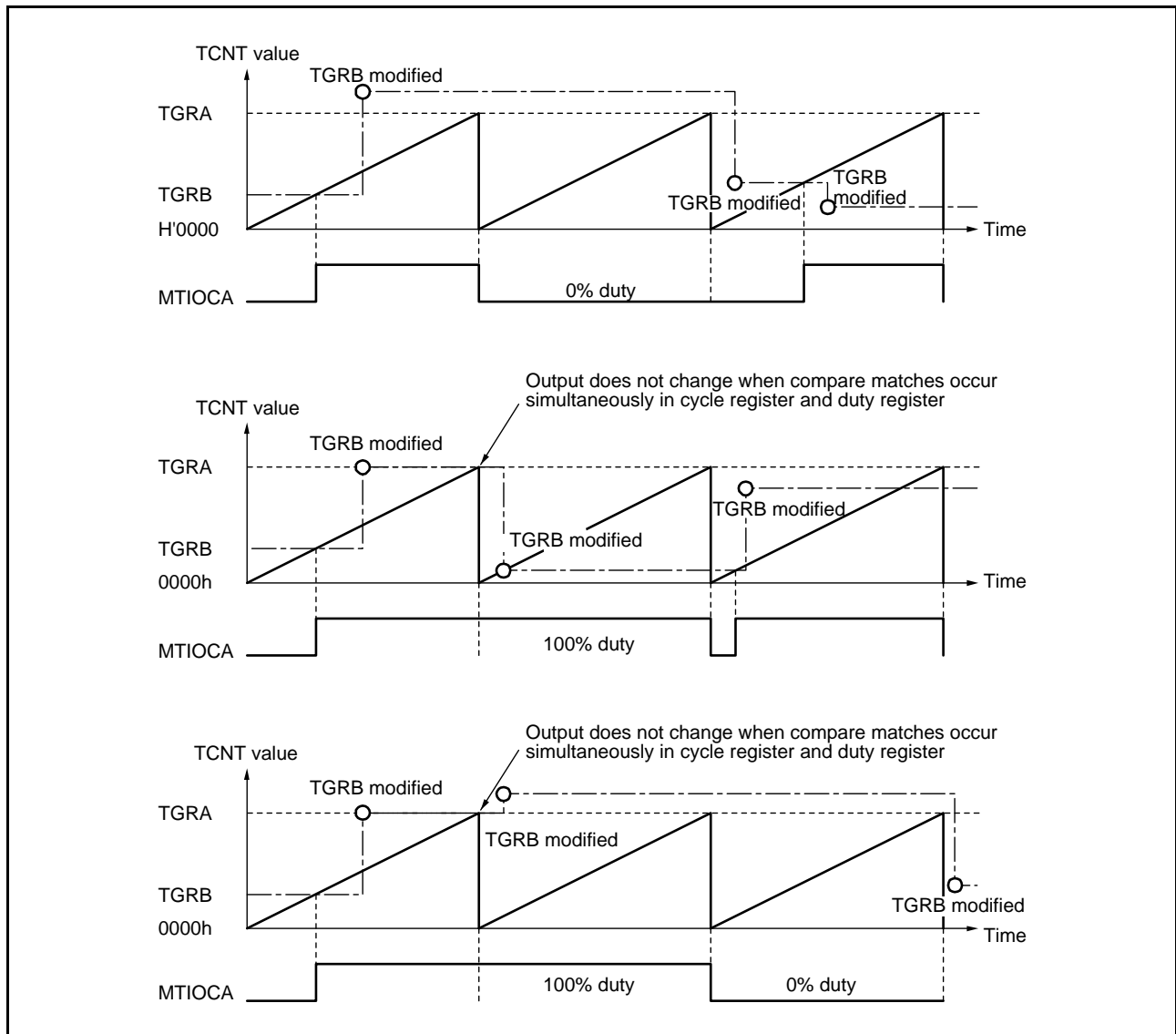


Figure 18.29 Examples of Duty Cycle Modification in PWM Mode

18.3.6 Phase Counting Mode

In phase counting mode, the phase difference between two external input clocks is detected and TCNT is incremented or decremented accordingly. This mode can be set for channels 1 and 2 (or channels 7 and 8).

When phase counting mode is specified, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC[2:0] and bits CKEG[1:0] in TCR. However, the functions of bits CCLR[1:0] in TCR and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

If an overflow occurs while TCNT is counting up, a TCIV interrupt is generated while the TCIEV bit in the corresponding TIER is 1. If an underflow occurs while TCNT is counting down, a TCIU interrupt is generated while the TCIEU bit in the corresponding TIER is 1.

The TCFD bit in TSR is the count direction flag. Read the TCFD flag to check whether TCNT is counting up or down.

Table 18.49 shows the correspondence between external clock pins and channels.

Table 18.49 Clock Input Pins in Phase Counting Mode

Channel	External Clock Input Pins	
	A-Phase	B-Phase
When channel 1 (or 7) is set to phase counting mode	MTCLKA (MTCLKE)	MTCLKB (MTCLKF)
When channel 2 (or 8) is set to phase counting mode	MTCLKC (MTCLKG)	MTCLKD (MTCLKH)

(1) Example of Phase Counting Mode Setting Procedure

Figure 18.30 shows an example of the phase counting mode setting procedure.

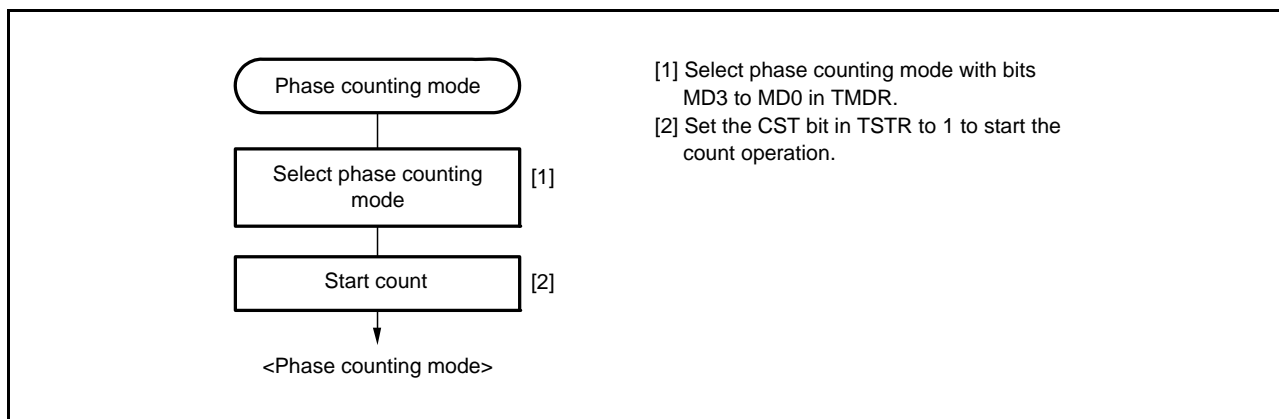


Figure 18.30 Example of Phase Counting Mode Setting Procedure

(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 18.31 shows an example of operation in phase counting mode 1, and Table 18.50 summarizes the TCNT up-/down-count conditions.

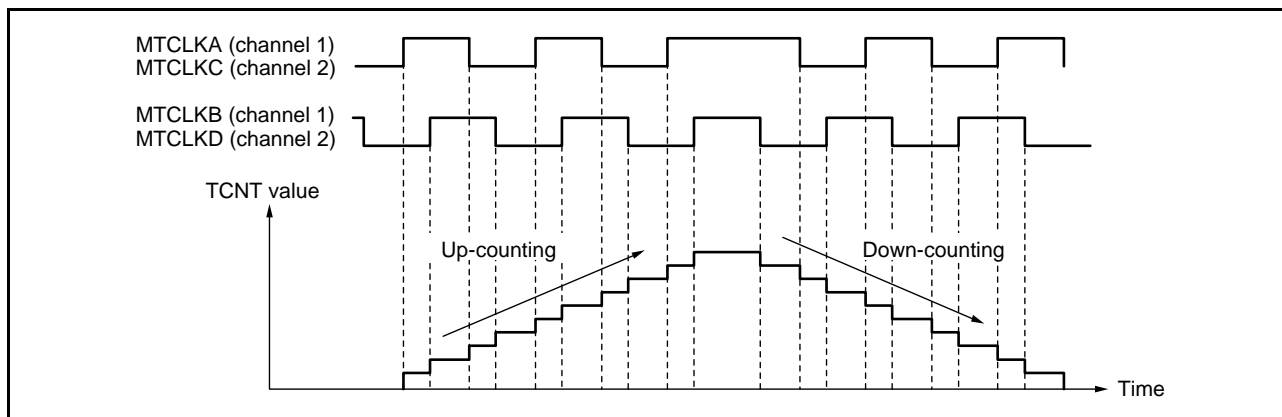


Figure 18.31 Example of Operation in Phase Counting Mode 1

Table 18.50 Up-/Down-Count Conditions in Phase Counting Mode 1

MTCLKA (Channel 1) MTCLKC (Channel 2)	MTCLKB (Channel 1) MTCLKD (Channel 2)	Operation
High level		Up-counting
Low level		
	Low level	
	High level	
High level		Down-counting
Low level		
	High level	
	Low level	

[Legend]

- : Rising edge
- : Falling edge

(b) Phase Counting Mode 2

Figure 18.32 shows an example of operation in phase counting mode 2, and Table 18.51 summarizes the TCNT up-/down-count conditions.

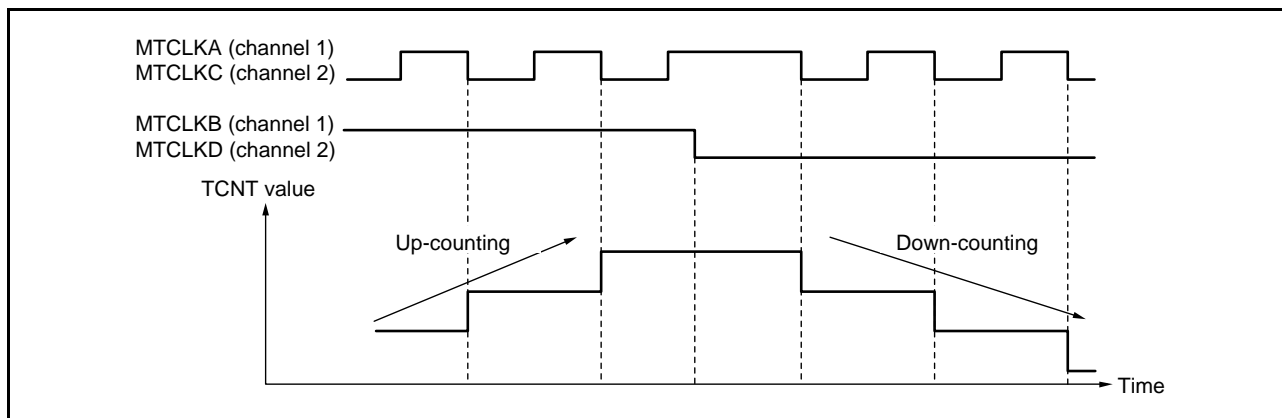


Figure 18.32 Example of Operation in Phase Counting Mode 2

Table 18.51 Up-/Down-Count Conditions in Phase Counting Mode 2

MTCLKA (Channel 1) MTCLKC (Channel 2)	MTCLKB (Channel 1) MTCLKD (Channel 2)	Operation
High level		Not counted (Don't care)
Low level		
	Low level	
	High level	Up-counting
High level		Not counted (Don't care)
Low level		
	High level	
	Low level	Down-counting

[Legend]

- : Rising edge
- : Falling edge

(c) Phase Counting Mode 3

Figure 18.33 shows an example of operation in phase counting mode 3, and Table 18.52 summarizes the TCNT up-/down-count conditions.

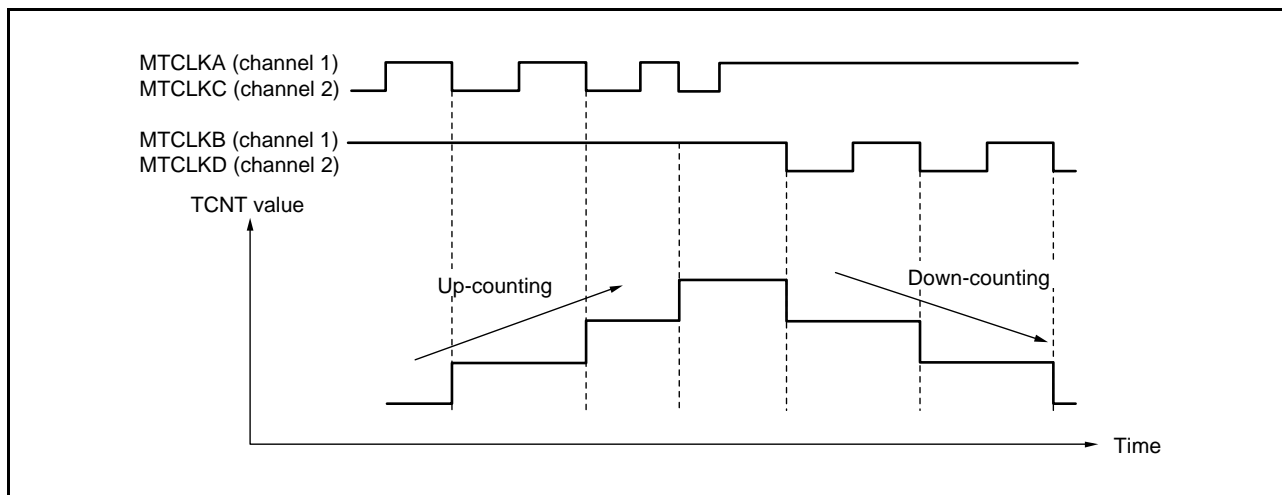


Figure 18.33 Example of Operation in Phase Counting Mode 3

Table 18.52 Up-/Down-Count Conditions in Phase Counting Mode 3

MTCLKA (Channel 1) MTCLKC (Channel 2)	MTCLKB (Channel 1) MTCLKD (Channel 2)	Operation
High level		Not counted (Don't care)
Low level		Not counted (Don't care)
	Low level	Up-counting
	High level	Up-counting
High level		Down-counting
Low level		Down-counting
	High level	Not counted (Don't care)
	Low level	Not counted (Don't care)

[Legend]

- : Rising edge
- : Falling edge

(d) Phase Counting Mode 4

Figure 18.34 shows an example of operation in phase counting mode 4, and Table 18.53 summarizes the TCNT up-/down-count conditions.

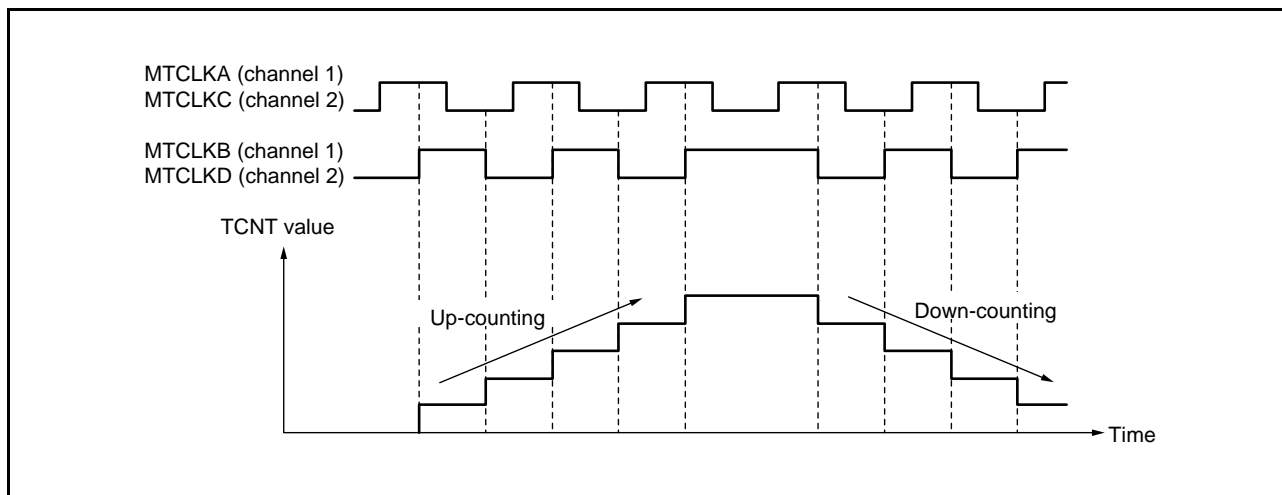


Figure 18.34 Example of Operation in Phase Counting Mode 4

Table 18.53 Up-/Down-Count Conditions in Phase Counting Mode 4

MTCLKA (Channel 1) MTCLKC (Channel 2)	MTCLKB (Channel 1) MTCLKD (Channel 2)	Operation
High level		Up-counting
Low level		Up-counting
	Low level	Not counted (Don't care)
	High level	Not counted (Don't care)
High level		Down-counting
Low level		Down-counting
	High level	Not counted (Don't care)
	Low level	Not counted (Don't care)

[Legend]

- : Rising edge
- : Falling edge

(3) Phase Counting Mode Application Example

Figure 18.35 shows an example in which channel 1 (or channel 7) is in phase counting mode, and channel 1 (or channel 7) is coupled with channel 0 (or channel 6) to input 2-phase encoder pulses of a servo motor in order to detect position or speed.

Channel 1 (or channel 7) is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to MTCLKA (or MTCLKE) and MTCLKB (or MTCLKF).

In channel 0 (or channel 6), MTUn.TGRC compare match is specified as the TCNT clearing source and MTUn.TGRA and TGRC are used for the compare match function and are set with the speed control cycle and position control cycle. MTUn.TGRB is used for input capture, with MTUn.TGRB and TGRD operating in buffer mode. The channel 1 (or channel 7) counter input clock is designated as the MTUn.TGRB input capture source, and the widths of 2-phase encoder 4-multiplication pulses are detected. (n = 0 or 6)

MTUm.TGRA and TGRB for channel 1 (or channel 7) are designated for the input capture function and MTUn.TGRA and TGRC compare matches in channel 0 (or channel 6) are selected as the input capture sources to store the up/down-counter values for the control cycles.

This procedure enables the accurate detection of position and speed. (n = 0 or 6, m = 1 or 7)

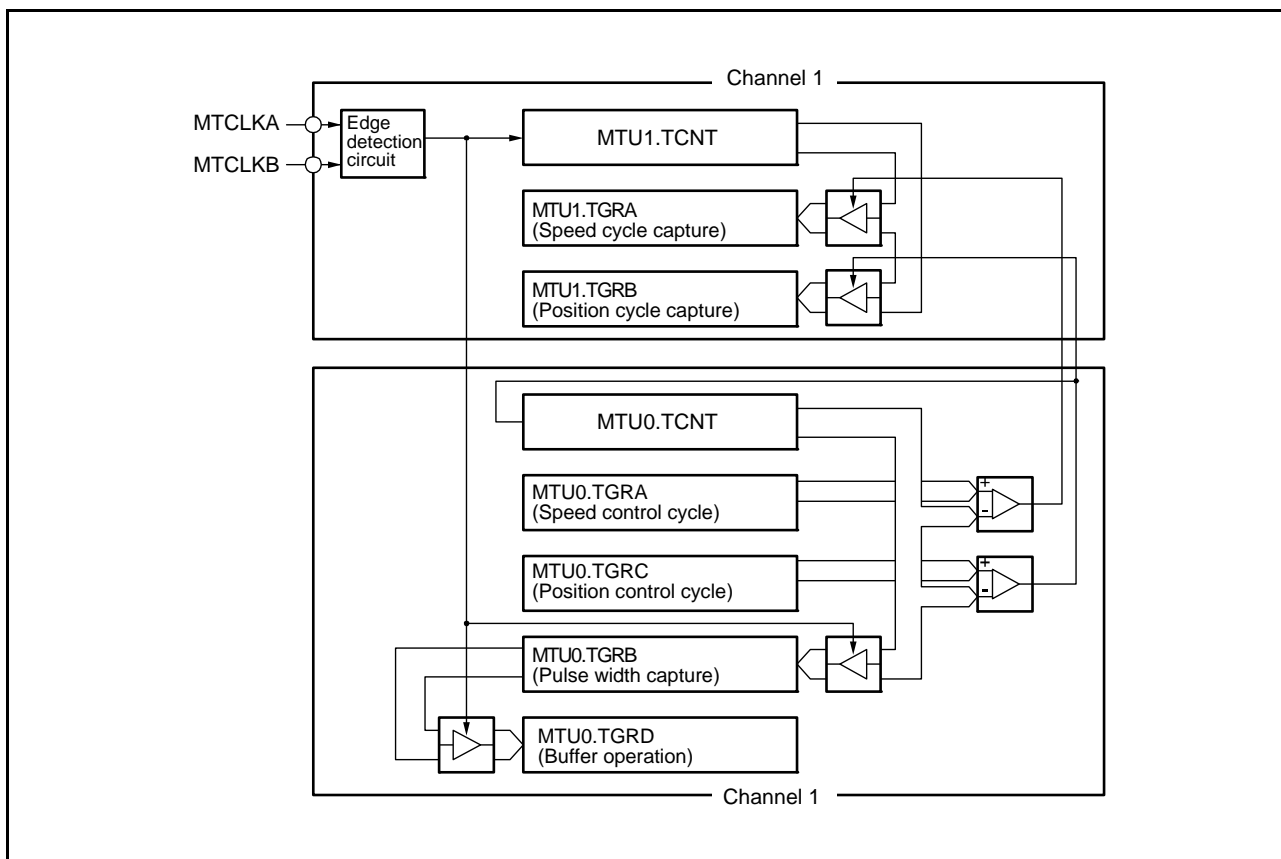


Figure 18.35 Phase Counting Mode Application Example (Unit 0)

18.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, three phases of positive and negative PWM waveforms that share a common wave transition point can be output by combining channels 3 and 4 (or channels 9 and 10).

When set for reset-synchronized PWM mode, the MTIOCnB, MTIOCnD, MTIOCmA, MTIOCmC, MTIOCmB, and MTIOCmD pins function as PWM output pins and timer counter n (MTUn.TCNT) functions as an up-counter. (n = 3 or 9, m = 4 or 10)

Table 18.54 shows the PWM output pins used. Table 18.55 shows the settings of the registers.

Table 18.54 Output Pins for Reset-Synchronized PWM Mode

Channel	Output Pin	Description
MTU3, MTU9	MTIOCnB	PWM output pin 1
	MTIOCnD	PWM output pin 1' (negative-phase waveform of PWM output 1)
MTU4, MTU10	MTIOCmA	PWM output pin 2
	MTIOCmC	PWM output pin 2' (negative-phase waveform of PWM output 2)
	MTIOCmB	PWM output pin 3
	MTIOCmD	PWM output pin 3' (negative-phase waveform of PWM output 3)

[Legend] n = 3 or 9, m = 4 or 10

Table 18.55 Register Settings for Reset-Synchronized PWM Mode

Register	Setting
MTUn.TCNT	Initial setting (0000h)
MTUm.TCNT	Initial setting (0000h)
MTUn.TGRA	Set the count cycle for MTUn.TCNT
MTUn.TGRB	Set the transition point of the PWM waveform to be output from the MTIOCnB and MTIOCnD pins
MTUm.TGRA	Set the transition point of the PWM waveform to be output from the MTIOCmA, and MTIOCmC pins
MTUm.TGRB	Set the transition point of the PWM waveform to be output from the MTIOCmB and MTIOCmD pins

[Legend] n = 3 or 9, m = 4 or 10

(1) Example of Procedure for Setting Reset-Synchronized PWM Mode

Figure 18.36 shows an example of procedure for setting the reset-synchronized PWM mode.

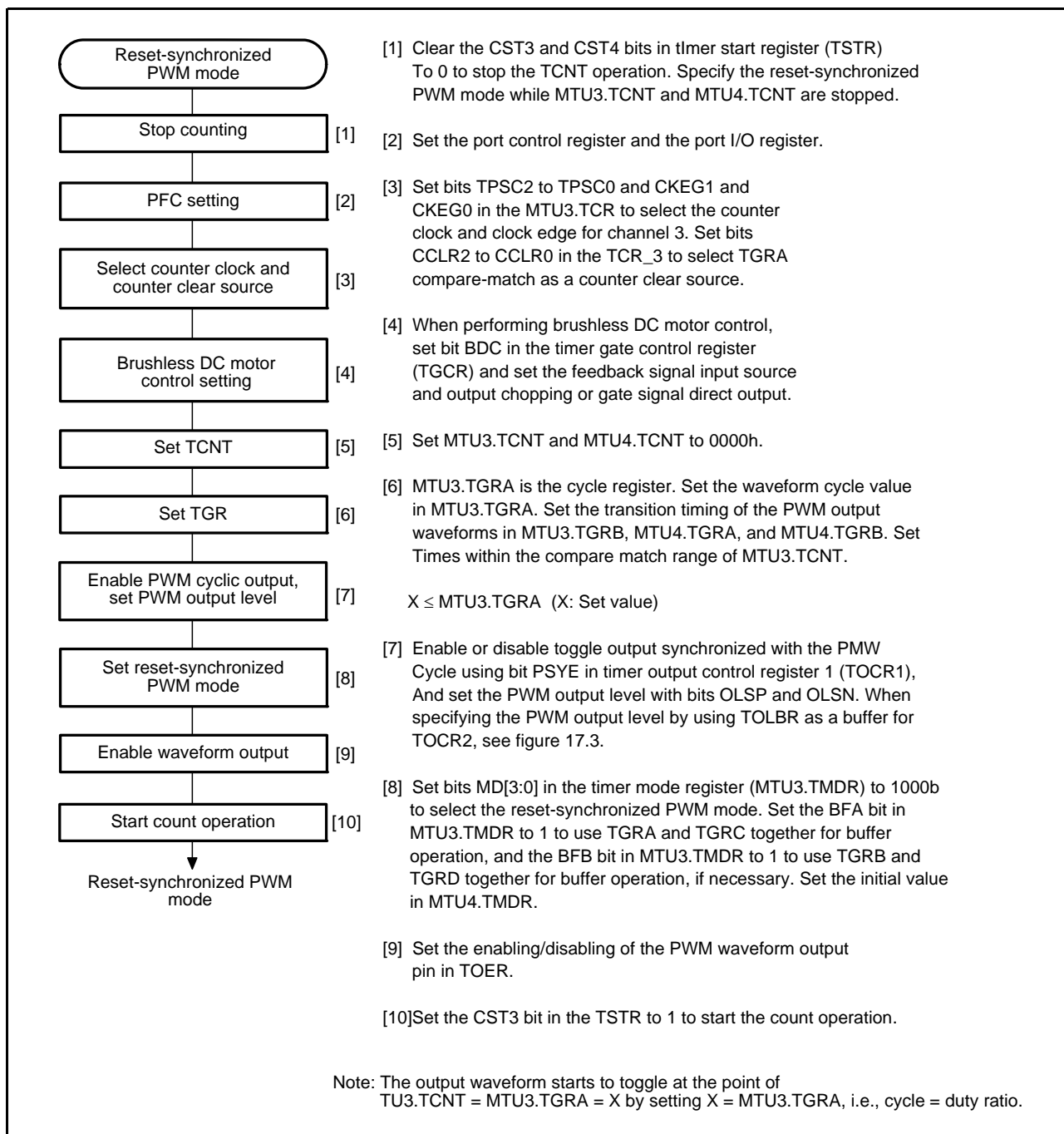
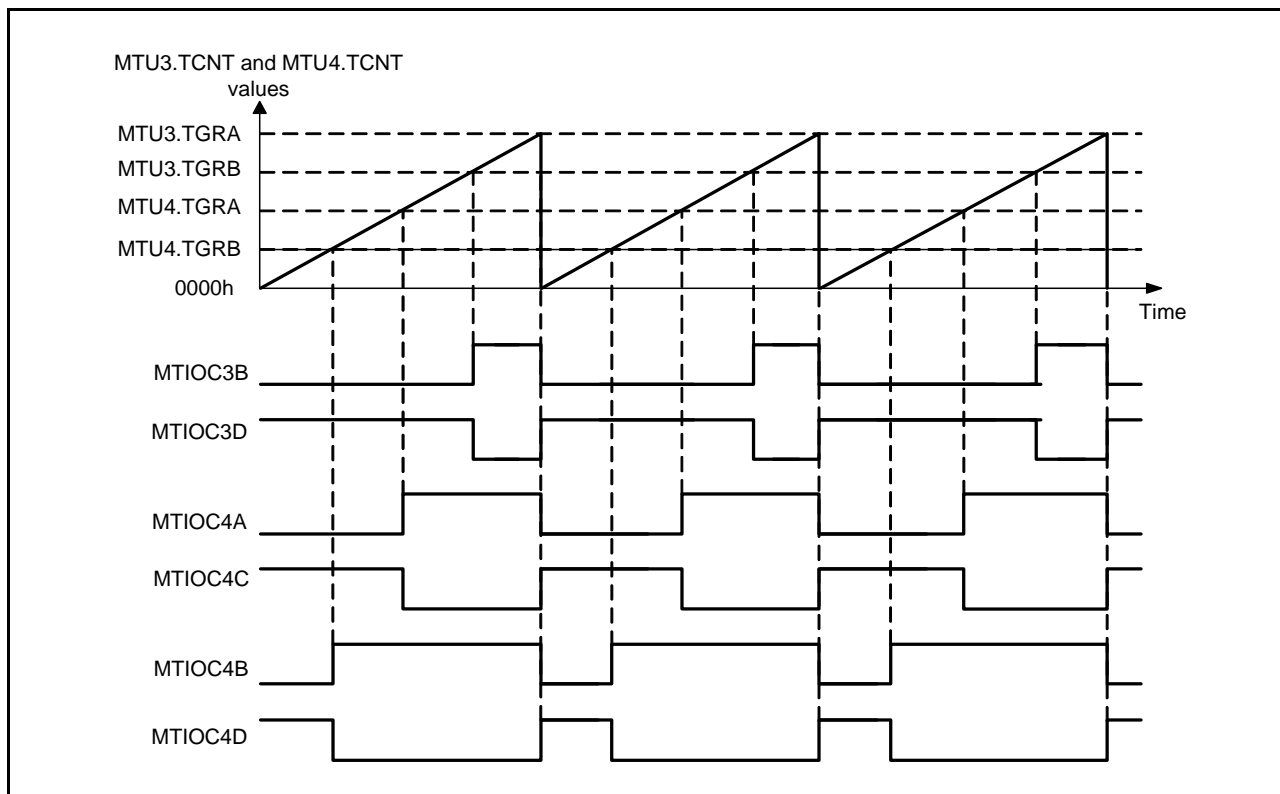


Figure 18.36 Procedure for Selecting Reset-Synchronized PWM Mode (Unit 0)

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 18.37 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 18.37 Example of Reset-Synchronized PWM Mode Operation
(When TOCR's OLSN = 1 and OLSP = 1 in Unit 0)**

18.3.8 Complementary PWM Mode

In complementary PWM mode, three phases of non-overlapping positive and negative PWM waveforms can be output by combining channels 3 and 4 (or channels 9 and 10). PWM waveforms without non-overlapping interval are also available.

In complementary PWM mode, MTIOCnB, MTIOCnD, MTIOCmA, MTIOCmB, MTIOCmC, and MTIOCmD pins function as PWM output pins, and the MTIOCnA pin can be set for toggle output synchronized with the PWM cycle. MTUn.TCNT and MTUm.TCNT function as up/down-counters.

Table 18.56 shows the PWM output pins used. Table 18.57 shows the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

(n = 3 or 9, m = 4 or 10)

Table 18.56 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
MTU3 MTU9	MTIOC3A/ MTIOC9A	Toggle output synchronized with PWM cycle (or I/O port)
	MTIOC3B/ MTIOC9B	PWM output pin 1
	MTIOC3C/ MTIOC9C	I/O port *
	MTIOC3D/ MTIOC9D	PWM output pin 1' (non-overlapping negative-phase waveform of PWM output 1; PWM output without non-overlapping interval is also available)
MTU4 MTU10	MTIOC4A/ MTIOC10A	PWM output pin 2
	MTIOC4C/ MTIOC10C	PWM output pin 2' (non-overlapping negative-phase waveform of PWM output 2; PWM output without non-overlapping interval is also available)
	MTIOC4B/ MTIOC10B	PWM output pin 3
	MTIOC4D/ MTIOC10D	PWM output pin 3' (non-overlapping negative-phase waveform of PWM output 3; PWM output without non-overlapping interval is also available)

Note : * Avoid setting the MTIOC3C/MTIOC9C pin as a timer I/O pin in complementary PWM mode.

Table 18.57 Register Settings for Complementary PWM Mode

Channel	Counter/ Register	Description	Read/Write from CPU
MTU3/ MTU9	MTUn. TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWER setting *
	MTUn. TGRA	Set MTUn.TCNT upper limit value (1/2 carrier cycle + dead time)	Maskable by TRWER setting *
	MTUn. TGRB	PWM output 1 compare register	Maskable by TRWER setting *
	MTUn. TGRC	MTUn.TGRA buffer register	Always readable/writable
	MTUn. TGRD	PWM output 1/MTUn.TGRB buffer register	Always readable/writable
MTU4/ MTU10	MTUm. TCNT	Starts up-counting after being initialized to 0000h	Maskable by TRWER setting *
	MTUm. TGRA	PWM output 2 compare register	Maskable by TRWER setting *
	MTUm. TGRB	PWM output 3 compare register	Maskable by TRWER setting *
	MTUm. TGRC	PWM output 2/MTUm.TGRA buffer register	Always readable/writable
	MTUm. TGRD	PWM output 3/MTUm.TGRB buffer register	Always readable/writable
Timer dead time data register (TDDR)	Set MTUm.TCNT and MTUn.TCNT offset value (dead time value)	Maskable by TRWER setting *	
Timer cycle data register (TCDR)	Set MTUm.TCNT upper limit value (1/2 carrier cycle)	Maskable by TRWER setting *	
Timer cycle buffer register (TCBR)	TCDR buffer register	Always readable/writable	
Subcounter (TCNTS)	Subcounter for dead time generation	Read-only	
Temporary register 1 (TEMP1)	PWM output 1/MTUn.TGRB temporary register	Not readable/writable	
Temporary register 2 (TEMP2)	PWM output 2/MTUm.TGRA temporary register	Not readable/writable	
Temporary register 3 (TEMP3)	PWM output 3/MTUm.TGRB temporary register	Not readable/writable	

Note: [Legend] n = 3 or 9, m = 4 or 10

* Access can be enabled or disabled according to the setting in TRWER (timer read/write enable register).

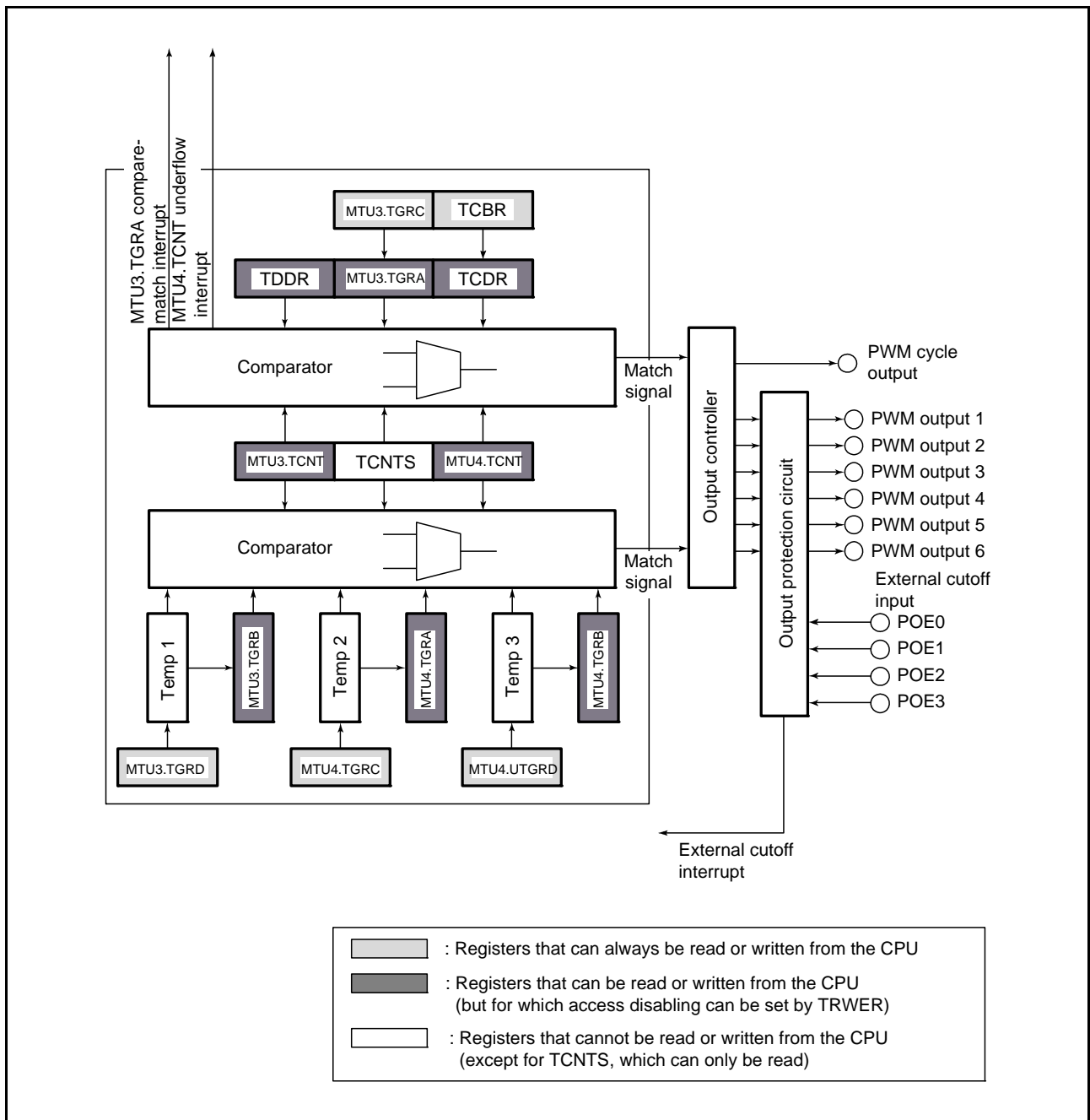


Figure 18.38 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

Figure 18.37 shows an example of the complementary PWM mode setting procedure.

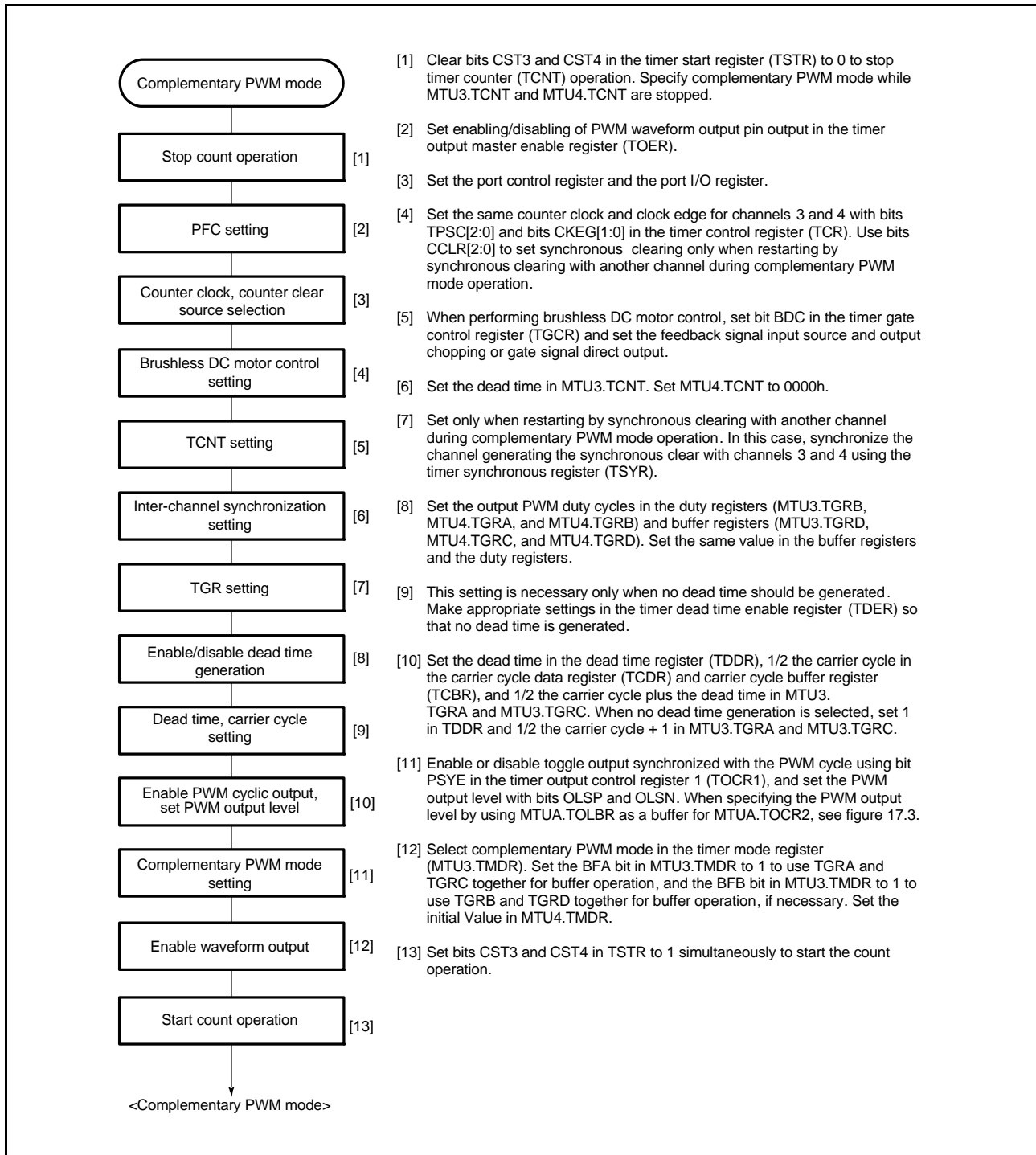


Figure 18.39 Example of Complementary PWM Mode (Unit 0) Setting Procedure

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases of PWM waveforms can be output. Figure 18.40 illustrates counter operation in complementary PWM mode, and Figure 18.41 shows an example of operation in complementary PWM mode.

(a) Counter Operation

In complementary PWM mode, three counters—MTUn.TCNT, MTUm.TCNT, and TCNTSj—in each unit perform up-/down-count operations.

MTUn.TCNT is automatically initialized to the value set in TDDRj when complementary PWM mode is selected and the CST bit in TSTRj is 0.

When the CST bit is set to 1, MTUn.TCNT counts up to the value set in MTUn.TGRA, then switches to down-counting when it matches MTUn.TGRA. When the MTUn.TCNT value matches TDDRj, the counter switches to up-counting, and the operation is repeated in this way.

MTUm.TCNT should be initialized to 0000h.

When the CST bit is set to 1, MTUm.TCNT counts up in synchronization with MTUn.TCNT, and switches to down-counting when it matches TCDR. On reaching 0000h, MTUm.TCNT switches to up-counting, and the operation is repeated in this way.

TCNTSj is a read-only counter. It does not need to be initialized.

When MTUn.TCNT matches TCDR during up-/down-counting of TCNT in channels 3 and 4 (or channels 9 and 10), TCNTSj starts down-counting, and when TCNTSj matches TCDR, the operation switches to up-counting. When TCNTSj matches MTUn.TGRA, it is cleared to 0000h.

When MTUm.TCNT matches TDDRj during down-counting of MTUn.TCNT and MTUm.TCNT, TCNTSj starts up-counting, and when TCNTSj matches TDDRj, the operation switches to down-counting. When TCNTSj reaches 0000h, it is set with the value in MTUn.TGRA.

TCNTSj is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation. (n = 3 or 9, m = 4 or 10, j = A, B)

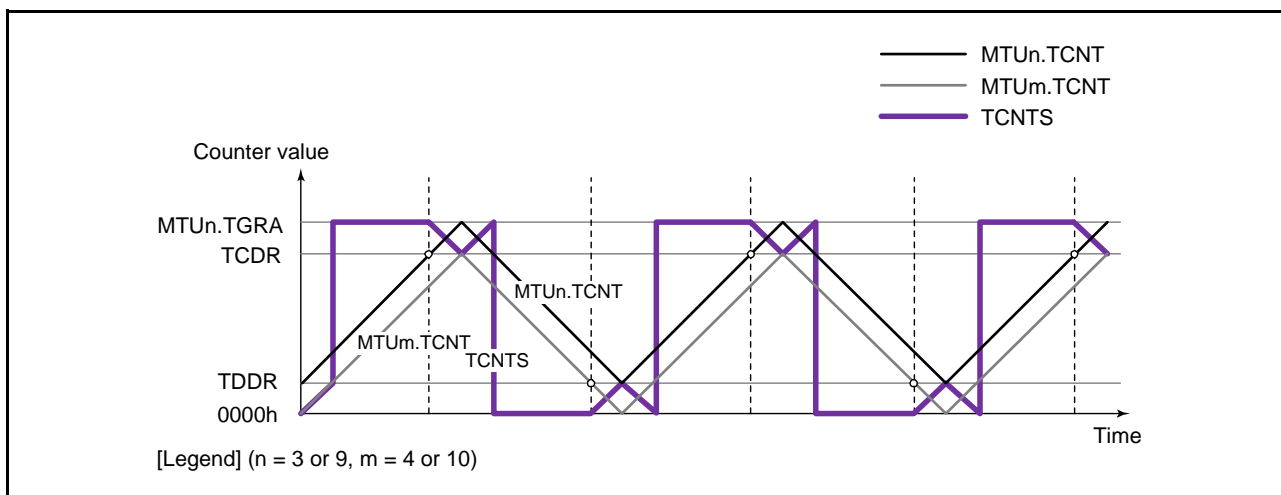


Figure 18.40 Counter Operation in Complementary PWM Mode

(b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used for each unit. Figure 18.41 shows an example of operation in complementary PWM mode.

MTUn.TGRB, MTUm.TGRA, and MTUm.TGRB are constantly compared with the counters to generate PWM waveforms. When these registers match the counter, the value set in bits OLSN and OLSP in the timer output control register (TOCR) is output. (n = 3 or 9, m = 4 or 10)

MTUn.TGRD, MTUm.TGRC, and MTUm.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. (n = 3 or 9, m = 4 or 10)

Data in a compare register can be changed by writing new data to the corresponding buffer register. The buffer registers can be read or written at any time.

The data written to a buffer register is constantly transferred to the temporary register in the Ta interval. Data is not transferred to the temporary register in the Tb interval. Data written to a buffer register in this interval is transferred to the temporary register at the end of the Tb interval.

The value transferred to a temporary register is transferred to the compare register when TCNTS for which the Tb interval ends matches MTUn.TGRA while TCNTS is counting up, or 0000h while counting down. The timing for transfer from the temporary register to the compare register can be selected with bits MD[3:0] in the timer mode register (TMDR). Figure 18.41 shows an example in which the trough is selected for the transfer timing. (n = 3 or 9, m = 4 or 10, j = A, B)

In the Tb interval in which data is not transferred to the temporary register (Tb1 in Figure 18.41), 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 MTUA.TCNTS in unit 0 or MTU9.TCNT, MTU10.TCNT and MTUB.TCNTS in unit 1) and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

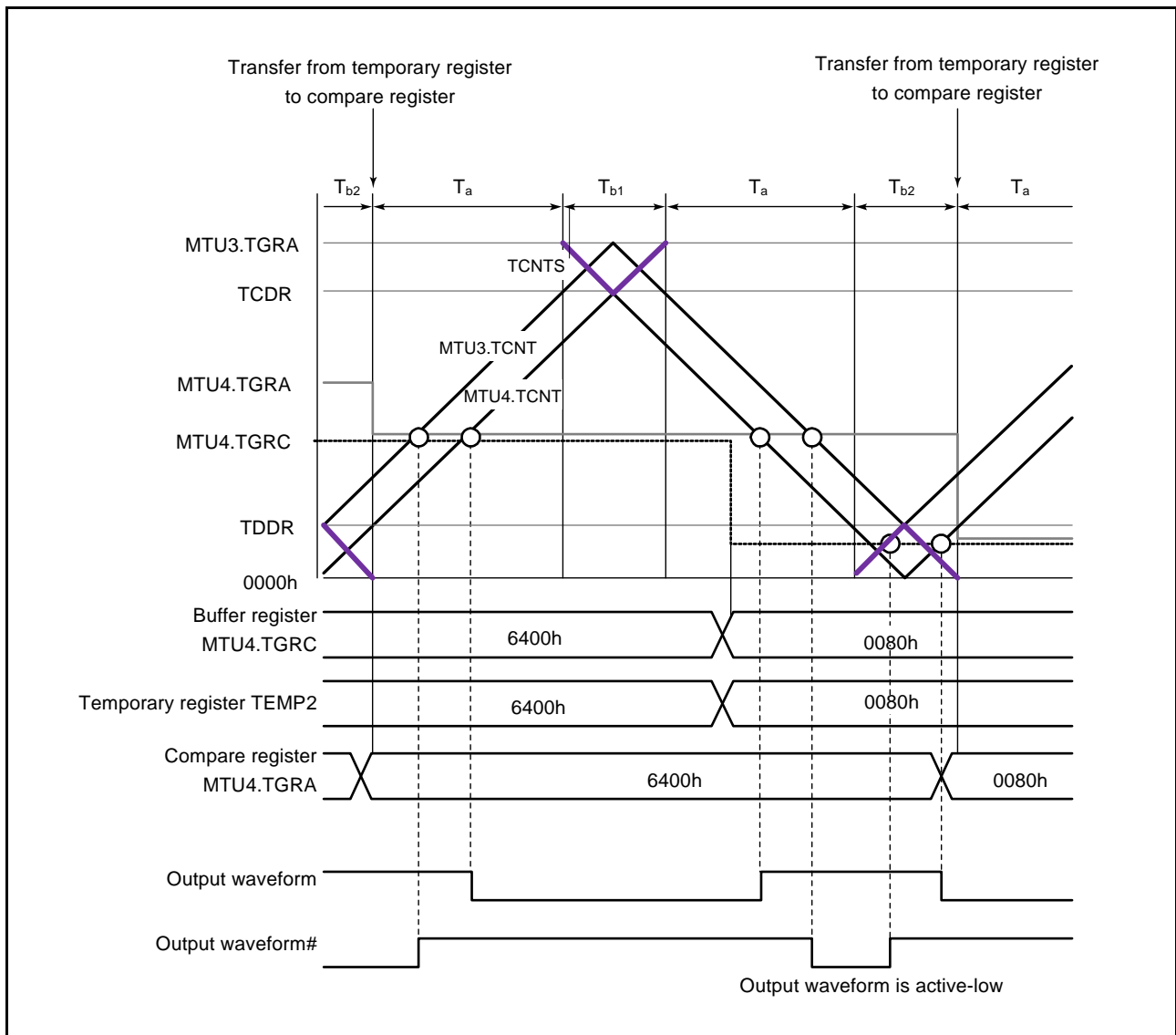


Figure 18.41 Example of Operation in Complementary PWM Mode

(c) Initial Setting

In complementary PWM mode, there are six registers that require initial setting. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).

Before setting complementary PWM mode with bits MD[3:0] in the timer mode register (TMDR), initial values should be set in the following registers.

MTUn.TGRC operates as the buffer register for MTUn.TGRA ($n = 3$ or 9), and should be set with $1/2$ the PWM carrier cycle + dead time T_d . The timer cycle buffer register (TCBR) operates as the buffer register for the timer cycle data register (TCDR), and should be set with $1/2$ the PWM carrier cycle ($j = A, B$). Set dead time T_d in the timer dead time data register (TDDR).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDER) should be cleared to 0, MTUn.TGRC and MTUn.TGRA ($n = 3$ or 9) should be set to $1/2$ the PWM carrier cycle + 1, and TDDR should be set to 1.

Set the respective initial PWM duty ratios in three buffer registers MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD in unit 0 or MTU9.TGRD, MTU10.TGRC, and MTU10.TGRD in unit 1.

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 MTUm.TCNT ($m = 4$ or 10) to 0000h before setting complementary PWM mode.

Table 18.58 Registers and Counters Requiring Initial Setting

Register and Counter	Setting
MTUn.TGRC ($n = 3, 9$)	$1/2$ PWM carrier cycle + dead time T_d ($1/2$ PWM carrier cycle + 1 when dead time generation is disabled by TDER)
TDDR	Dead time T_d (1 when dead time generation is disabled by TDER)
TCBR	$1/2$ PWM carrier cycle
MTUn.TGRD, MTUm.TGRC, MTUm.TGRD ($n = 3$ or $9, m = 4$ or 10)	Initial PWM duty ratio for each phase
MTUm.TCNT ($m = 4$ or 10)	0000h

Note: The value set in MTUn.TGRC ($n = 3$ or 9) should be the sum of $1/2$ the PWM carrier cycle set in TCBR and dead time T_d set in TDDR. When dead time generation is disabled by TDER, TGRC should be set to $1/2$ the PWM carrier cycle + 1.

(d) PWM Output Level Setting

In complementary PWM mode, the PWM pulse output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2).

The output level can be set for each of the three positive phases and three negative phases of 6-phase output.

Complementary PWM mode should be cleared before setting or changing output levels.

(e) Dead Time Setting

In complementary PWM mode, PWM pulses are output with a non-overlapping relationship between the positive and negative phases. This non-overlap time is called the dead time.

The non-overlap time is set in the timer dead time data register (TDDR). The value set in TDDR is used as the MTUn.TCNT counter start value and creates a non-overlapping interval between MTUn.TCNT and MTUm.TCNT. Complementary PWM mode should be cleared before changing the contents of TDDR. ($n = 3$ or $9, m = 4$ or 10)

(f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER bit in the timer dead time enable register (TDER) to 0. TDER can be cleared to 0 only when 0 is written to it after reading TDER = 1.

MTUn.TGRA and MTUn.TGRC (n = 3 or 9) should be set to 1/2 PWM carrier cycle + 1 and the timer dead time data register (TDDR) should be set to 1.

By the above settings, PWM waveforms without dead time can be obtained. Figure 18.42 shows an example of operation without dead time.

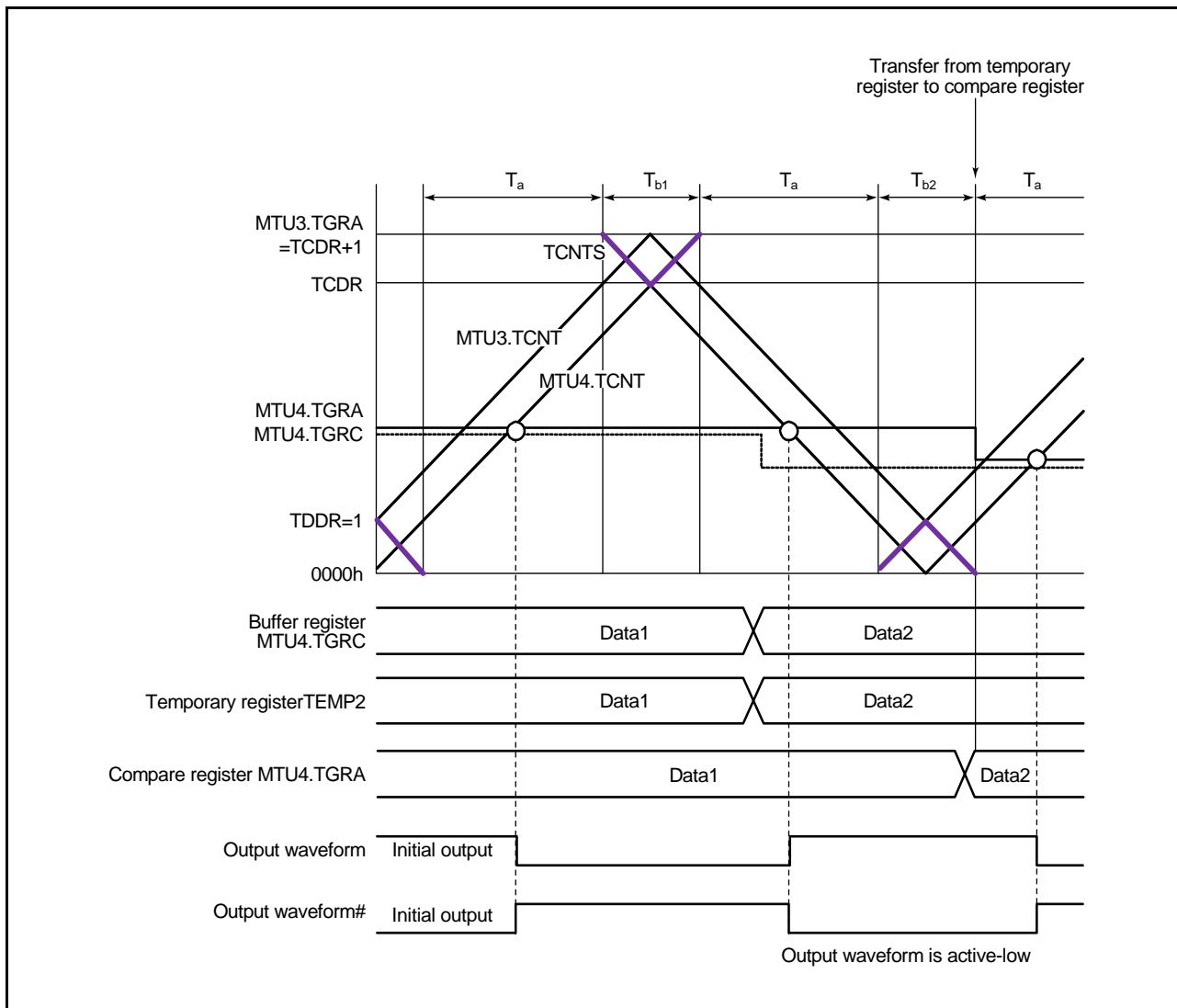


Figure 18.42 Example of Operation without Dead Time

(g) PWM Cycle Setting

In complementary PWM mode, the PWM pulse cycle is set in two registers—MTUn.TGRA (n = 3 or 9), in which the MTUn.TCNT upper limit value is set, and TCDR, in which the MTUm.TCNT (m = 4 or 10) upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: MTU3.TGRA or MTU9.TGRA setting = TCDR setting + TDDR setting

Without dead time: MTU3.TGRA or MTU9.TGRA setting = TCDR setting + 1

The MTUn.TGRA and TCDR settings are made by setting values in buffer registers MTUn.TGRC and TCBR. The values set in MTUn.TGRC and TCBR are transferred simultaneously to MTUn.TGRA and TCDR with the transfer timing selected with bits MD[3:0] in the timer mode register (TMDR). (n = 3 or 9)

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 18.43 illustrates the operation when the PWM cycle is updated at the crest.

See the following section, (h) Register Data Updating, for the method of updating the data in each buffer register.

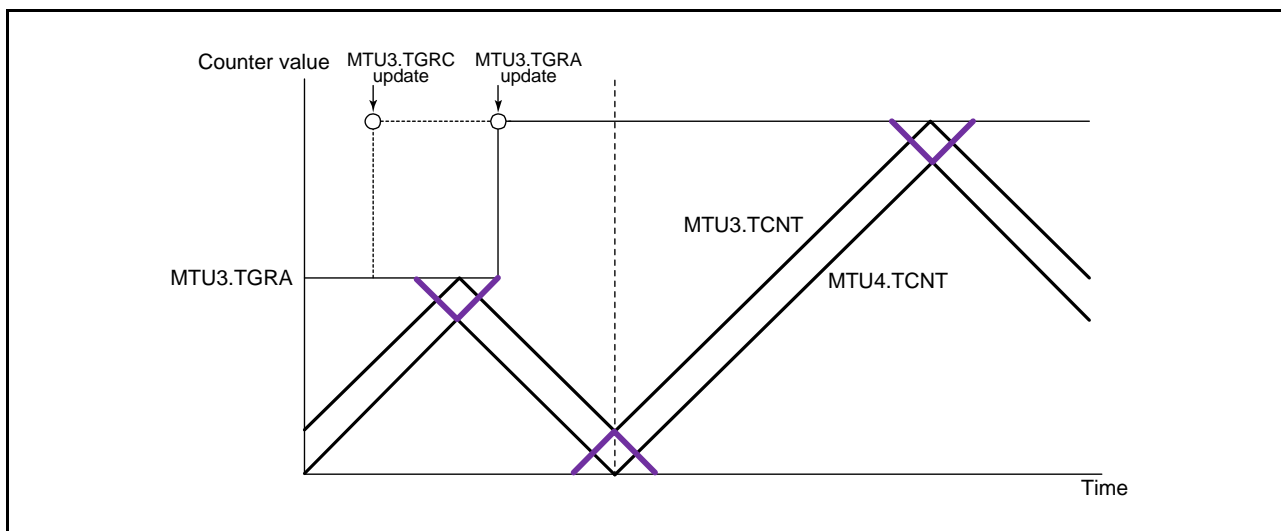


Figure 18.43 Example of PWM Cycle Updating

(h) Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five registers (PWM duty and carrier cycle registers) that have buffer registers and can be updated during operation.

There is a temporary register between each of these registers and its buffer register. While subcounter TCNTS is not counting, if buffer register data is updated, the temporary register value also changes. Data is not transferred from buffer registers to temporary registers while TCNTS is counting; in this case, the value written to a buffer register is transferred after TCNTS halts.

The temporary register value is transferred to the compare register at the data update timing set with bits MD[3:0] in the timer mode register (TMDR). Figure 18.44 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 MTUm.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 MTUm.TGRD. (m = 4 or 10)

Even when not updating all five registers or when not updating the MTUm.TGRD data, be sure to write to MTUm.TGRD after writing data to the registers to be updated. In this case, the data written to MTUm.TGRD should be the same as the data prior to the write operation. (m = 4 or 10)

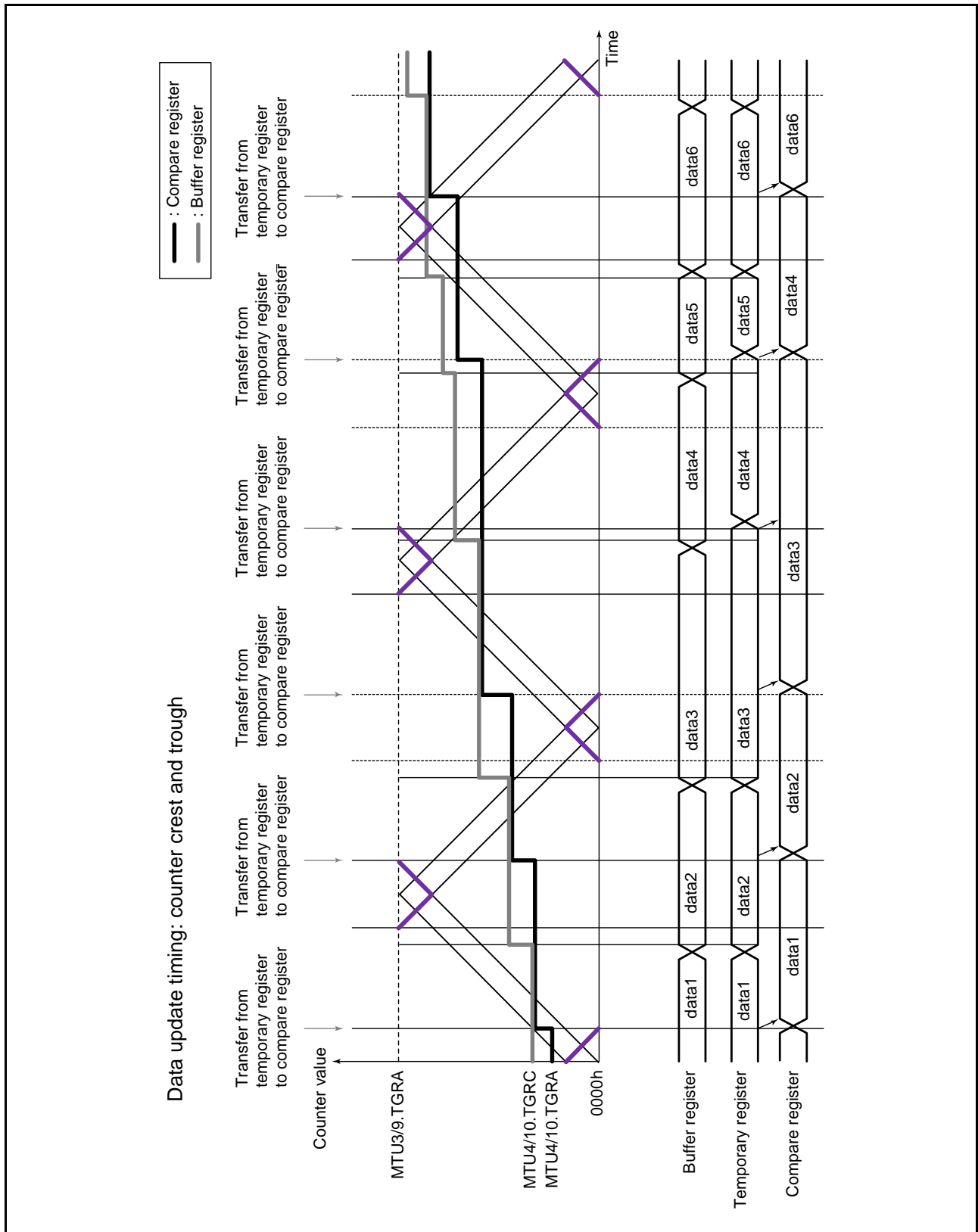


Figure 18.44 Example of Data Updating in Complementary PWM Mode

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1N to OLS3N and OLS1P to OLS3P in timer output control register 2 (TOCR2). This initial output is the non-active level of the PWM pulse and continues from when complementary PWM mode is set with the timer mode register (TMDR) until MTU4.TCNT (or MTU10.TCNT) exceeds the value set in the dead time register (TDDR). Figure 18.45 shows an example of the initial output in complementary PWM mode.

An example of the waveform when the initial PWM duty ratio is smaller than the TDDR value is shown in Figure 18.46.

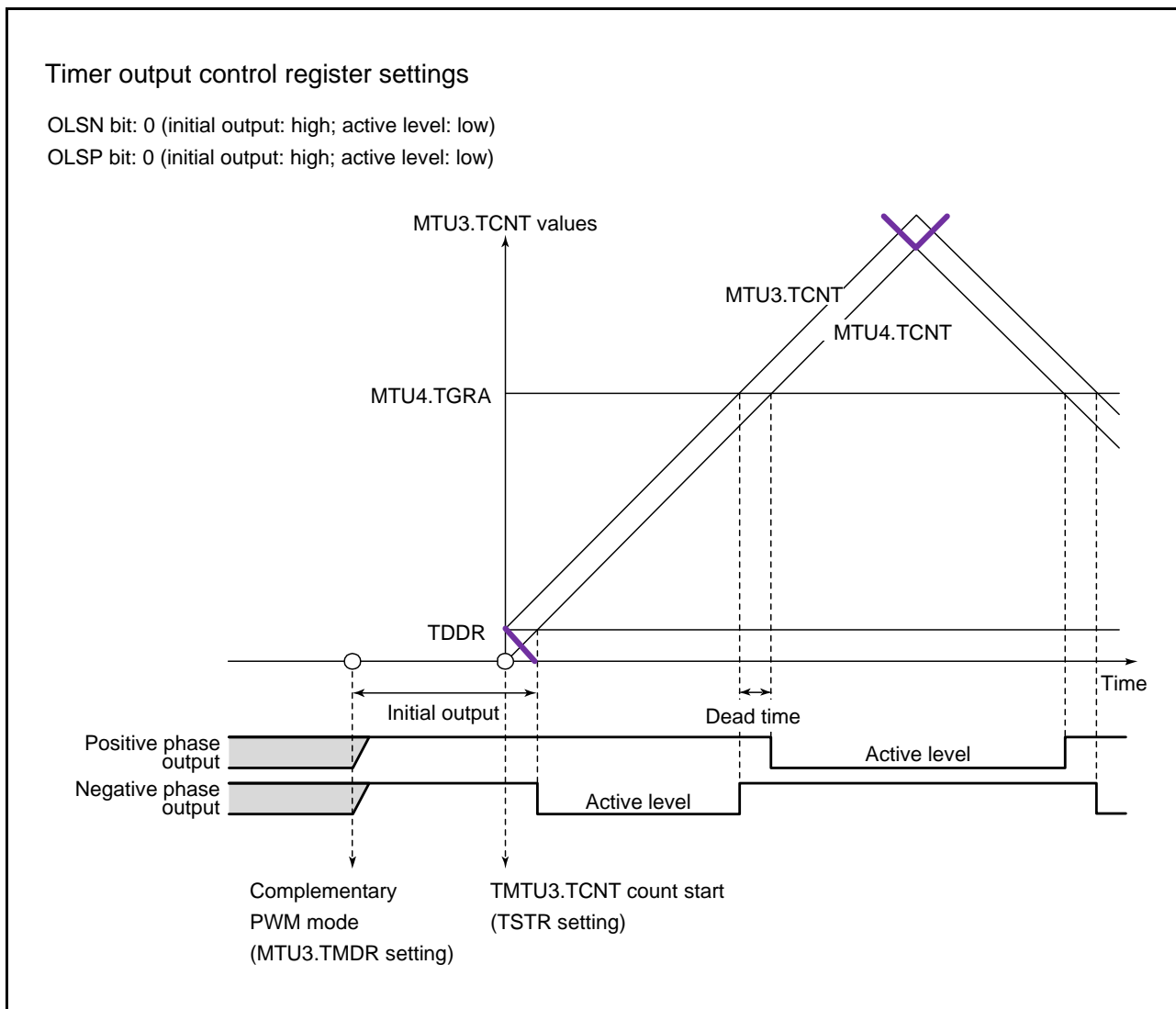


Figure 18.45 Example of Initial Output in Complementary PWM Mode (1) (Unit 0)

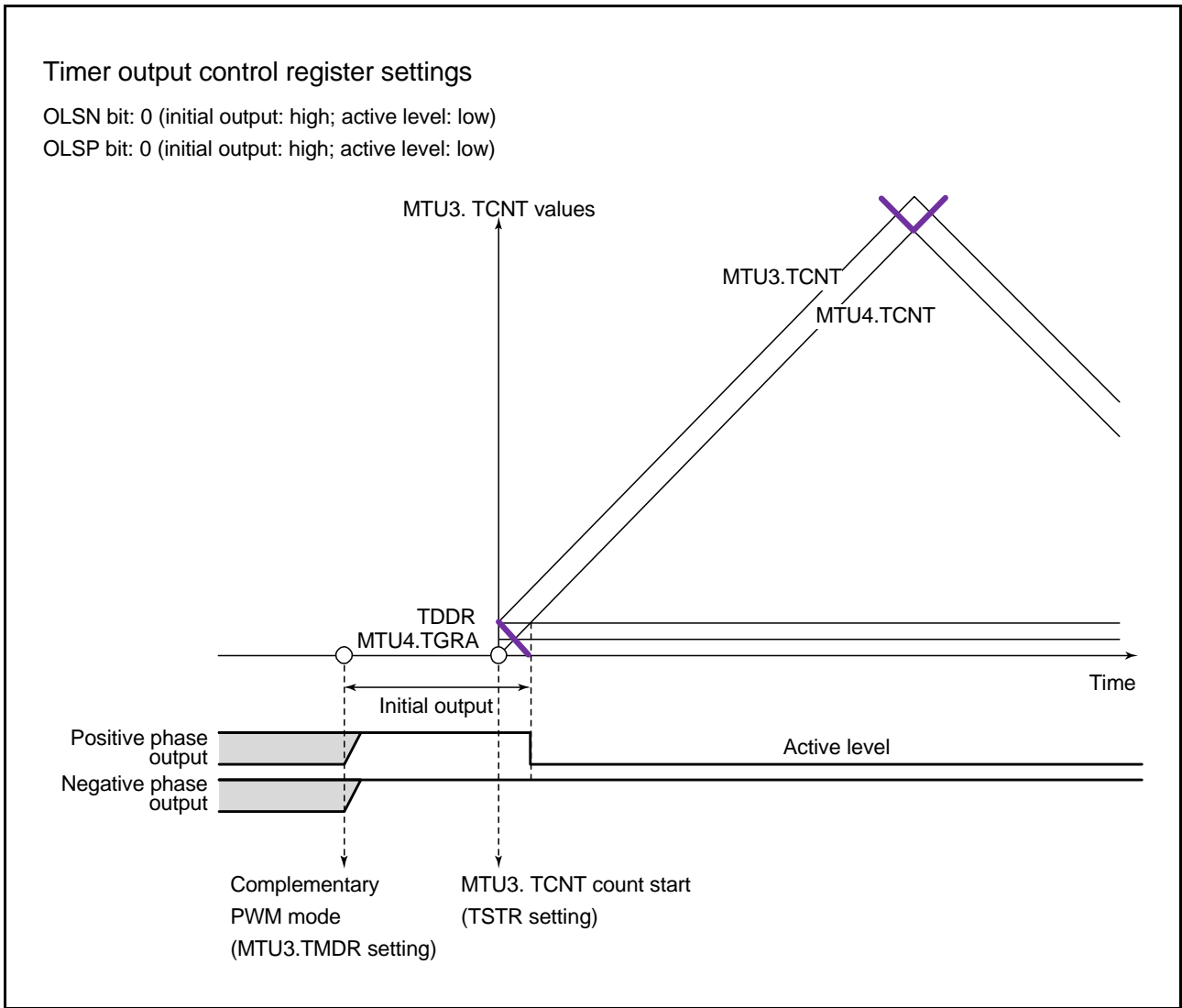


Figure 18.46 Example of Initial Output in Complementary PWM Mode (2) (Unit 0)

(j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, three phases of PWM waveforms are output with a non-overlap time between the positive and negative phases. This non-overlap time is called the dead time.

A PWM waveform is generated by output of the level selected in the timer output control register in the event of a compare match between a counter and a data register. While TCNTS is counting, the data register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100%. The relative timing of turn-on and turn-off compare match occurrence may vary, but the compare match that turns off each phase takes precedence to secure the dead time and ensure that the positive-phase and negative-phase turn-on times do not overlap. Figure 18.47 to Figure 18.49 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 value indicated in solid lines in the figure, and the turn-on timing by a compare match with the counter indicated in dotted lines, which operates with a delay of the dead time behind the solid-line counter. 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 $a \rightarrow b \rightarrow c \rightarrow d$ (or $c \rightarrow d \rightarrow a' \rightarrow b'$) as shown in Figure 18.47.

If compare matches deviate from the $a \rightarrow b \rightarrow c \rightarrow d$ 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 $c \rightarrow d \rightarrow a' \rightarrow b'$ 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 18.48, if compare match c follows compare match a before compare match b, compare match b is ignored and the negative phase is turned on by compare match d. This is because turning off the positive phase has priority due to the occurrence of compare match c (positive-phase off timing) before compare match b (positive-phase on timing) (consequently, the waveform does not change because the positive phase goes from off to off).

Similarly, in the example in Figure 18.49, compare match a' with new data in the temporary register occurs before compare match c, but until compare match c, which turns off the positive phase, other compare matches are ignored. As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare matches at turn-off timings take precedence, and turn-on timing compare matches that occur before a turn-off timing compare match are ignored.

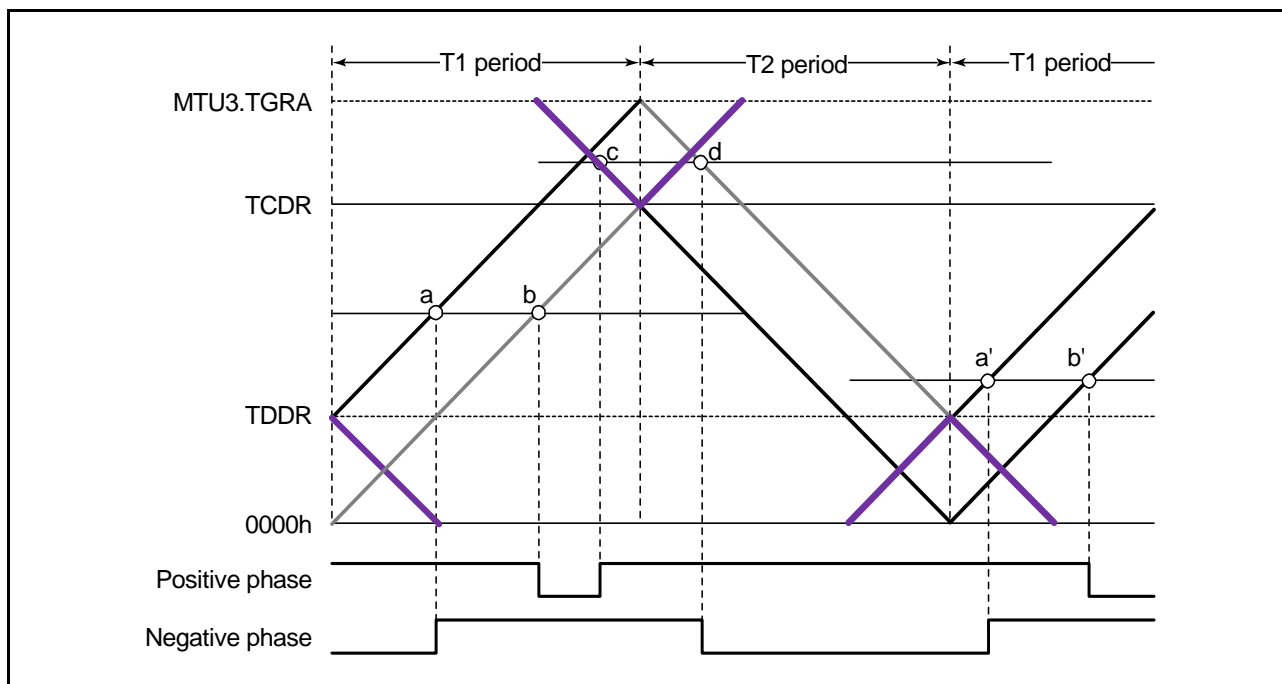


Figure 18.47 Example of Waveform Output in Complementary PWM Mode (1) (Unit 0)

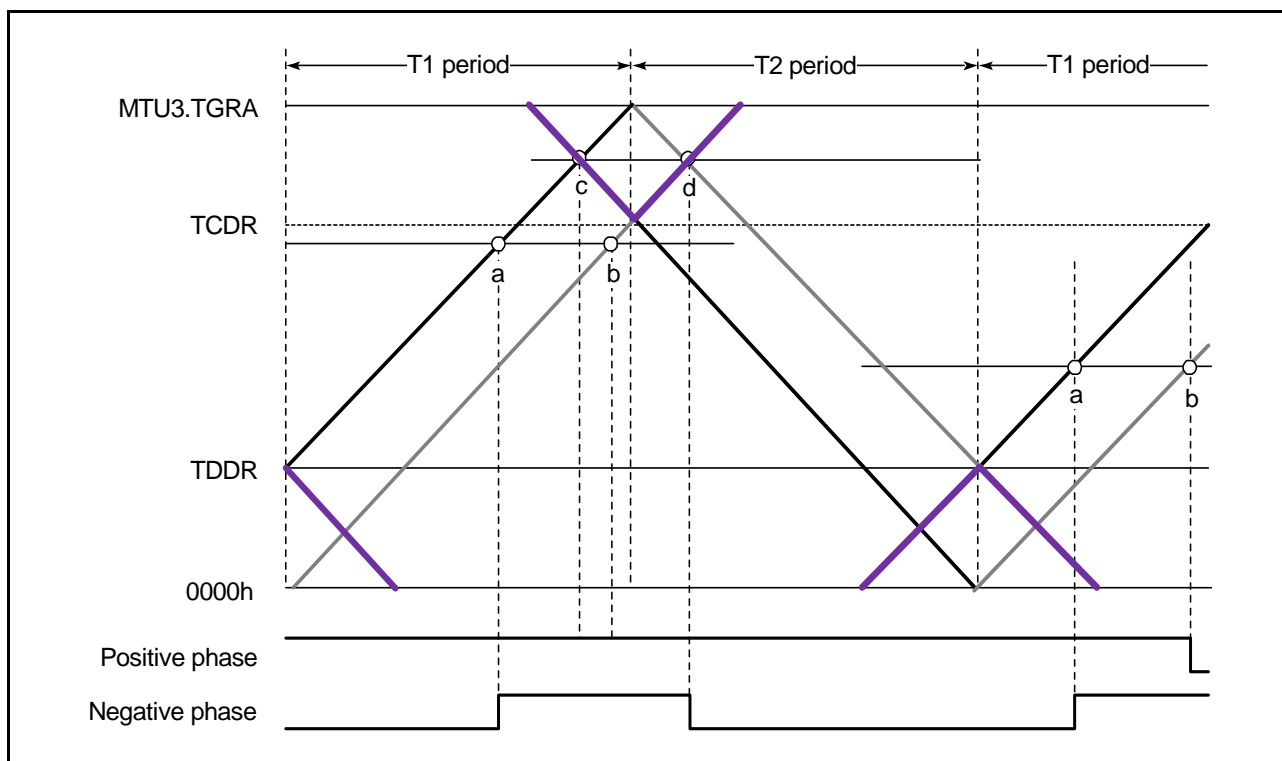


Figure 18.48 Example of Waveform Output in Complementary PWM Mode (2) (Unit 0)

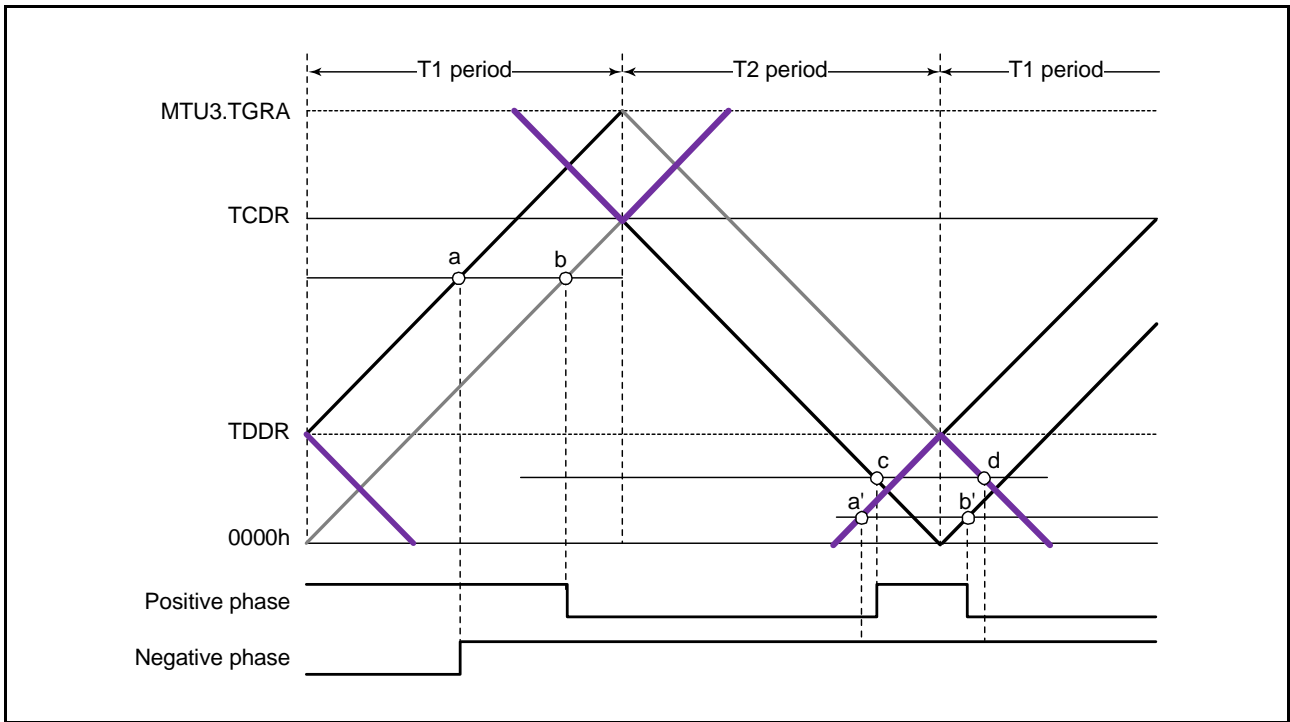


Figure 18.49 Example of Waveform Output in Complementary PWM Mode (3) (Unit 0)

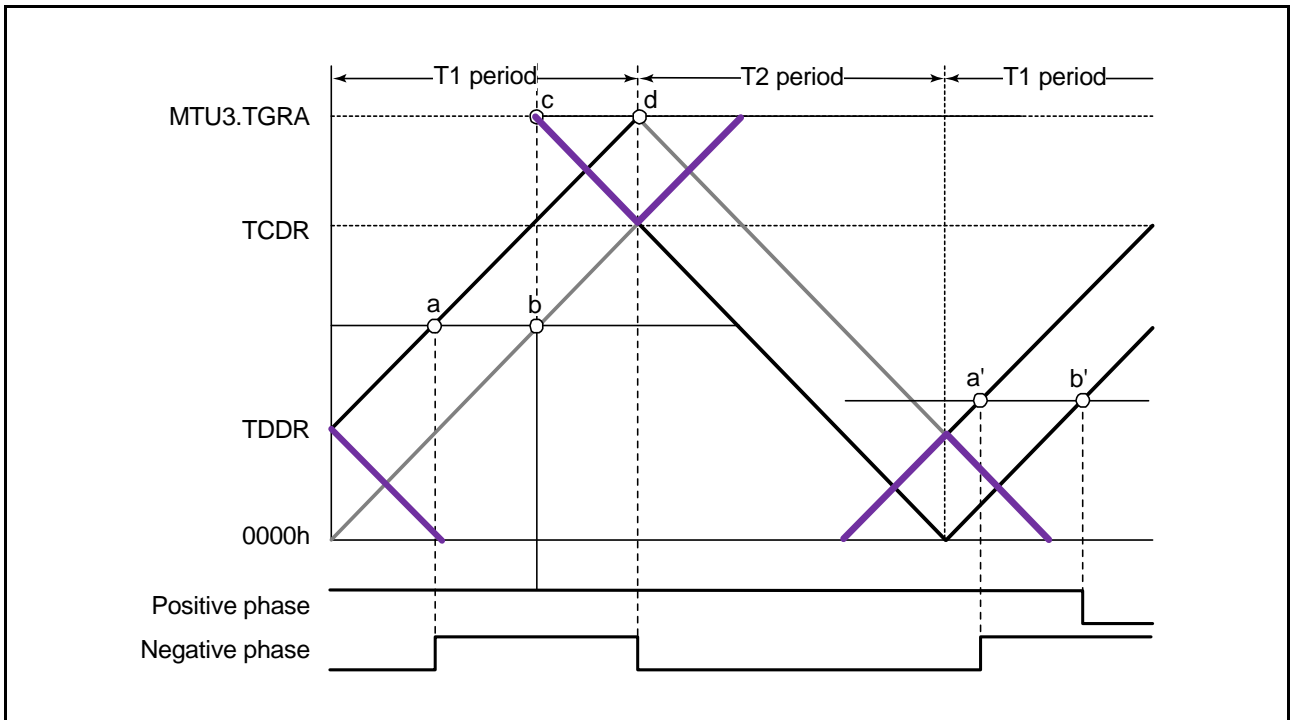


Figure 18.50 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1) (Unit 0)

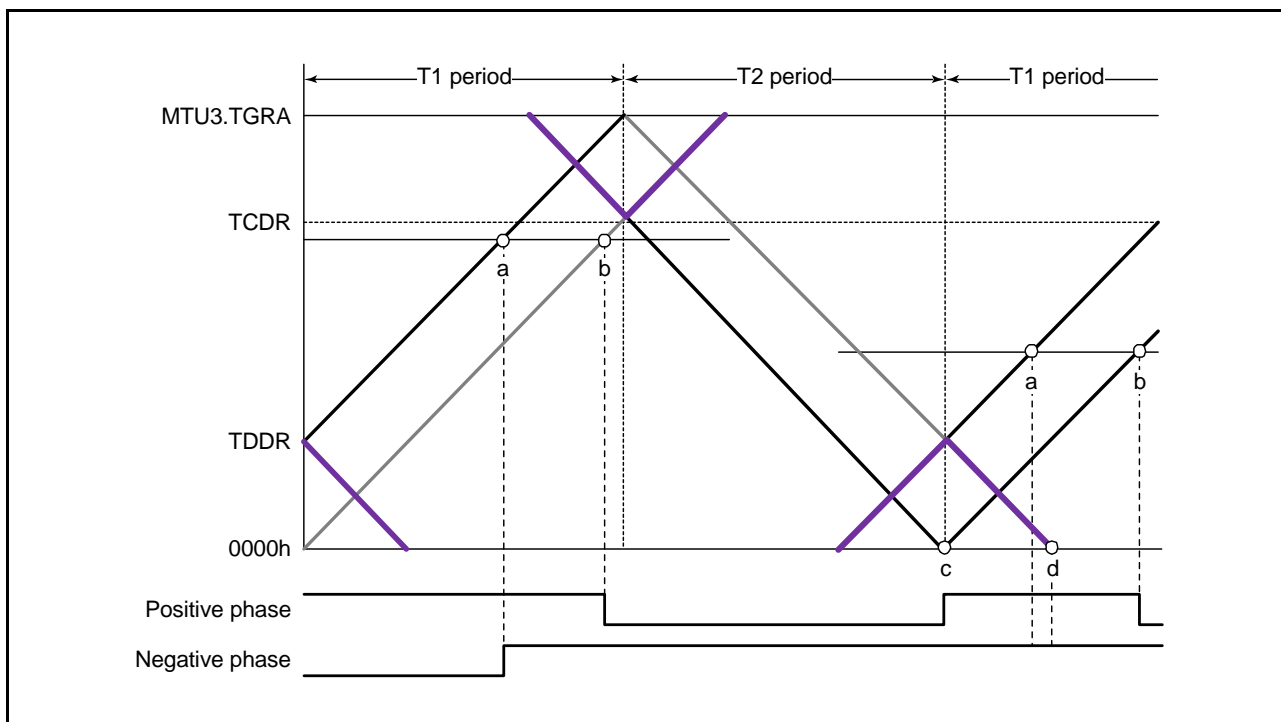


Figure 18.51 Example of 0% and 100% Waveform Output in Complementary PWM Mode (2) (Unit 0)

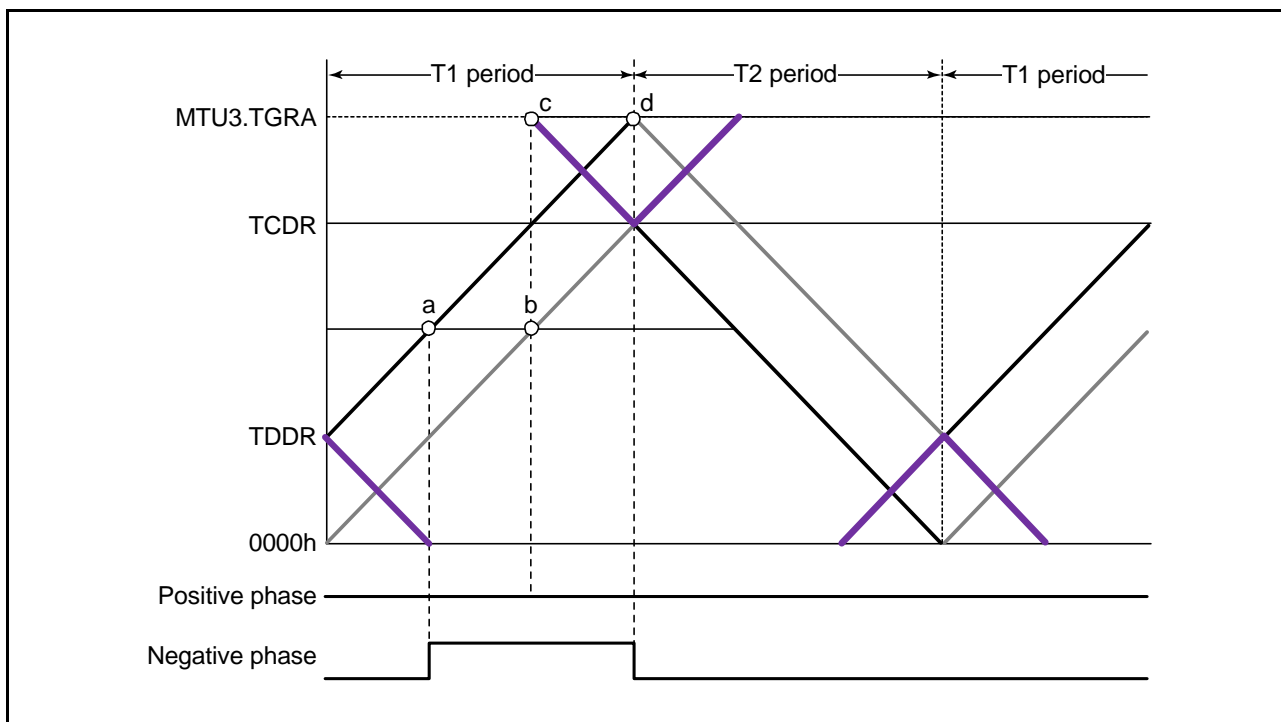


Figure 18.52 Example of 0% and 100% Waveform Output in Complementary PWM Mode (3) (Unit 0)

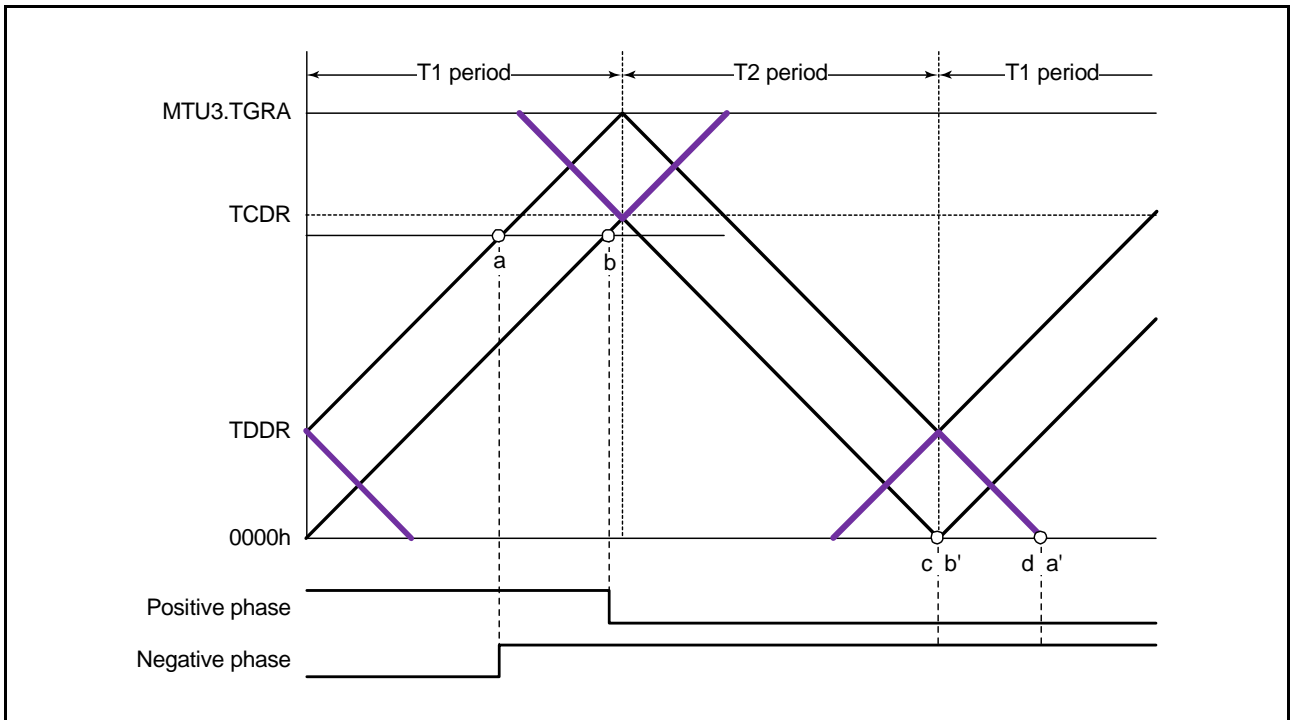


Figure 18.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (4) (Unit 0)

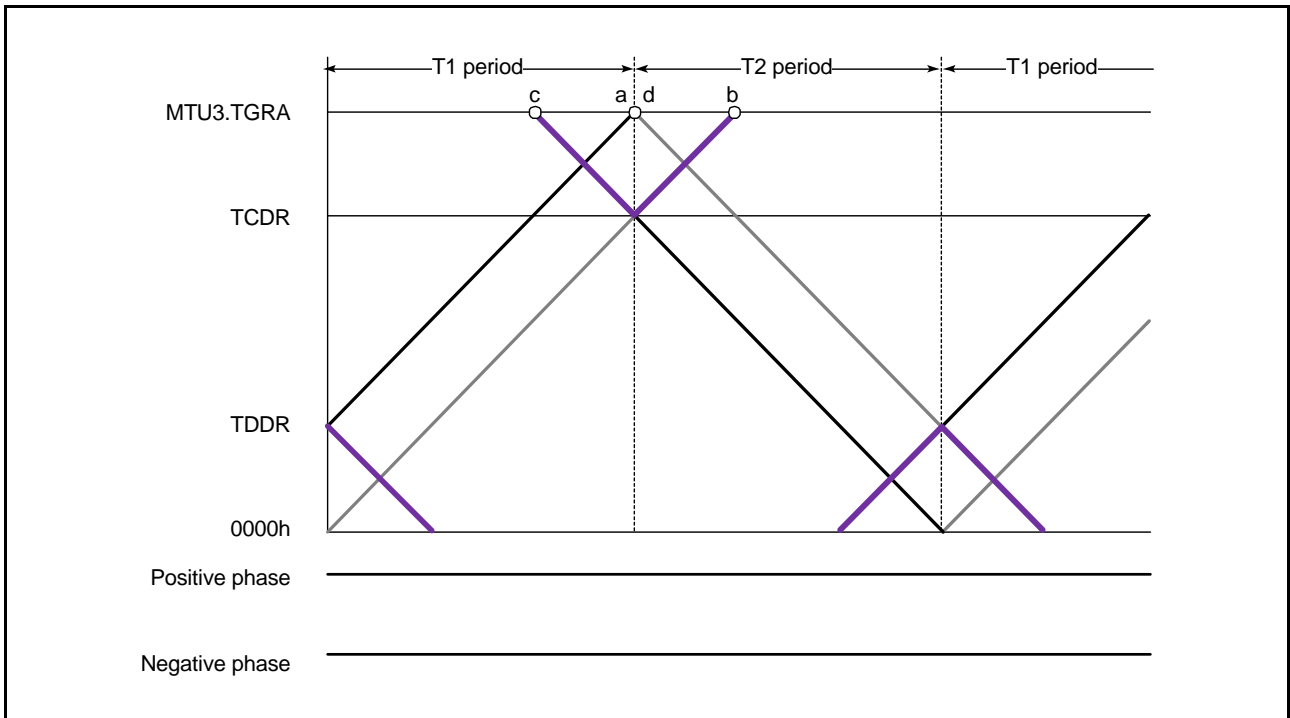


Figure 18.54 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5) (Unit 0)

(k) 0% and 100% Duty Ratio Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty ratio waveforms can be output as required. Figure 18.50 to Figure 18.53 show output examples.

A 100% duty ratio 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 ratio waveform is output when the data register value is set to the same value as MTUn.TGRA (n = 3 or 9). 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.

(l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output in synchronization with the PWM carrier cycle can be generated by setting the PSYE bit to 1 in the timer output control register (TOCR). An example of a toggle output waveform is shown in Figure 18.55.

This output is toggled by a compare match between MTUn.TCNT and MTUn.TGRA and a compare match between MTUm.TCNT and 0000h.

The MTIOCnA pin is assigned for this toggle output. The initial output is high. (n = 3 or 9, m = 4 or 10)

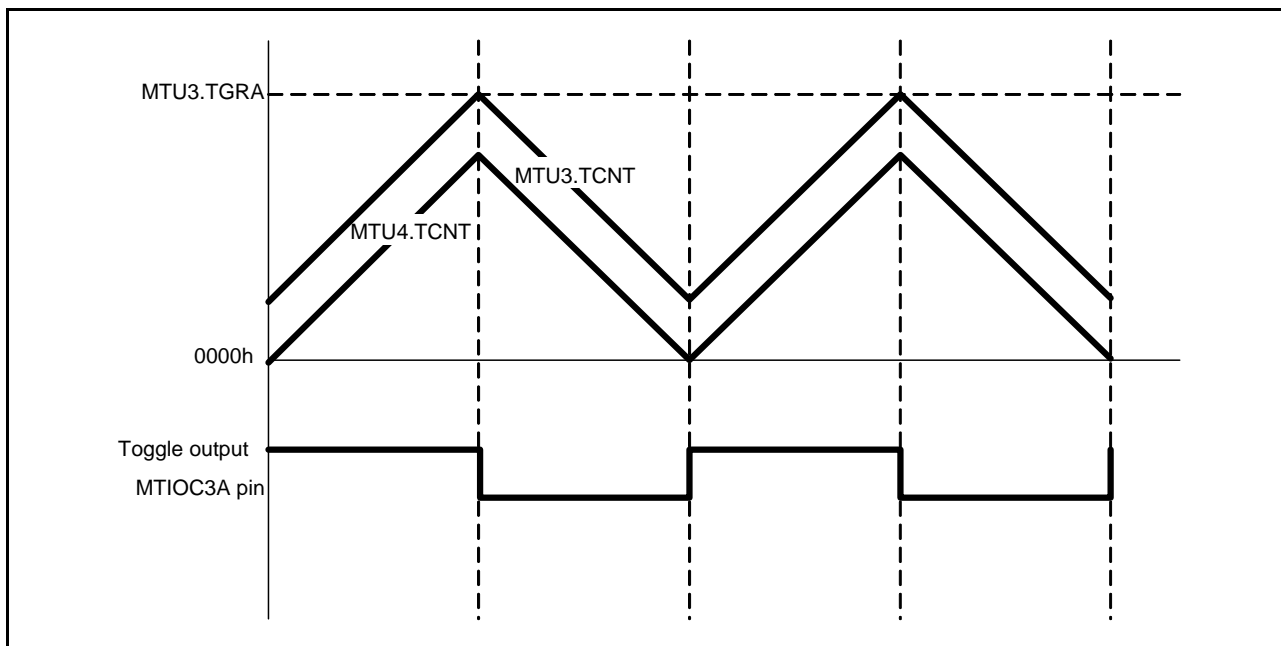


Figure 18.55 Example of Toggle Output Waveform Synchronized with PWM Output (Unit 0)

(m) Counter Clearing by Another Channel

In complementary PWM mode, MTUn.TCNT, MTUm.TCNT, and TCNTS can be cleared by another channel when a mode for synchronization with another channel is specified through the timer synchronous register (TSYR) and synchronous clearing is selected with bits CCLR[2:0] in the timer control register (TCR). (n = 3 or 9, m = 4 or 10)

Figure 18.56 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

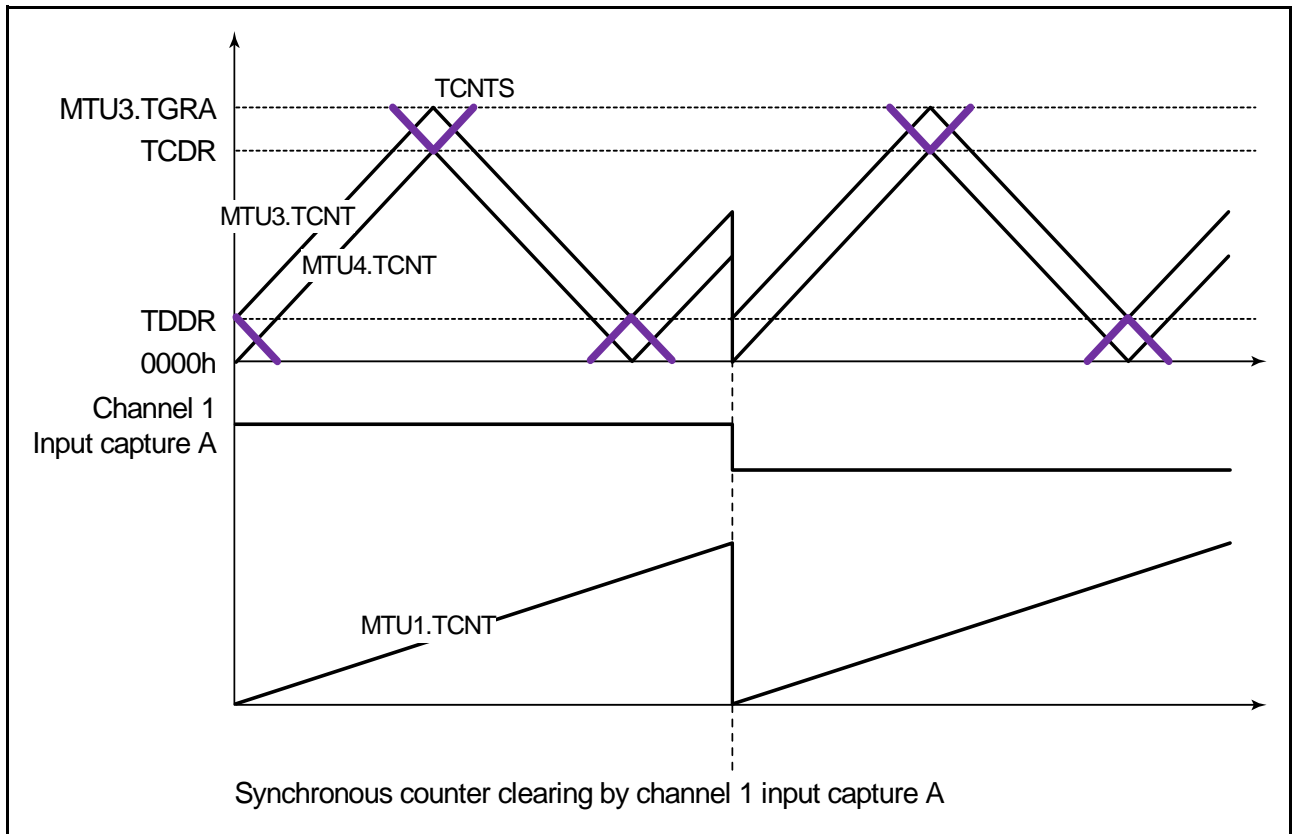


Figure 18.56 Counter Clearing Synchronized with Another Channel (Unit 0)

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCR to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval at the trough in complementary PWM mode and controls abrupt change in duty ratio at synchronous counter clearing.

Initial output suppression through the WRE bit = 1 is applicable only when synchronous clearing occurs in the Tb interval at the trough as indicated by (10) or (11) in Figure 18.57. When synchronous clearing occurs outside that interval, the initial value specified by the OLS bits in TOCR is output. Even in the Tb interval at the trough, if synchronous clearing occurs in the initial value output period (indicated by (1) in Figure 18.57) immediately after the counters start operation, initial value output is not suppressed.

Synchronous clearing generated in channels 0 to 2 or 6 to 8 can cause counter clearing in the MTU.

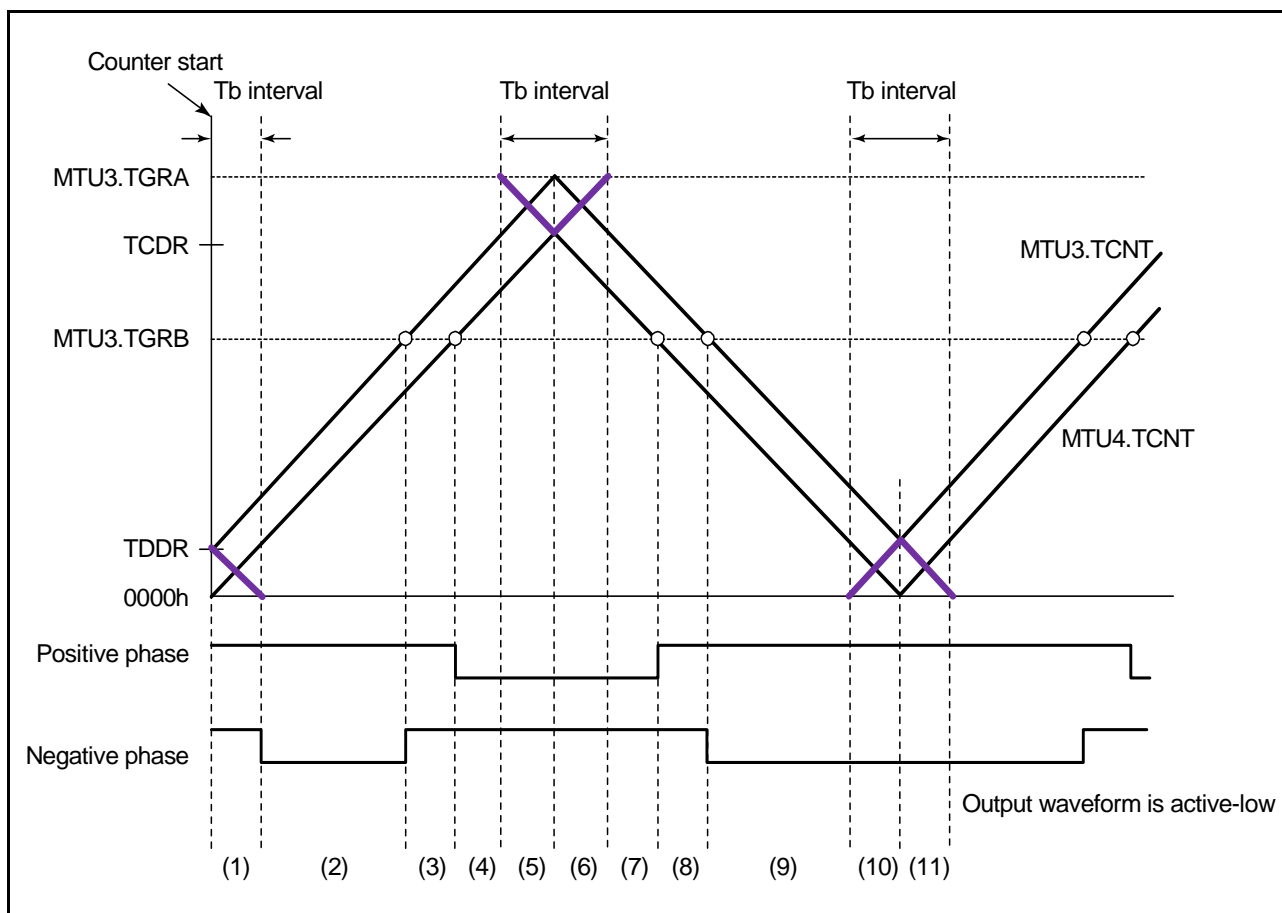


Figure 18.57 Timing for Synchronous Counter Clearing (Unit 0)

- 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 18.58.

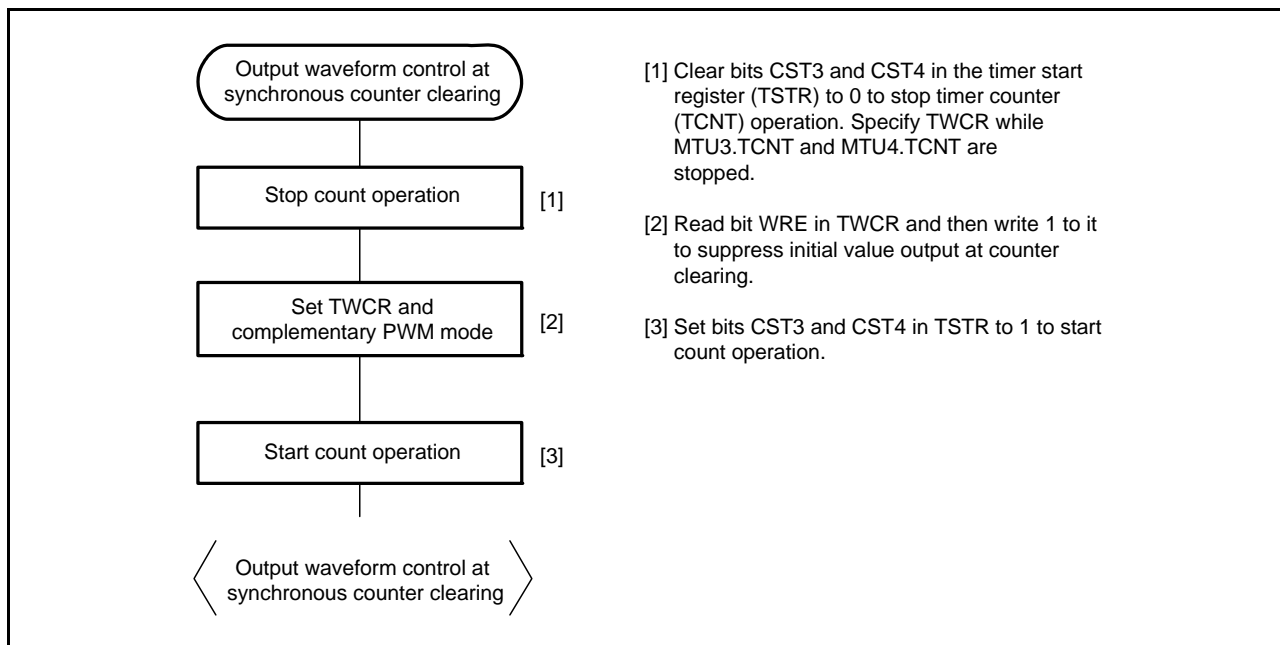


Figure 18.58 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 18.59 to Figure 18.62 show examples of output waveform control in which the MTU operates in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCR is set to 1. In the examples shown in Figure 18.59 to Figure 18.62, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 18.57, respectively.

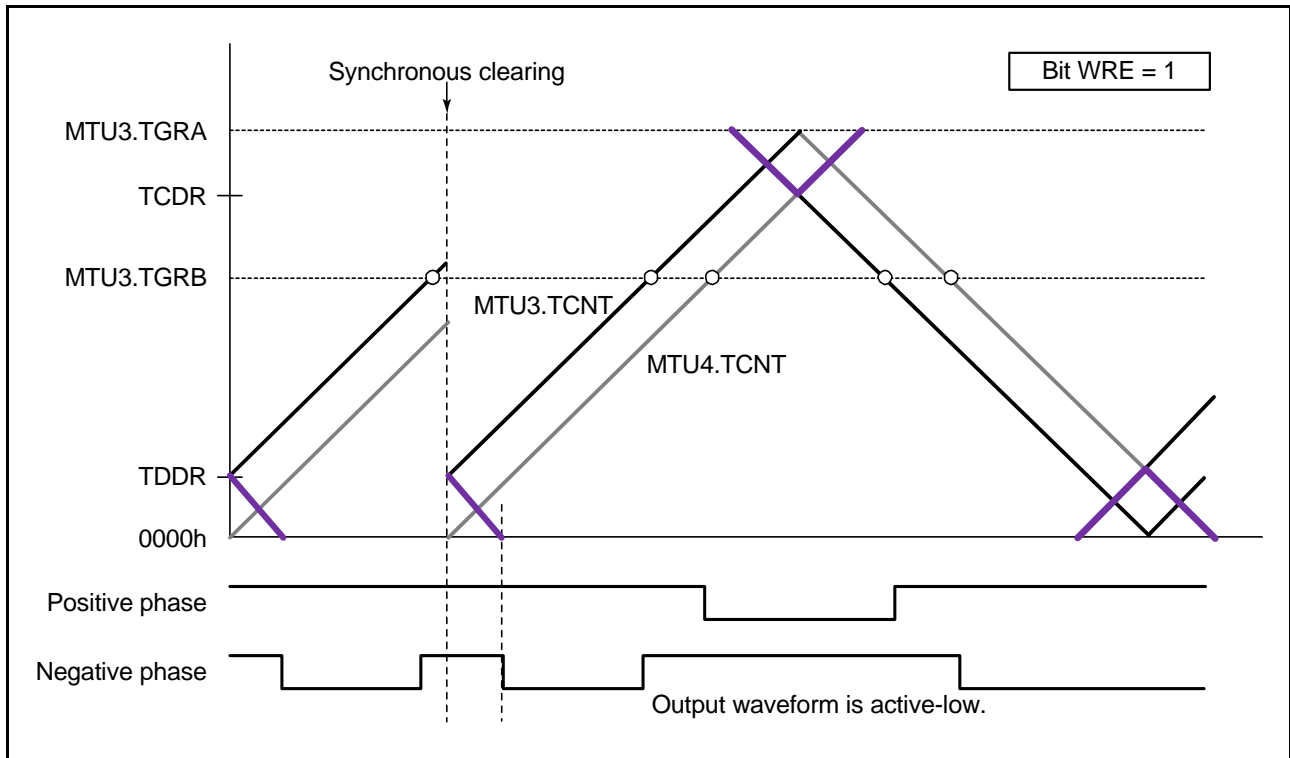


Figure 18.59 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 18.57; Bit WRE of TWCR is 1) (Unit 0)

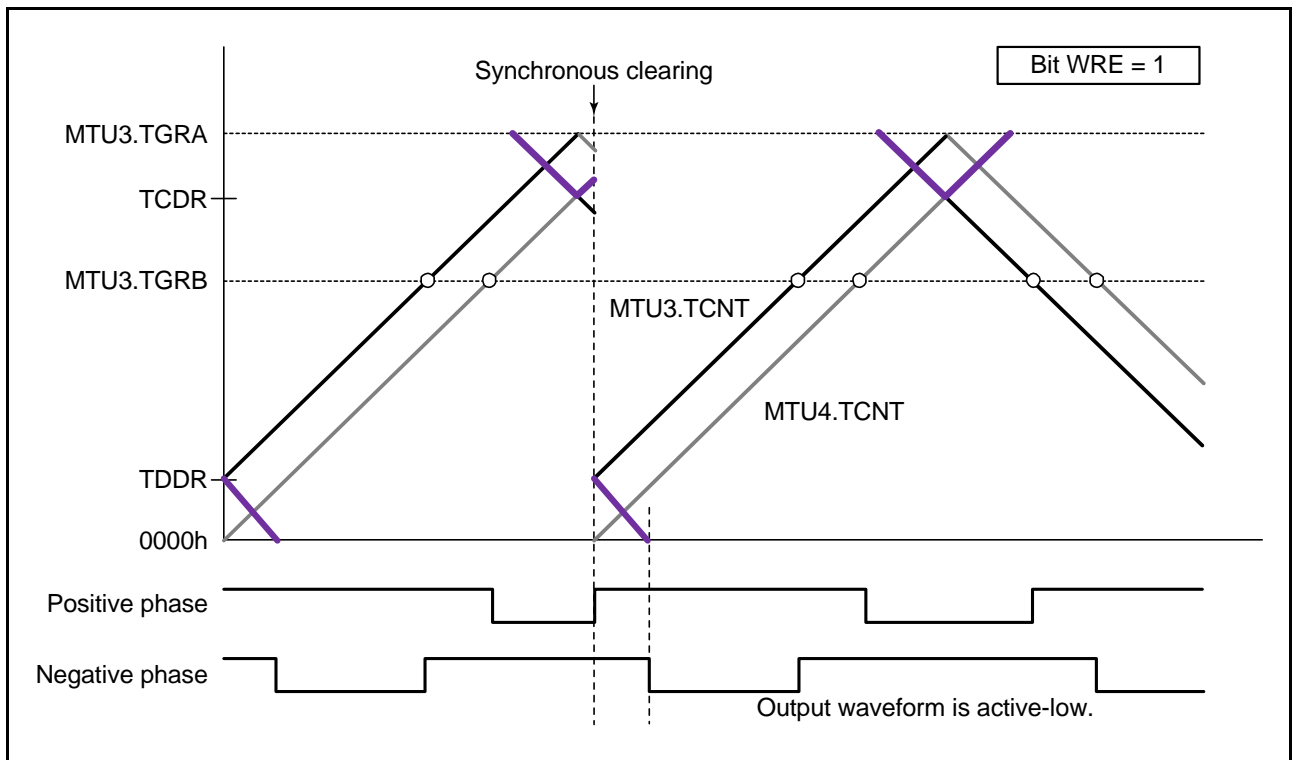


Figure 18.60 Example of Synchronous Clearing in Interval Tb at Crest
 (Timing (6) in Figure 18.57; Bit WRE of TWCR is 1) (Unit 0)

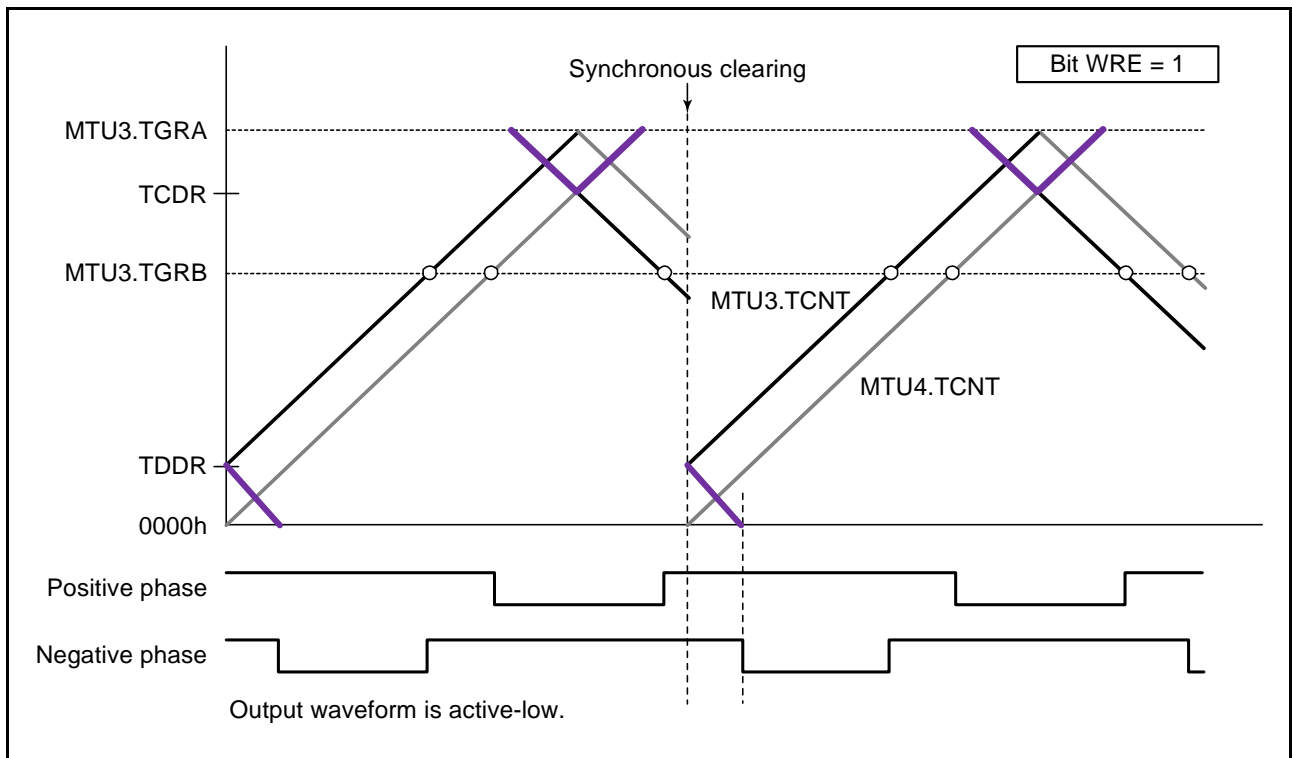


Figure 18.61 Example of Synchronous Clearing in Dead Time during Down-Counting
 (Timing (8) in Figure 18.57; Bit WRE of TWCR is 1) (Unit 0)

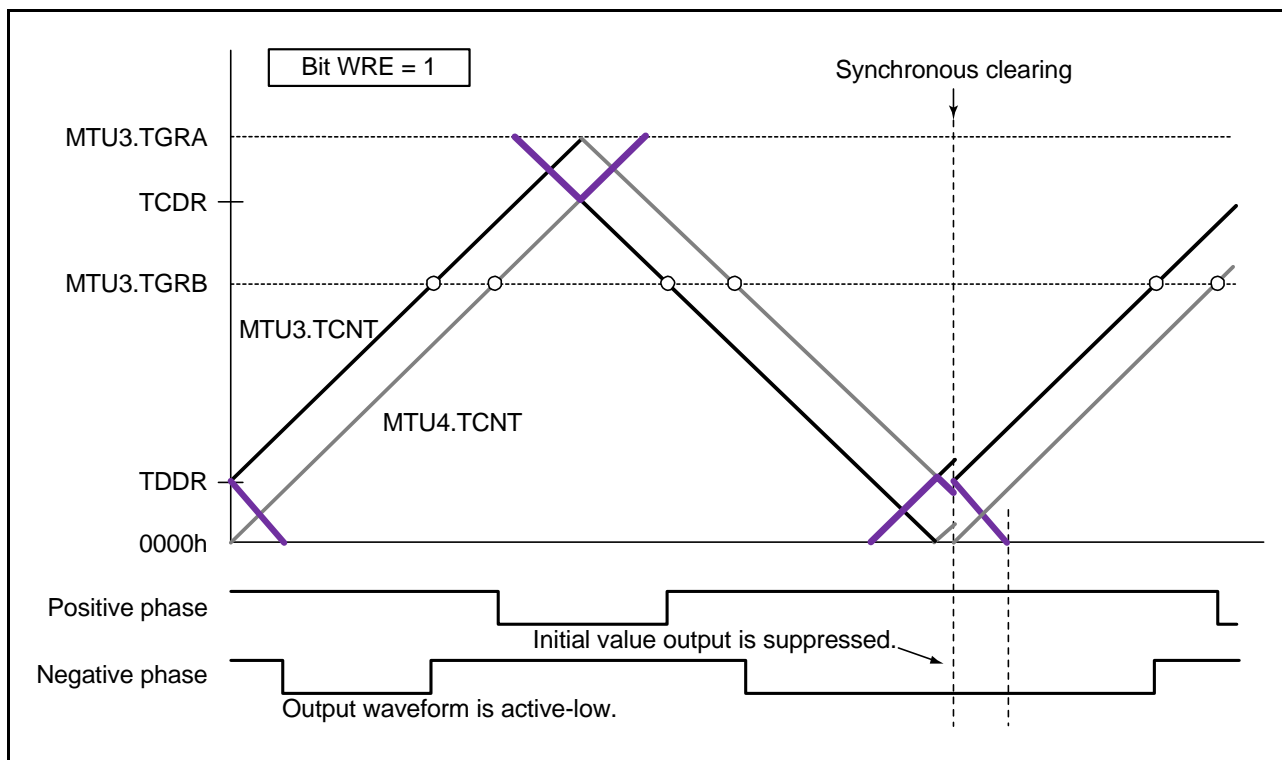


Figure 18.62 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 18.57; Bit WRE of TWCR is 1) (Unit 0)

(o) Counter Clearing by MTUn.TGRA Compare Match (n = 3 or 9)

In complementary PWM mode, MTUn.TCNT, MTUm.TCNT, and TCNTS can be cleared by MTUn.TGRA compare match when the CCE bit is set in the timer waveform control register (TWCR). (n = 3 or 9, m = 4 or 10)

Figure 18.63 illustrates an operation example.

- Note 1. Use this function only in complementary PWM mode 1 (transfer at crest).
- Note 2. Do not specify synchronous clearing by another channel (do not set the SYNC0 to SYNC4 bits in the timer synchronous register (TSYR) to 1 or the CE0A to CE0D and CE1A to CE1D bits in the timer synchronous register (TSYR) to 1).
- Note 3. Do not set the PWM duty ratio to 0000h.
- Note 4. Do not set the PSYE bit in timer output control register 1 (TOCR1) to 1.

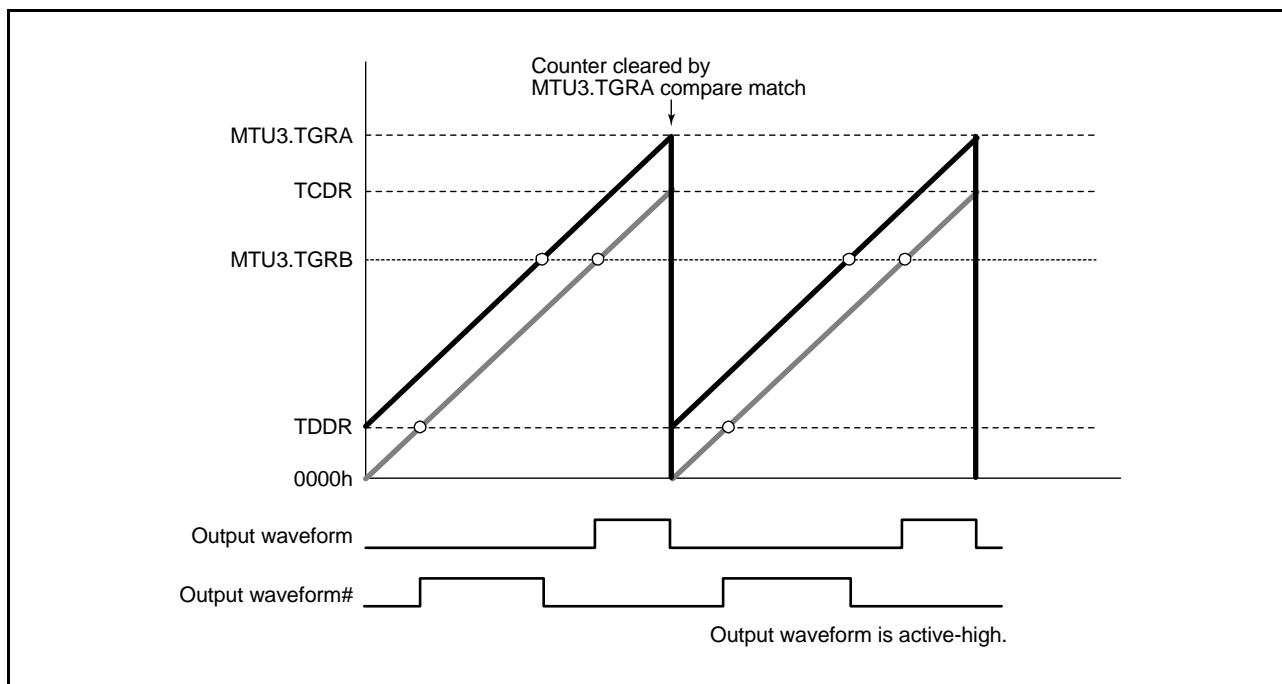


Figure 18.63 Example of Counter Clearing Operation by MTU3.TGRA Compare Match (Unit 0)

(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 18.64 to Figure 18.67 show examples of brushless DC motor driving waveforms created using TGCR.

To switch the output phases for a 3-phase brushless DC motor by means of external signals detected with a Hall element, etc., clear the FB bit in TGCR to 0. In this case, the external signals indicating the magnetic pole position should be input to timer input pins MTIOC0A, MTIOC0B, and MTIOC0C in channel 0 (set with PFC). When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.

When the FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCR is cleared to 0 or set to 1. The driving waveforms are output from the 6-phase output pins for complementary PWM mode.

With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the N bit or P bit to 1. When the N bit or P bit is 0, the level output is selected.

The active level of the 6-phase output (on output level) can be set with the OLSN and OLSP bits in the timer output control register (TOCR) regardless of the setting of the N and P bits.

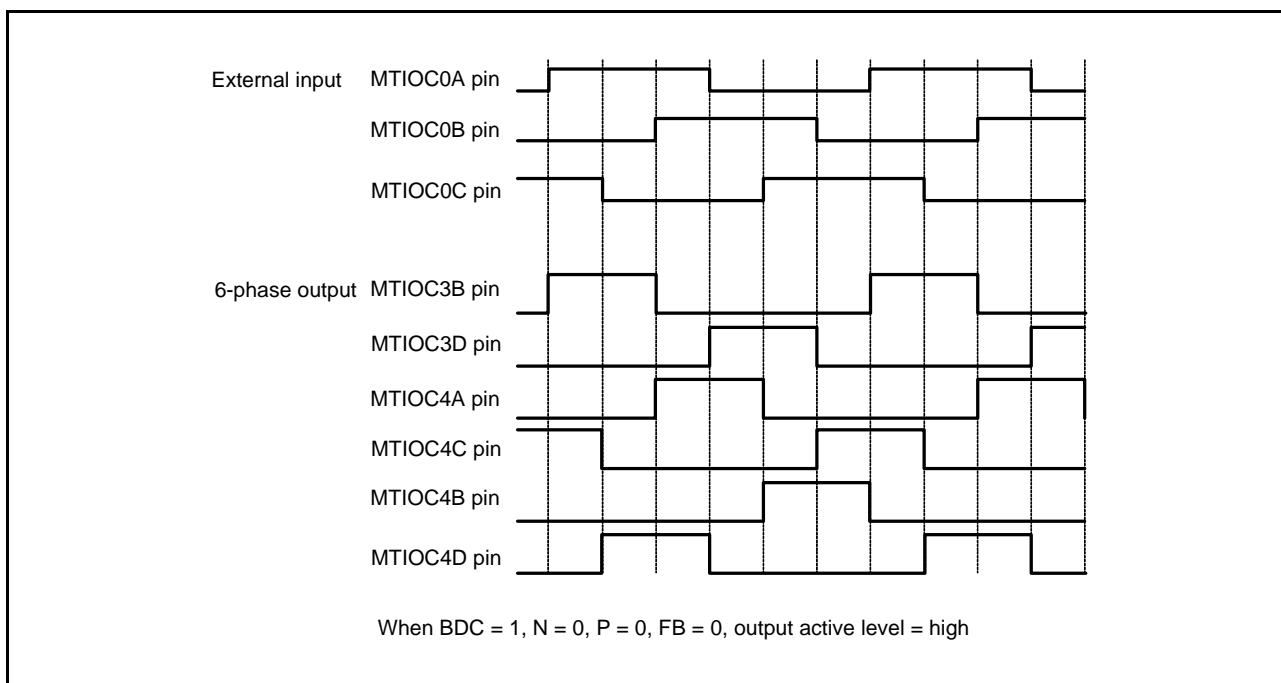


Figure 18.64 Example of Output Phase Switching by External Input (1) (Unit 0)

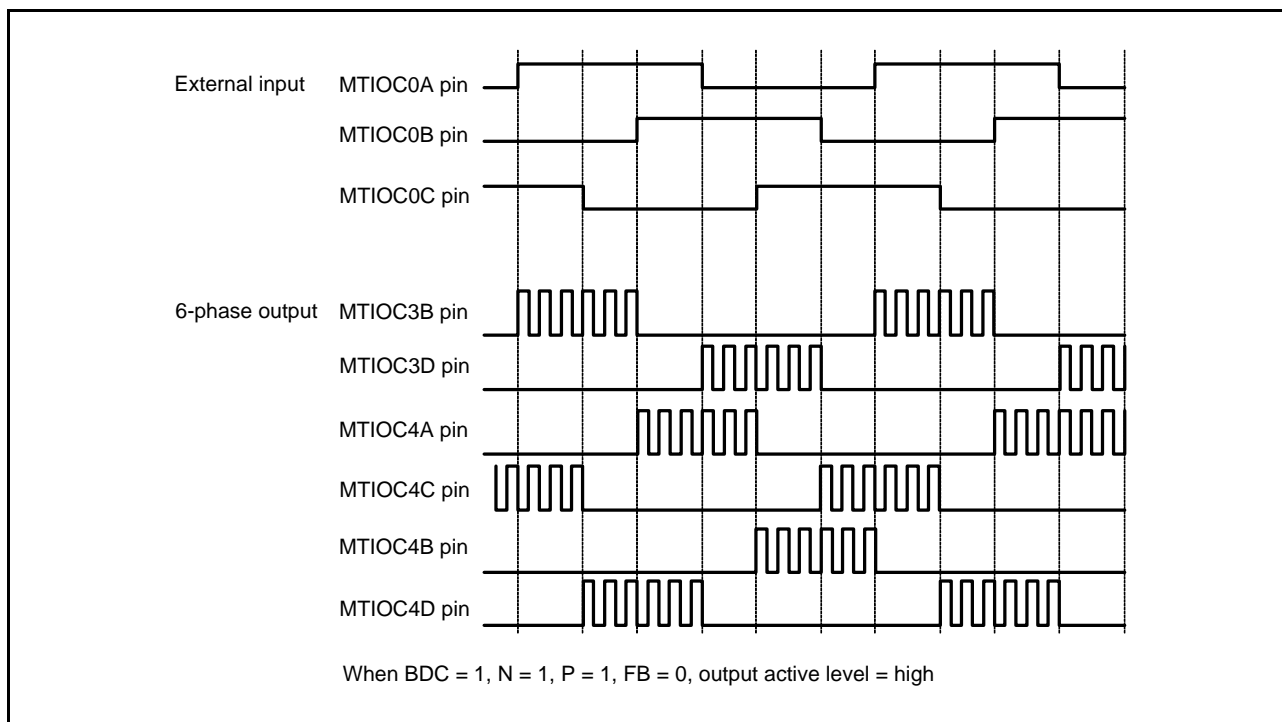


Figure 18.65 Example of Output Phase Switching by External Input (2) (Unit 0)

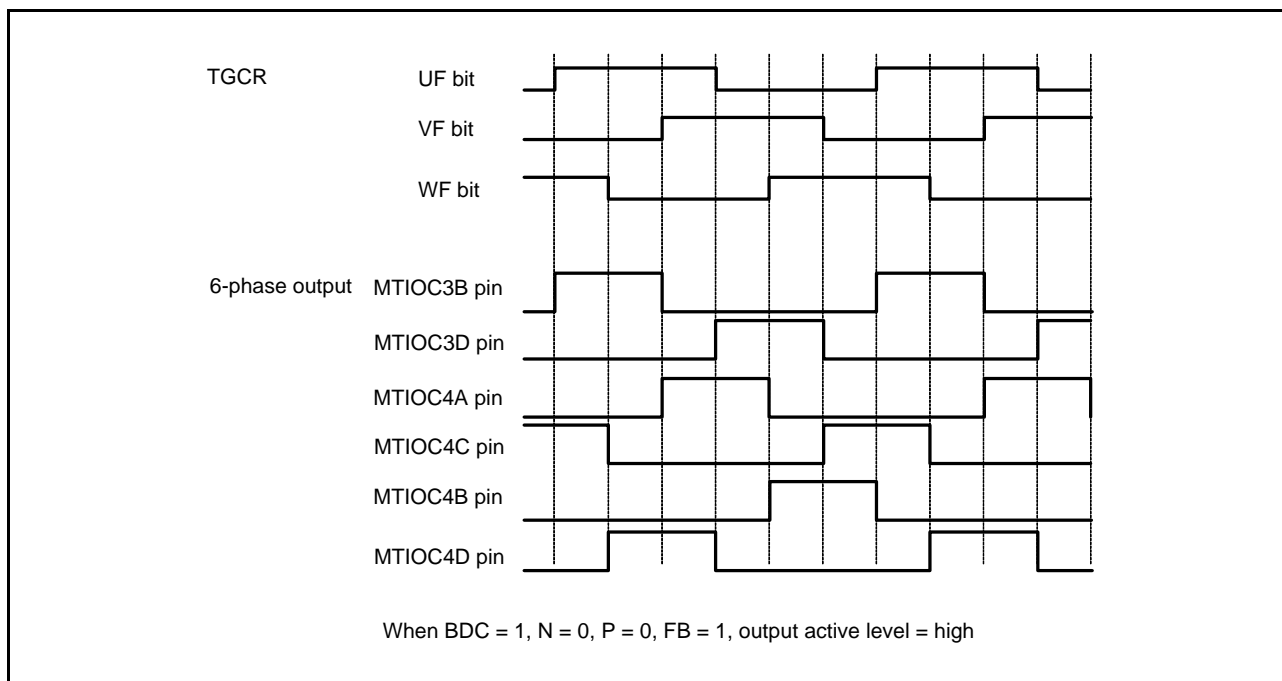


Figure 18.66 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1) (Unit 0)

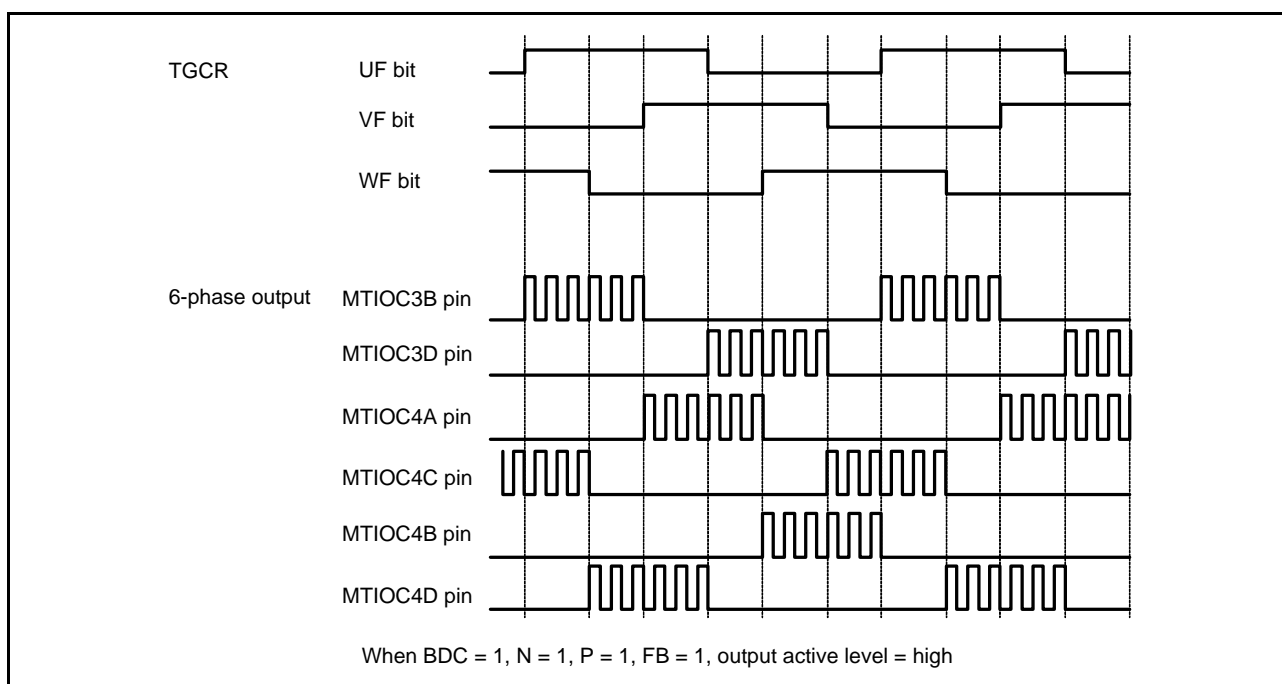


Figure 18.67 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2) (Unit 0)

(q) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using MTUn.TGRA compare match, MTUm.TCNT underflow (trough), or compare match on a channel other than channels 3, 4, 9, and 10.

When start requests using MTUn.TGRA compare match are specified, A/D conversion can be started at the crest of the MTUn.TCNT count.

A/D converter start requests can be specified by setting the TTGE bit to 1 in the timer interrupt enable register (TIER). To issue an A/D converter start request at an MTUm.TCNT underflow (trough), set the TTGE2 bit in MTUm.TIER to 1. (n = 3 or 9, m = 4 or 10)

(3) Interrupt Skipping in Complementary PWM Mode

Interrupts TGIA_n (at the crest) and TCIV_m (at the trough) in channel 3 or 9 and channel 4 or 10 can be skipped up to seven times by making settings in the timer interrupt skipping set register (TITCR). (n = 3 or 9, m = 4 or 10)

Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the timer buffer transfer register (TBTER). For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, below.

A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the timer A/D converter request control register (TADCR). For the linkage with the A/D converter start request delaying function, refer to section 18.3.9, A/D Converter Start Request Delaying Function.

The timer interrupt skipping setting register (TITCR) should be set while the TGIA_n and TCIV_m interrupt requests are disabled by the settings of MTU_n.TIER and MTU_m.TIER under the conditions in which TGIA₃ and TGIA₄ interrupt requests by compare match are never generated. Before changing the skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter. (n = 3 or 9, m = 4 or 10)

(a) Example of Interrupt Skipping Operation Setting Procedure

Figure 18.68 shows an example of the interrupt skipping operation setting procedure. Figure 18.69 shows the periods during which interrupt skipping count can be changed.

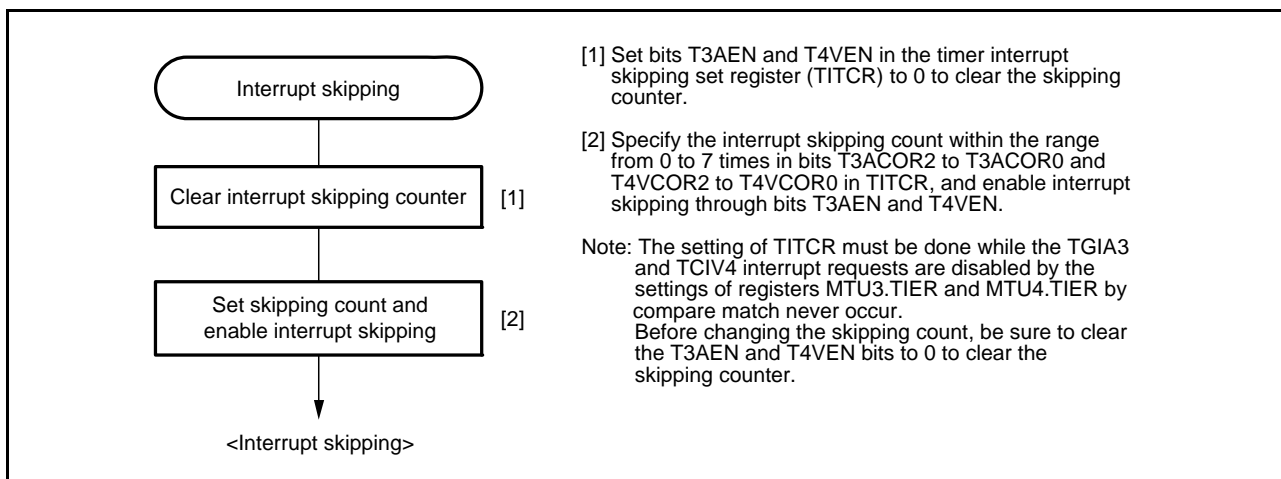


Figure 18.68 Example of Interrupt Skipping Operation Setting Procedure

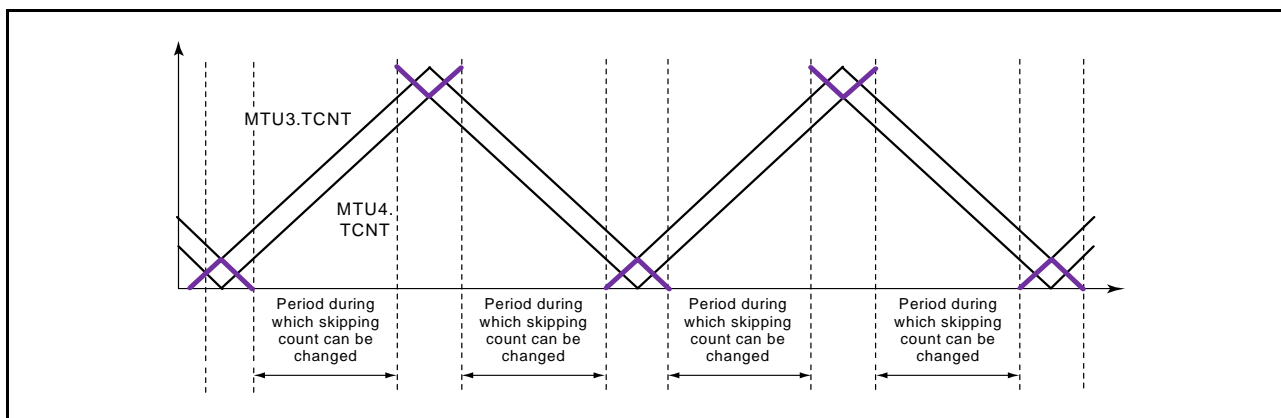


Figure 18.69 Periods during which Interrupt Skipping Count can be Changed (Unit 0)

(b) Example of Interrupt Skipping Operation

Figure 18.70 shows an example of MTU3.TGIA interrupt skipping in which the interrupt skipping count is set to three by the T3ACOR bits and the T3AEN bit is set to 1 in the timer interrupt skipping set register (TITCR).

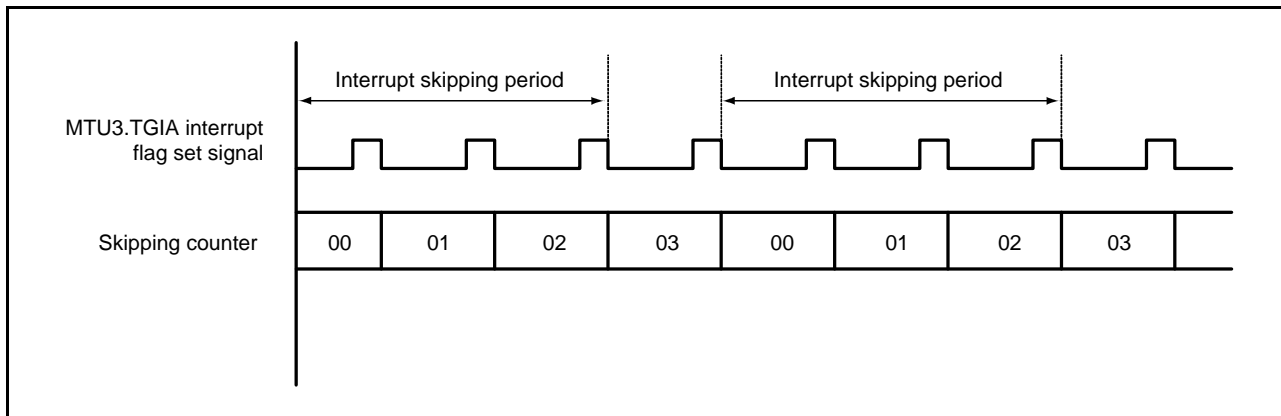


Figure 18.70 Example of Interrupt Skipping Operation (Unit 0)

(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 BTE1 and BTE0 bits in the timer buffer transfer set register (TBTER).

Figure 18.71 shows an example of operation when buffer transfer is disabled (BTE1 = 1 and BTE0 = 0). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 18.72 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE1 = 1 and BTE0 = 0). While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period.

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 18.73 shows the relationship between the T3AEN and T4VEN bit settings in TITCR and buffer transfer-enabled period.

Note: This function must always be used in combination with interrupt skipping.

When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0).

If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.

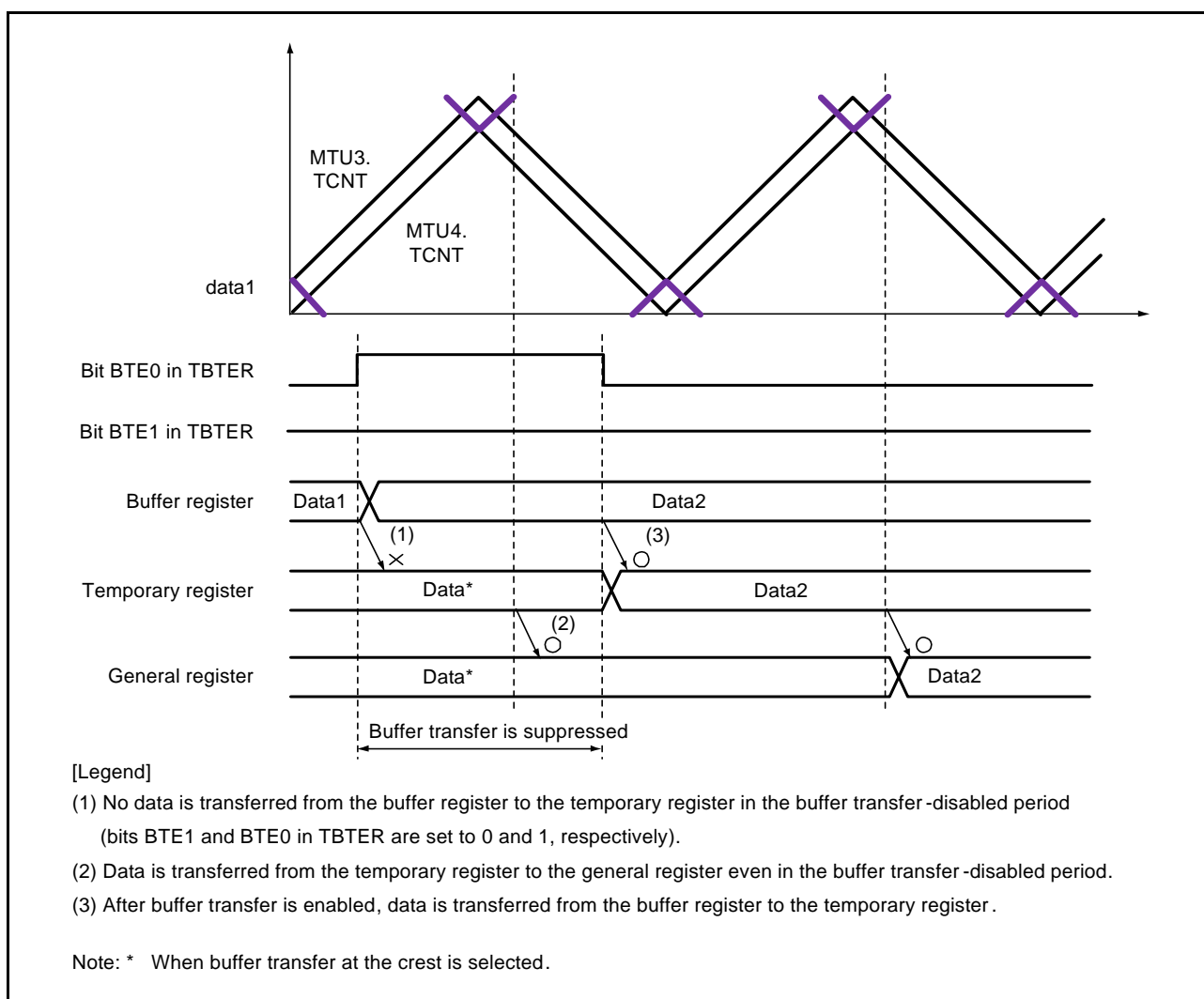


Figure 18.71 Example of Operation when Buffer Transfer is Disabled (BTE1 = 0 and BTE0 = 1) (Unit 0)

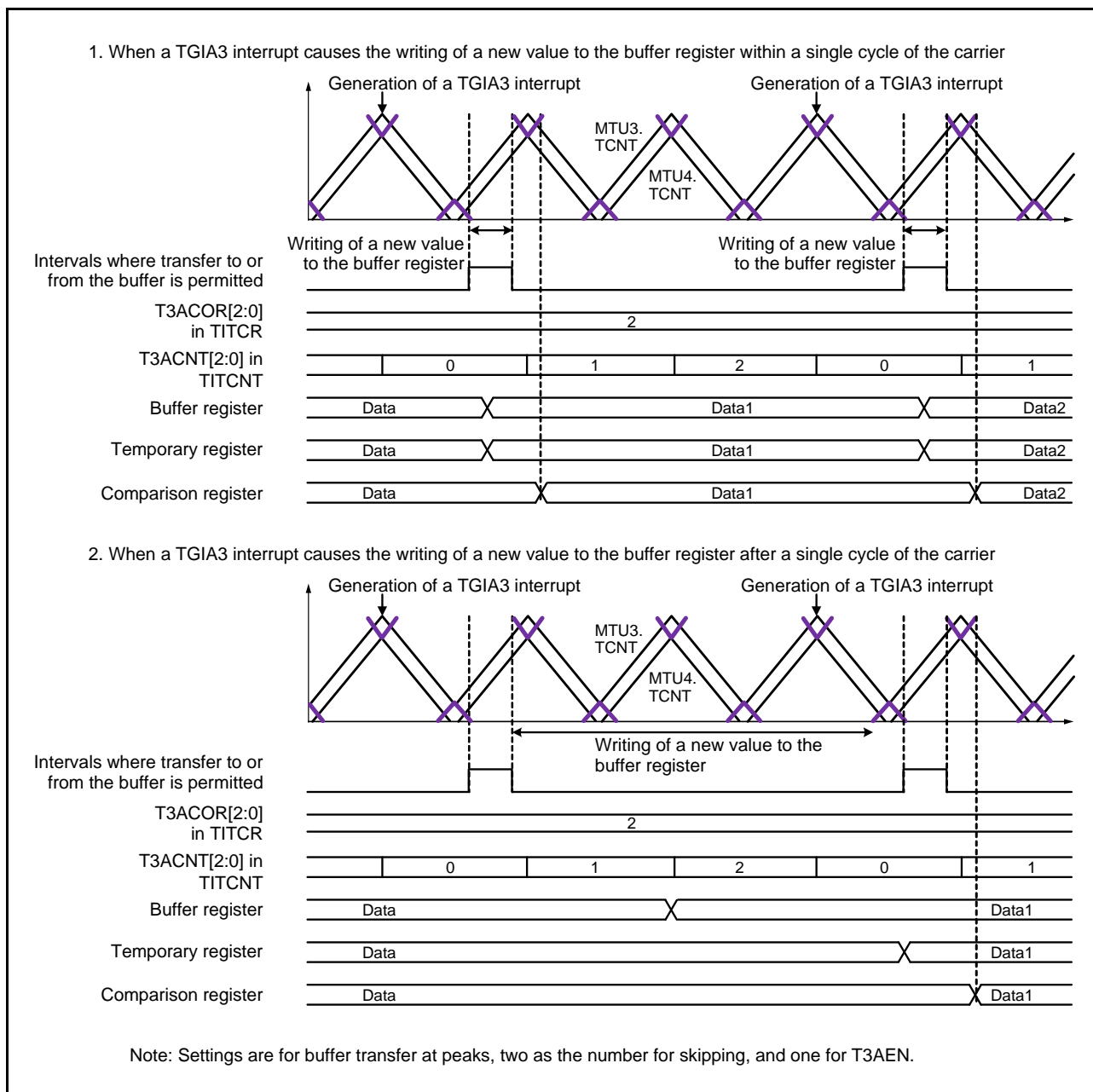


Figure 18.72 Example of Operation (Unit 0) in Buffer Transfer when Interrupt Skipping and Coupled Operation (BTE1 = 1 and BTE0 = 0) Have been Set

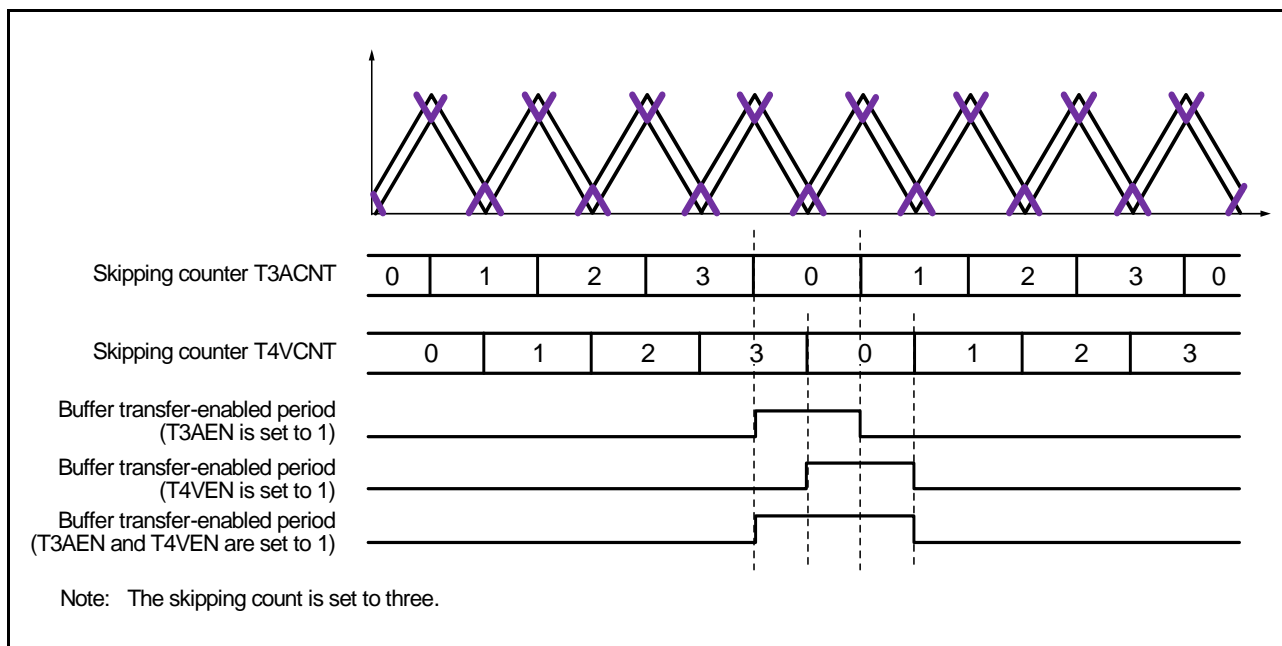


Figure 18.73 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period (Unit 0)

(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

With the exception of the buffer registers, which can be modified at any time, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of the RWE bit in the timer read/write enable register (TRWER). The applicable registers are some of the registers in channels 3 and 4 in unit 0 and channels 9 and 10 in unit 1 shown below:

Unit 0: 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, MTUA.TOER, MTUA.TOCR1, MTUA.TOCR2, MTUA.TGCR, MTUA.TCDR, and MTUA.TDDR

Unit 1: 22 registers in total

MTU9.TCR and MTU10.TCR, MTU9.TMDR and MTU10.TMDR, MTU9.TIORH and MTU10.TIORH, MTU9.TIORL and MTU10.TIORL, MTU9.TIER and MTU10.TIER, MTU9.TCNT and MTU10.TCNT, MTU9.TGRA and MTU10.TGRA, MTU9.TGRB and MTU10.TGRB, MTUB.TOER, MTUB.TOCR1, MTUB.TOCR2, MTUB.TGCR, MTUB.TCDR, and MTUB.TDDR

This function can disable CPU access to the mode registers, control registers, and counters to prevent miswriting due to CPU runaway. In the access-disabled state, the applicable registers are read as undefined and writing to these registers is ignored.

(b) Halting of PWM Output by External Signal

The 6-phase PWM output pins can be set to the high-impedance state automatically by inputting specified external signals.

See section 19, Port Output Enable 2 (POE2), for details.

(c) Halting of PWM Output when Oscillator is Stopped

Upon detecting that the clock input to this LSI has stopped, the 6-phase PWM output pins are automatically set to the high-impedance state. Note that the pin states are not guaranteed when the clock is restarted.

See section 8.11, Oscillation Stop Detection Function.

18.3.9 A/D Converter Start Request Delaying Function

A/D converter start requests can be issued in channel 4 or 10 by making settings in the timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (MTUn.TADCORA and MTUn.TADCORB), and timer A/D converter start request cycle set buffer registers (MTUn.TADCOBRA and MTUn.TADCOBRB). (n = 4 or 10)

The A/D converter start request delaying function compares MTUn.TCNT with MTUn.TADCORA or MTUn.TADCORB, and when their values match, the function issues a respective A/D converter start request (TRGnAN or TRGnBN).

A/D converter start requests (TRGnAN and TRGnBN) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITAnVE, ITBmAE, and ITBnVE bits in TADCR. (n = 4 or 10, m=3,7)

(1) Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 18.74 shows an example of procedure for specifying the A/D converter start request delaying function.

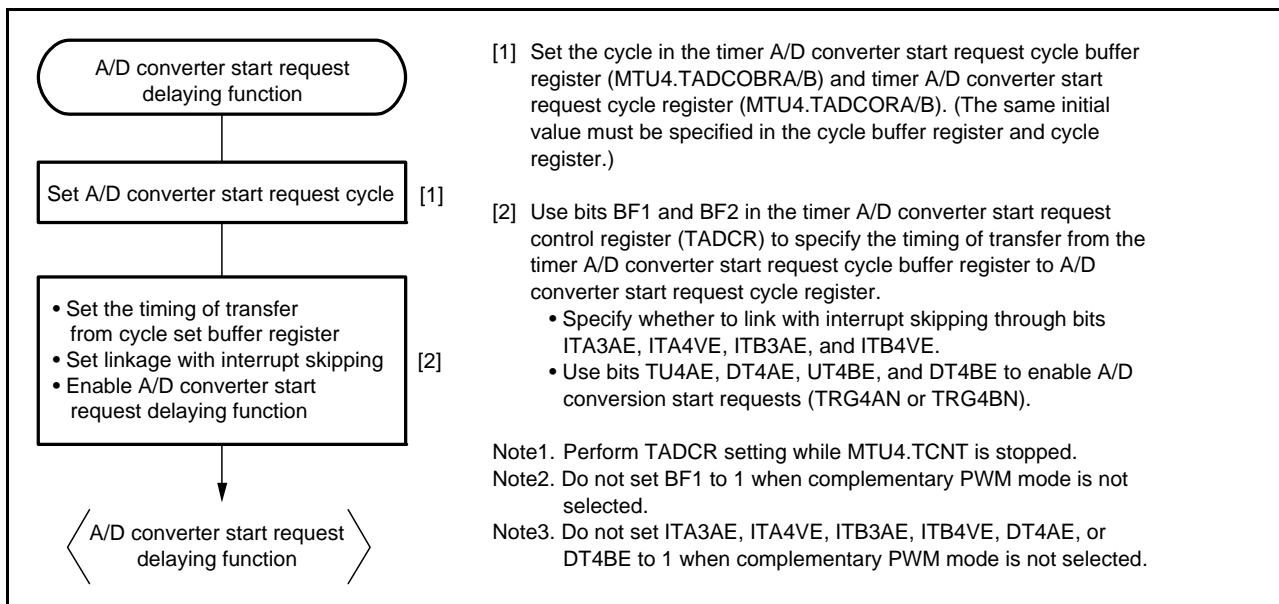


Figure 18.74 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

(2) Basic Example of A/D Converter Start Request Delaying Function Operation

Figure 18.75 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.

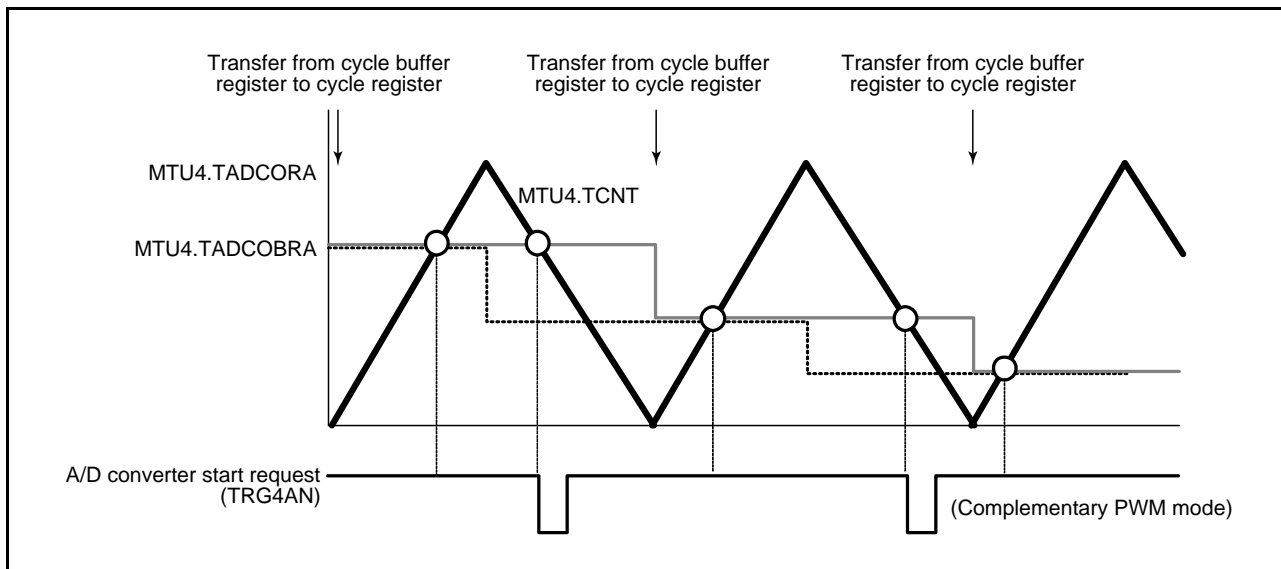


Figure 18.75 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation (Unit 0)

(3) Buffer Transfer

The data in the timer A/D converter start request cycle set registers (MTUn.TADCORA and MTUn.TADCORB) is updated by writing data to the timer A/D converter start request cycle set buffer registers (MTUn.TADCOBRA and MTUn.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF1 and BF0 bits in the timer A/D converter start request control register (MTUn.TADCR). (n = 4 or 10)

(4) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR).

Figure 18.76 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTUn.TCNT up-counting and down-counting and A/D converter start requests are linked with interrupt skipping.

Figure 18.77 shows another example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTUn.TCNT up-counting and A/D converter start requests are linked with interrupt skipping. (n = 4 or 10)

Note: This function should be used in combination with interrupt skipping.

When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), make sure that A/D converter start requests are not linked with interrupt skipping (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).

Note that TRG4ABN (TRG4AN or TRG4BN) is output as the A/D converter start request signal in this case.

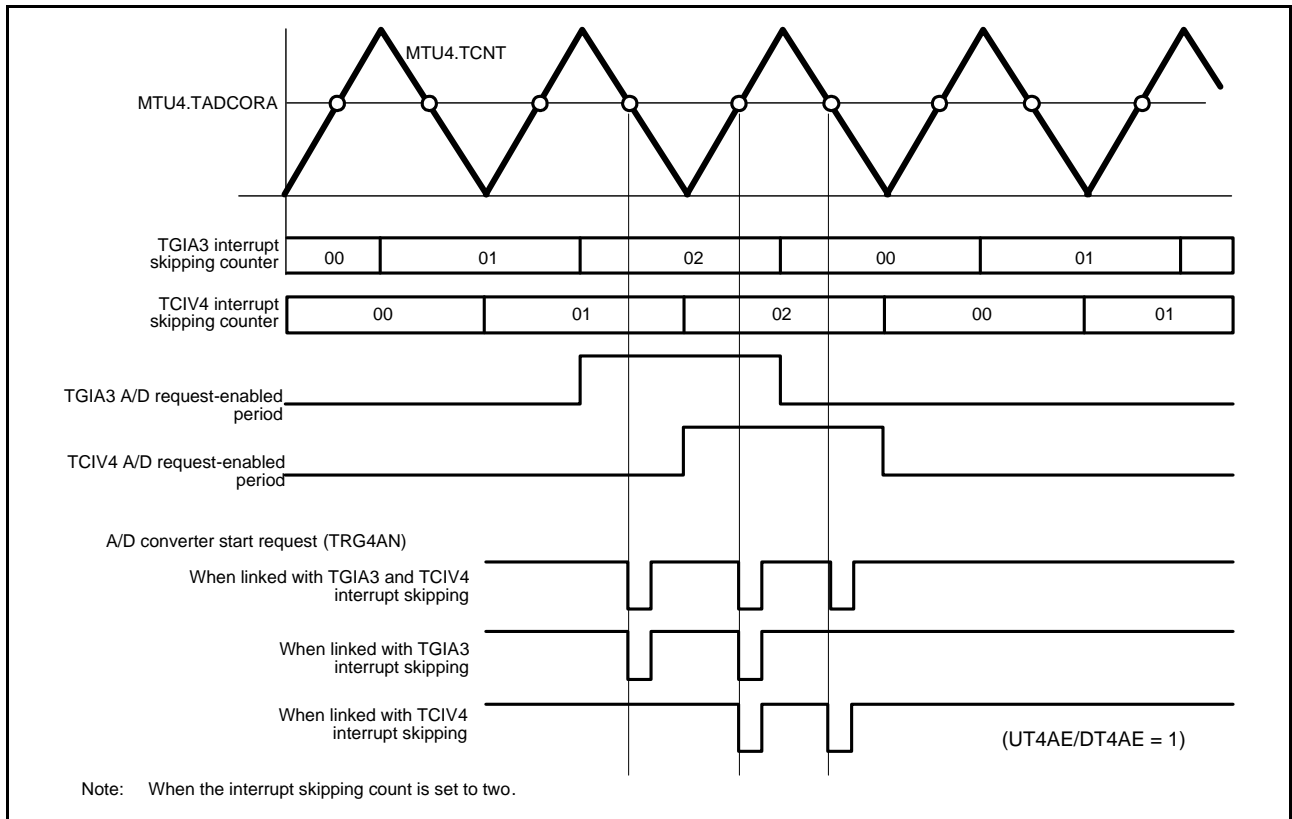


Figure 18.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (Unit 0) (when TRG4AN Output is Enabled during TCNT Up-Counting and Down-Counting)

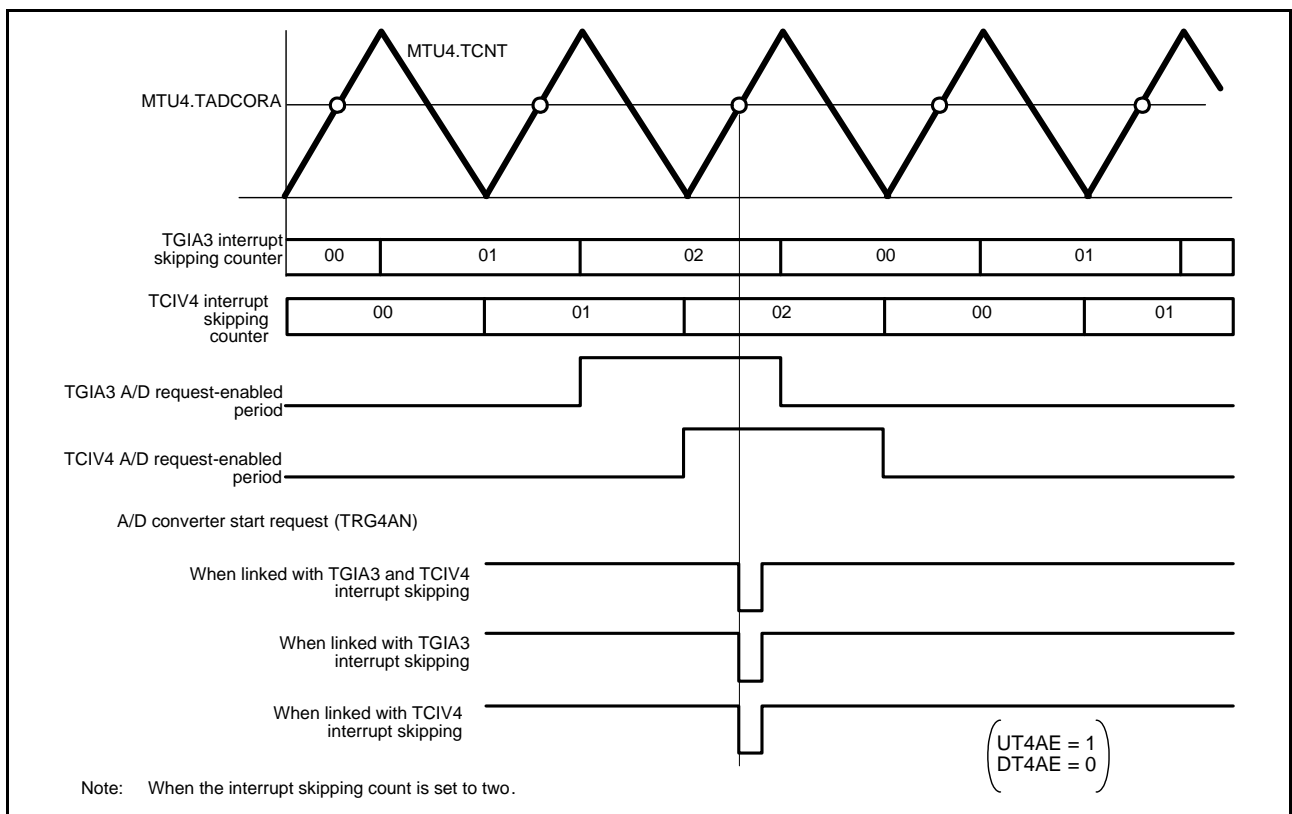


Figure 18.77 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (Unit 0) (when TRG4AN Output is Enabled during TCNT Up-Counting)

18.3.10 External Pulse Width Measurement

The pulse widths of up to three external input lines can be measured in channel 5 (or channel 11).

(1) Example of External Pulse Width Measurement Setting Procedure

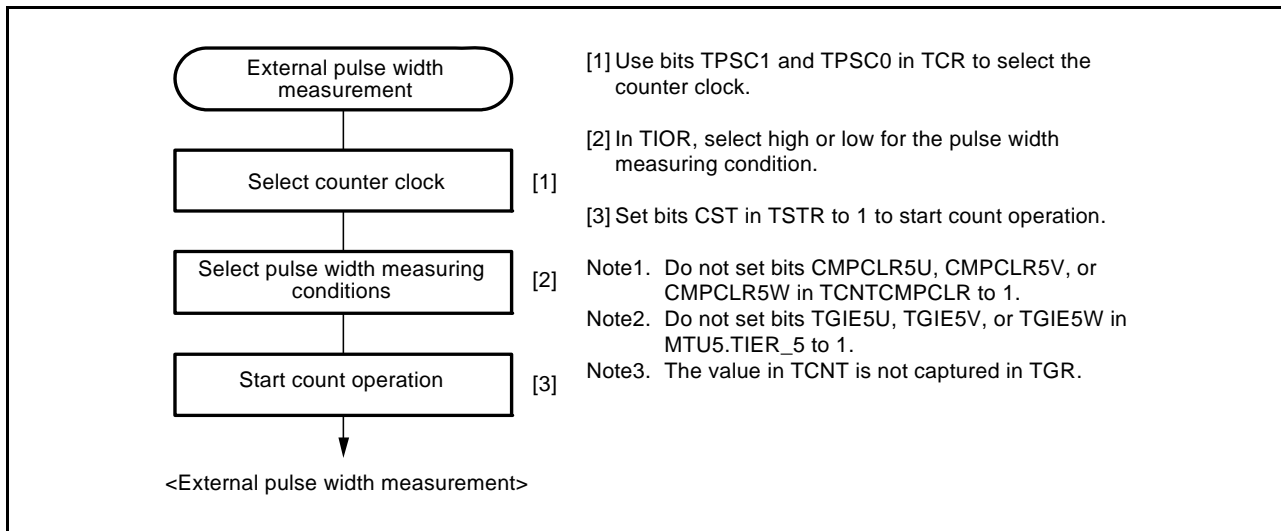


Figure 18.78 Example of External Pulse Width Measurement Setting Procedure (Unit 0)

(2) Example of External Pulse Width Measurement

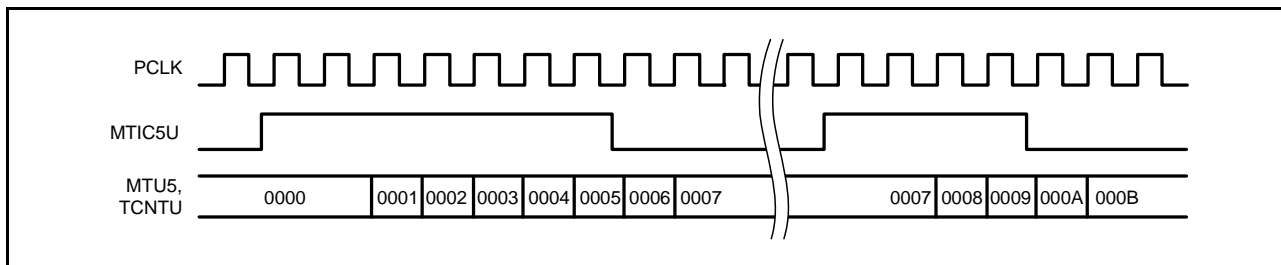


Figure 18.79 Example of External Pulse Width Measurement (Measuring High Pulse Width)

18.3.11 Dead Time Compensation

By measuring the delay of the output waveform and reflecting it to duty ratio, the external pulse width measurement function can be used as the dead time compensation function while the complementary PWM is in operation.

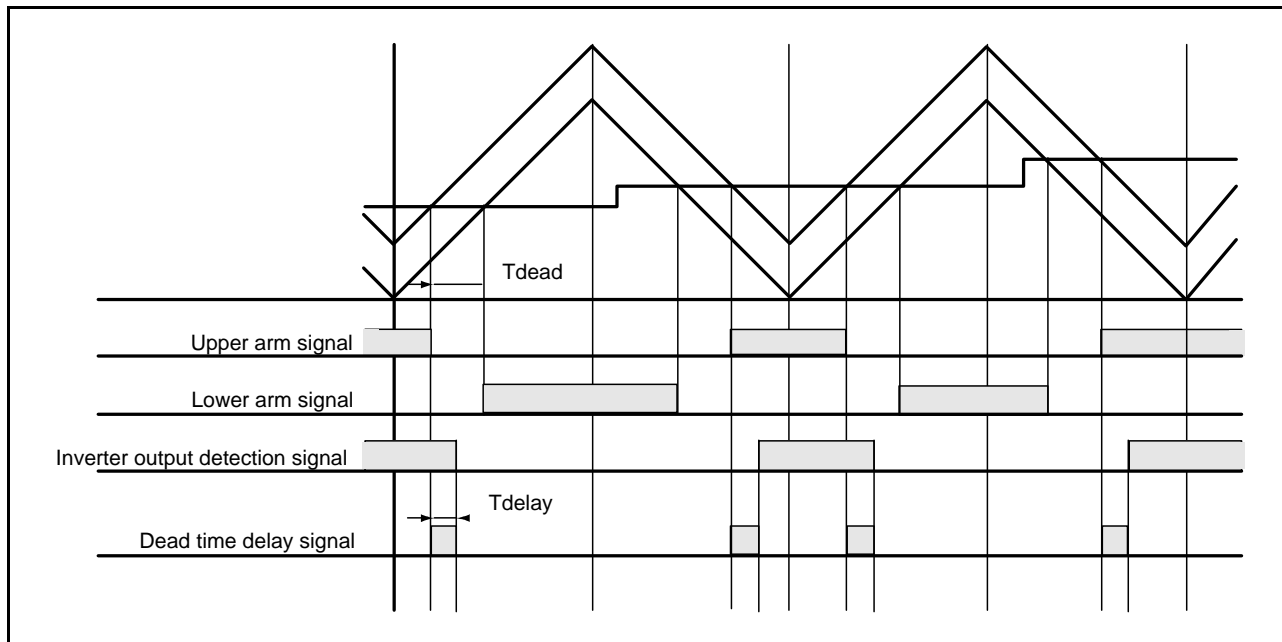


Figure 18.80 Delay in Dead Time in Complementary PWM Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 18.81 shows an example of dead time compensation setting procedure by using three counters in channel 5 (or channel 11).

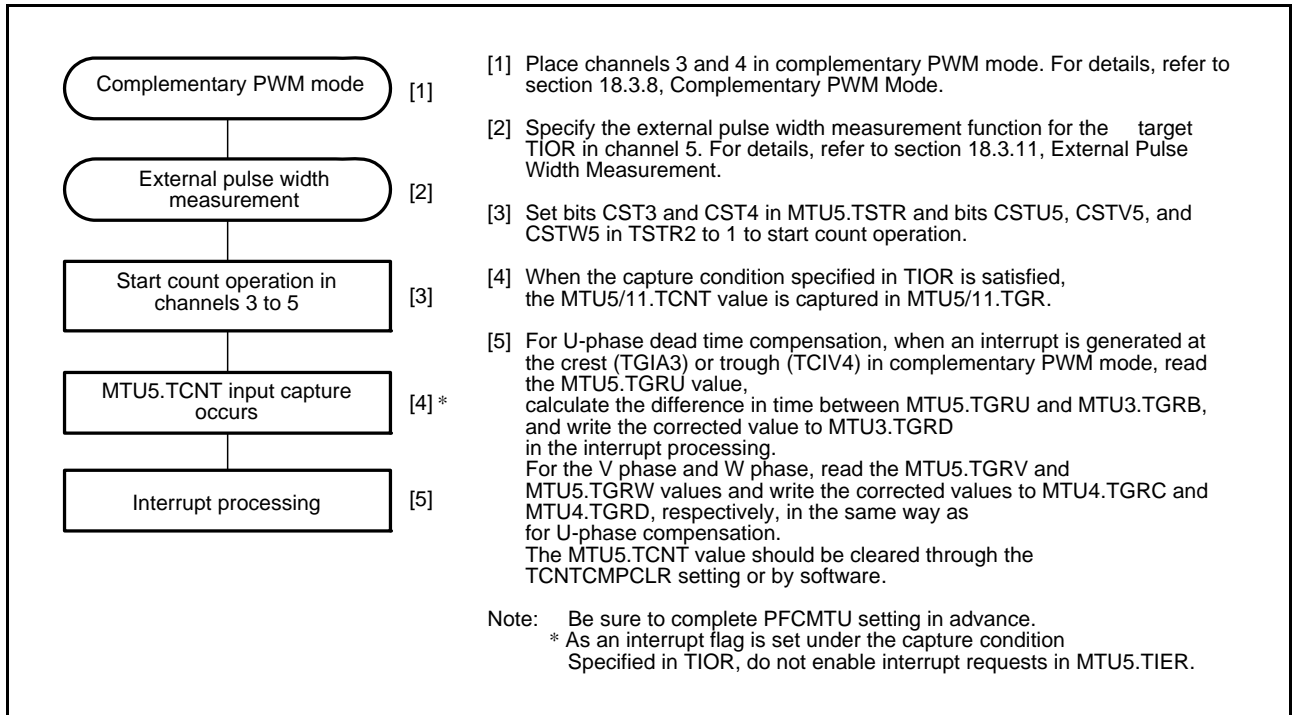


Figure 18.81 Example of Dead Time Compensation Setting Procedure (Unit 0)

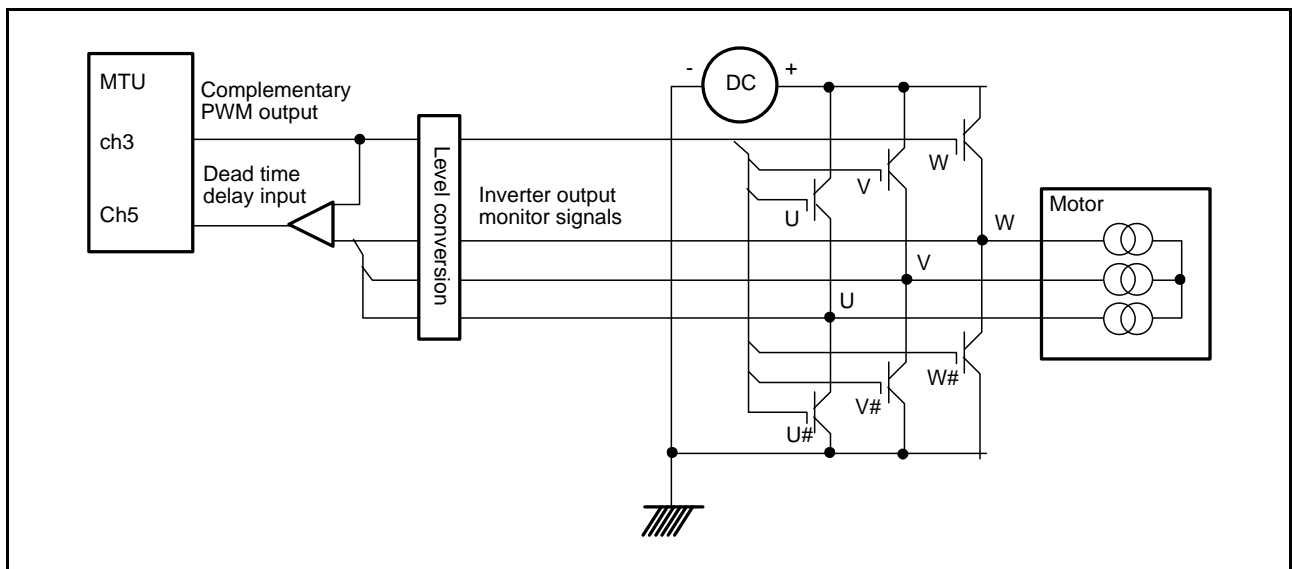


Figure 18.82 Example of Motor Control Circuit Configuration

18.3.12 TCNT Capture at Crest and/or Trough in Complementary PWM Operation

The TCNT value is captured in TGR at either the crest or trough or at both the crest and trough during complementary PWM operation. The timing for capturing in TGR can be selected by TIOR.

Figure 18.83 is an operating example in which TCNT is used as a free-running counter without being cleared, and the TCNT value is captured in TGR at the specified timing (either crest or trough, or both crest and trough).

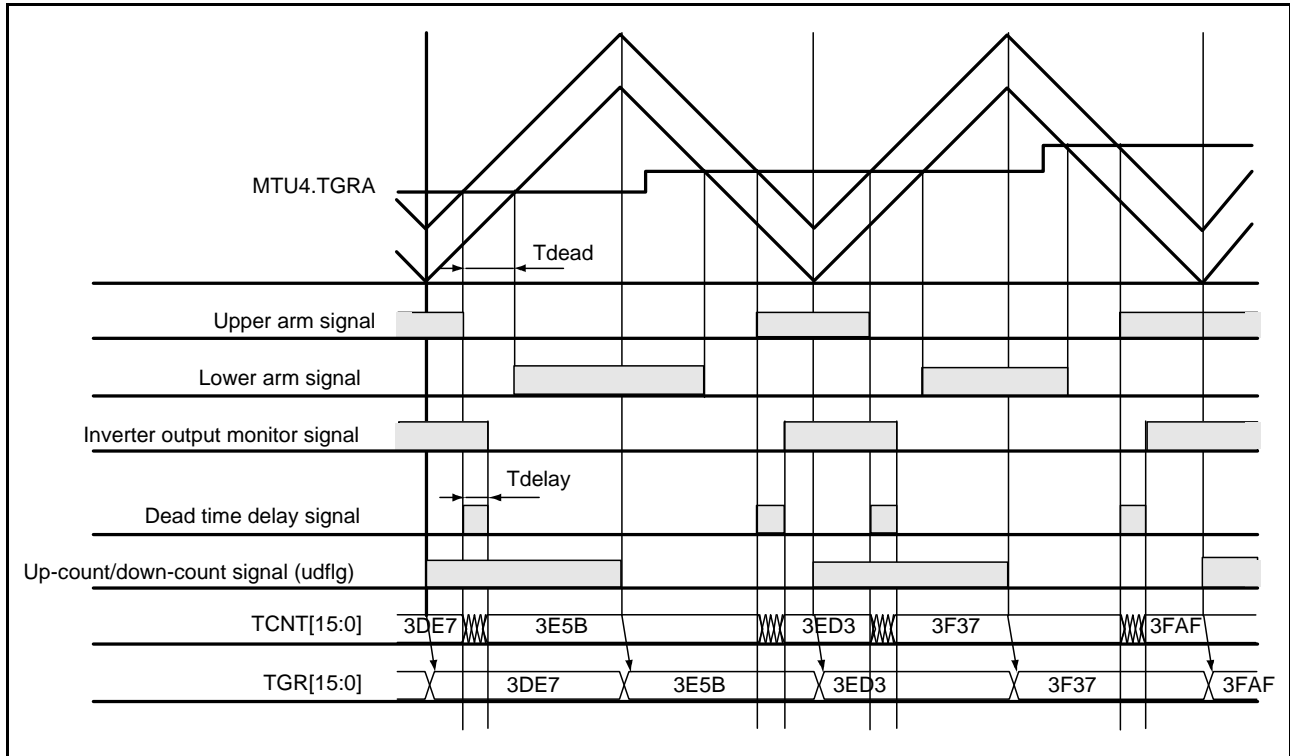


Figure 18.83 TCNT Capture at Crest and/or Trough in Complementary PWM Operation

18.4 Interrupt Sources

18.4.1 Interrupt Sources and Priorities

There are three kinds of MTU interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and 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, see section 11, Interrupt Control Unit (ICUa).

Table 18.59 and Table 18.60 list the MTU interrupt sources.

Table 18.59 MTU Interrupt Sources (1) (Unit 0)

Channel	Name	Interrupt Source	DMACA Activation	DTC Activation	Priority
0	TGIA0	MTU0.TGRA input capture/compare match	Possible	Possible	High ↑ Low
	TGIB0	MTU0.TGRB input capture/compare match	Not possible	Possible	
	TGIC0	MTU0.TGRC input capture/compare match	Not possible	Possible	
	TGID0	MTU0.TGRD input capture/compare match	Not possible	Possible	
	TCIV0	MTU0.TCNT overflow	Not possible	Not possible	
	TGIE0	MTU0.TGRE compare match	Not possible	Not possible	
	TGIF0	MTU0.TGRF compare match	Not possible	Not possible	
1	TGIA1	MTU1.TGRA input capture/compare match	Possible	Possible	
	TGIB1	MTU1.TGRB input capture/compare match	Not possible	Possible	
	TCIV1	MTU1.TCNT overflow	Not possible	Not possible	
	TCIU1	MTU1.TCNT underflow	Not possible	Not possible	
2	TGIA2	MTU2.TGRA input capture/compare match	Possible	Possible	
	TGIB2	MTU2.TGRB input capture/compare match	Not possible	Possible	
	TCIV2	MTU2.TCNT overflow	Not possible	Not possible	
	TCIU2	MTU2.TCNT underflow	Not possible	Not possible	
3	TGIA3	MTU3.TGRA input capture/compare match	Possible	Possible	
	TGIB3	MTU3.TGRB input capture/compare match	Not possible	Possible	
	TGIC3	MTU3.TGRC input capture/compare match	Not possible	Possible	
	TGID3	MTU3.TGRD input capture/compare match	Not possible	Possible	
	TCIV3	MTU3.TCNT overflow	Not possible	Not possible	
4	TGIA4	MTU4.TGRA input capture/compare match	Possible	Possible	
	TGIB4	MTU4.TGRB input capture/compare match	Not possible	Possible	
	TGIC4	MTU4.TGRC input capture/compare match	Not possible	Possible	
	TGID4	MTU4.TGRD input capture/compare match	Not possible	Possible	
	TCIV4	MTU4.TCNT overflow/underflow	Not possible	Possible	
5	TGIU5	MTU5.TGRU input capture/compare match	Not possible	Possible	
	TGIV5	MTU5.TGRV input capture/compare match	Not possible	Possible	
	TGIW5	MTU5.TGRW input capture/compare match	Not possible	Possible	

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Table 18.60 MTU Interrupt Sources (2) (Unit 1)

Channel	Name	Interrupt Source	DMACA Activation	DTC Activation	Priority		
6	TGIA6	MTU6.TGRA input capture/compare match	Possible	Possible	High ↑ Low		
	TGIB6	MTU6.TGRB input capture/compare match	Not possible	Possible			
	TGIC6	MTU6.TGRC input capture/compare match	Not possible	Possible			
	TGID6	MTU6.TGRD input capture/compare match	Not possible	Possible			
	TCIV6	MTU6.TCNT overflow	Not possible	Not possible			
	TGIE6	MTU6.TGRE compare match	Not possible	Not possible			
7	TGIA7	MTU7.TGRA input capture/compare match	Possible	Possible	High ↑ Low		
	TGIB7	MTU7.TGRB input capture/compare match	Not possible	Possible			
	TCIV7	MTU7.TCNT overflow	Not possible	Not possible			
	TCIU7	MTU7.TCNT underflow	Not possible	Not possible			
8	TGIA8	MTU8.TGRA input capture/compare match	Possible	Possible		High ↑ Low	
	TGIB8	MTU8.TGRB input capture/compare match	Not possible	Possible			
	TCIV8	MTU8.TCNT overflow	Not possible	Not possible			
	TCIU8	MTU8.TCNT underflow	Not possible	Not possible			
9	TGIA9	MTU9.TGRA input capture/compare match	Possible	Possible			High ↑ Low
	TGIB9	MTU9.TGRB input capture/compare match	Not possible	Possible			
	TGIC9	MTU9.TGRC input capture/compare match	Not possible	Possible			
	TGID9	MTU9.TGRD input capture/compare match	Not possible	Possible			
	TCIV9	MTU9.TCNT overflow	Not possible	Not possible			
10	TGIA10	MTU10.TGRA input capture/compare match	Possible	Possible	High ↑ Low		
	TGIB10	MTU10.TGRB input capture/compare match	Not possible	Possible			
	TGIC10	MTU10.TGRC input capture/compare match	Not possible	Possible			
	TGID10	MTU10.TGRD input capture/compare match	Not possible	Possible			
	TCIV10	MTU10.TCNT overflow/underflow	Not possible	Possible			
11	TGIU11	MTU11.TGRU input capture/compare match	Not possible	Possible		High ↑ Low	
	TGIV11	MTU11.TGRV input capture/compare match	Not possible	Possible			
	TGIW11	MTU11.TGRW input capture/compare match	Not possible	Possible			

Note: This table shows 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 TGIE bit in TIER is set to 1 when a TGR input capture/compare match occurs on a channel. The MTU has 21 input capture/compare match interrupts in unit 0 (six for channel 0, four each for channels 3 and 4, two each for channels 1 and 2, and three for channel 5) and 21 input capture/compare match interrupts in unit 1 (six for channel 6, four each for channels 9 and 10, two each for channels 7 and 8, and three for channel 11).

(2) Overflow Interrupt

An interrupt is requested if the TCIEV bit in TIER is set to 1 when a TCNT overflow occurs on a channel. The MTU has five overflow interrupts (one for each channel) in unit 0 and five overflow interrupts (one for each channel) in unit 1.

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when a TCNT underflow occurs on a channel. The MTU has two underflow interrupts (one each for channels 1 and 2) in unit 0 and two underflow interrupts (one each for channels 7 and 8) in unit 1.

18.4.2 DTC and DMACA Activation

(1) DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt in each channel or the overflow interrupt in channel 4. For details, see section 16, Data Transfer Controller (DTCa).

Unit 0 of 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 channels 0 and 3, two each for channels 1 and 2, five for channel 4, and three for channel 5. Unit 1 also provides a total of 20 input capture/compare match interrupts and overflow interrupts that can be used as DTC activation sources: four each for channels 6 and 9, two each for channels 7 and 8, five for channel 10, and three for channel 11.

(2) DMACA Activation

The DMACA can be activated by the TGRA input capture/compare match interrupt in each channel. For details, see section 14, DMA Controller (DMACA).

Unit 0 of the MTU provides a total of five TGRA input capture/compare match interrupts that can be used as DMACA activation sources: one each for channels 0 to 4. Unit 1 also provides a total of five TGRA input capture/compare match interrupts that can be used as DMACA activation sources: one each for channels 6 to 10.

When the DMACA is activated by the MTU, the activation source is cleared when the DMACA requests the internal bus mastership. Therefore, the request for DMACA transfer may be kept pending for a certain period even after the activation source is cleared depending on the internal bus state.

(3) EXDMAC Activation

The EXDMA can be activated by the TGRA input capture/compare match interrupt in each channel. For details, see section 15, EXDMA Controller (EXDMAC).

18.4.3 A/D Converter Activation

The A/D converter can be activated by one of the following five methods in the MTU. Table 18.61 shows 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 MTUn.TCNT Trough in Complementary PWM Mode (n = 4 or 10)

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 operation is performed while the TTGE2 bit in MTUn.TIER is set to 1, the A/D converter can be activated at the trough of MTUn.TCNT count (MTUn.TCNT = 0000h). (n = 4 or 10)

A/D converter start request signal TRGAN is issued to the A/D converter under either of the following conditions.

- When a TGRA input capture/compare match occurs on a channel while the TTGE bit in TIER is set to 1
- When the MTUn.TCNT count reaches the trough (MTUn.TCNT = 0000h) during complementary PWM operation while the TTGE2 bit in MTUn.TIER is set to 1 (n = 4 or 10)

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 (MTU0)

A/D converter start request signal TRG0EN is issued to the A/D converter when a compare match occurs between MTU0.TCNT and MTU0.TGRE.

While the TGFE flag in MTU0.TSR2 is set to 1 by the occurrence of a compare match between MTU0.TCNT and TGRE, A/D converter start request signal TRG0EN is issued to the A/D converter when a compare match occurs between TCNT and TGRE in channel 0. 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 TCNT0 and TGRF0 (MTU0)

The A/D converter can be activated by generating A/D converter start request signal TRG0FN when a compare match occurs between TCNT0 and TGRF0 in channel 0.

When a compare match occurs between TCNT0 and TGRF0 in channel 0, A/D converter start request signal TRG0FN is issued to the A/D converter. 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 in TGRA0 or TGRB0 (MTU0)

The A/D converter can be activated when an input capture or compare match occurs between TCNT0 and TGRA0 or TGRB0 in channel 0.

When an input capture or compare match occurs between TCNT0 and TGRA0 or TGRB0 in channel 0, A/D converter start request signal TRG0AN or TRG0BN is issued to the A/D converter. 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 (MTU4)

The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN when the MTU4.TCNT count matches the TADCORA or TADCORB value if the UT4AE, DT4AE, UT4BE, or DT4BE bit in the A/D converter start request control register (TADCR) is set to 1. For details, refer to section 18.3.9, 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 18.61 Interrupt Sources and A/D Converter Start Request Signals

Target Registers	A/D Start Request Source	A/D Converter Start Request Signal
MTUn.TGRA and MTUn.TCNT (n = 0)	Input capture/compare match	TRGAN
MTUn.TGRA and MTUn.TCNT (n = 1 or 7)		
MTUn.TGRA and MTUn.TCNT (n = 2 or 8)		
MTUn.TGRA and MTUn.TCNT (n = 3 or 9)		
MTUn.TGRA and MTUn.TCNT (n = 4 or 10)		
MTUn.TCNT (n = 4 or 10)	MTUn.TCNT trough in complementary PWM mode (n = 4 or 10)	
MTUn.TGRA and MTUn.TCNT (n = 0)	Input capture/compare match	TRGnAN (n = 0)
MTUn.TGRB and MTUn.TCNT (n = 0)		TRGnBN (n = 0)
MTUn.TGRE and MTUn.TCNT (n = 0)	Compare match	TRGnEN (n = 0)
MTUn.TGRF and MTUn.TCNT (n = 0)		TRGnFN (n = 0)
TADCORA and MTUn.TCNT or TADCORB and MTUn.TCNT (n = 4 or 10)		TRGnABN (n = 4 or 10)

18.5 Operation Timing

18.5.1 Input/Output Timing

(1) TCNT Count Timing

Figure 18.84 and Figure 18.85 show the TCNT count timing in internal clock operation, Figure 17.55 shows the TCNT count timing in external clock operation (normal mode), and Figure 18.87 shows the TCNT count timing in external clock operation (phase counting mode).

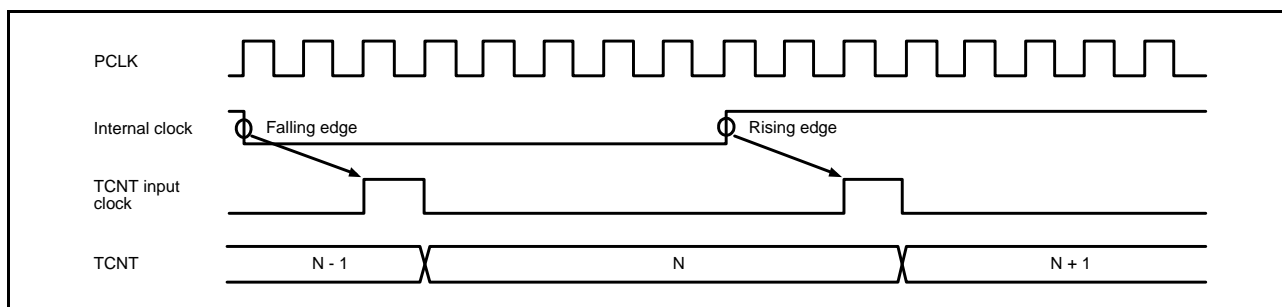


Figure 18.84 Count Timing in Internal Clock Operation (Channels 0 to 4 and 6 to 10)

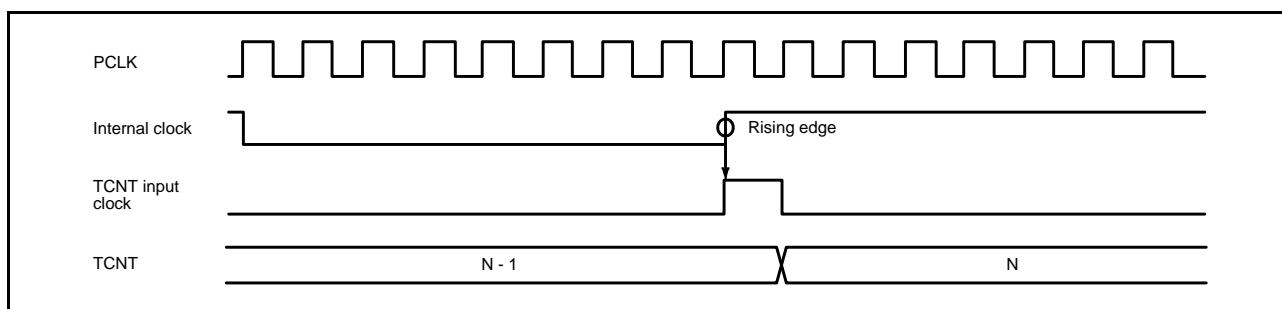


Figure 18.85 Count Timing in Internal Clock Operation (Channels 5 and 11)

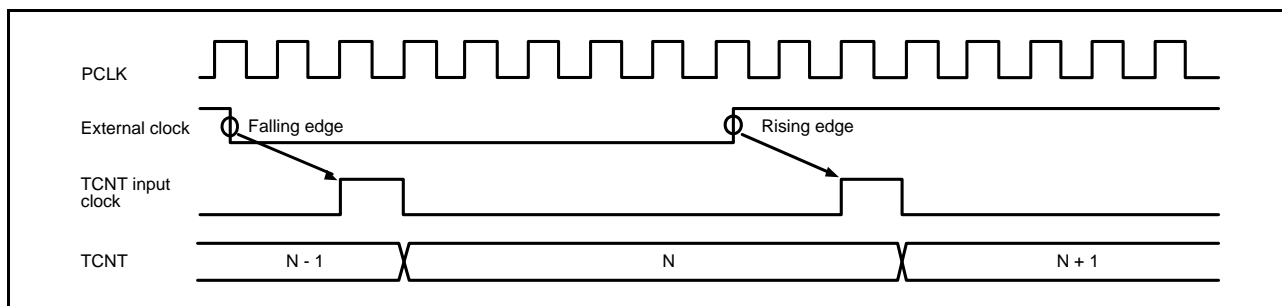


Figure 18.86 Count Timing in External Clock Operation (Channels 0 to 4 and 6 to 10)

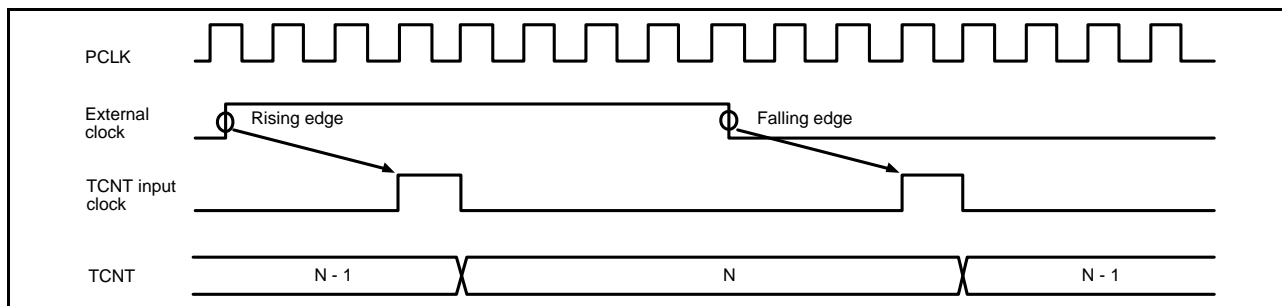


Figure 18.87 Count Timing in External Clock Operation (Phase Counting Mode)

(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 by TCNT is updated). When a compare match signal is generated, the value set in TIOR is output from 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 18.88 shows the output compare output timing (normal mode or PWM mode) and Figure 18.89 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

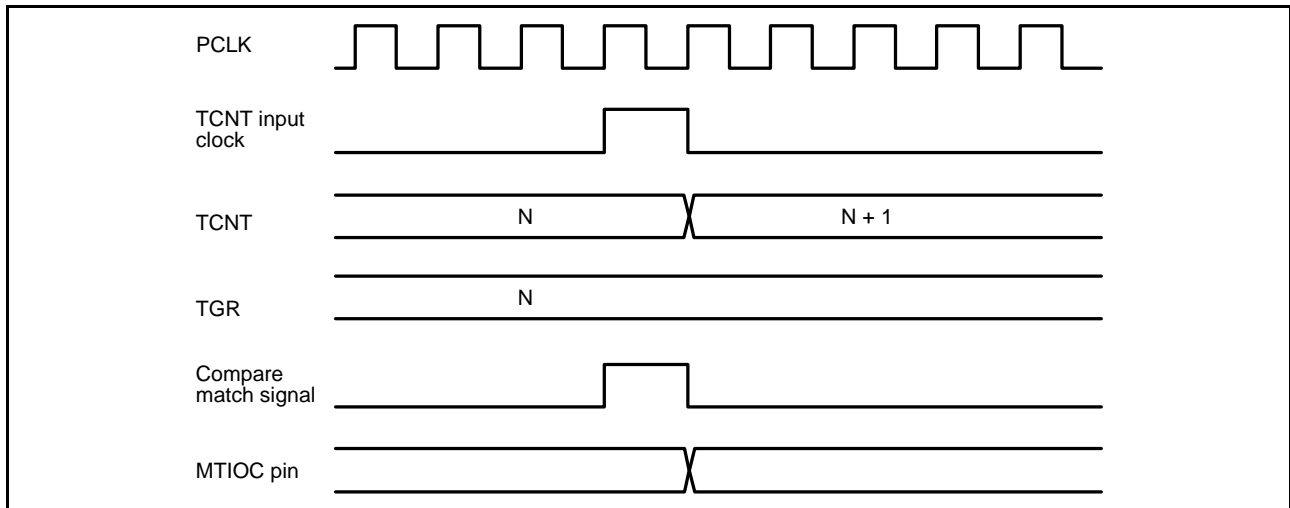


Figure 18.88 Output Compare Output Timing (Normal Mode or PWM Mode)

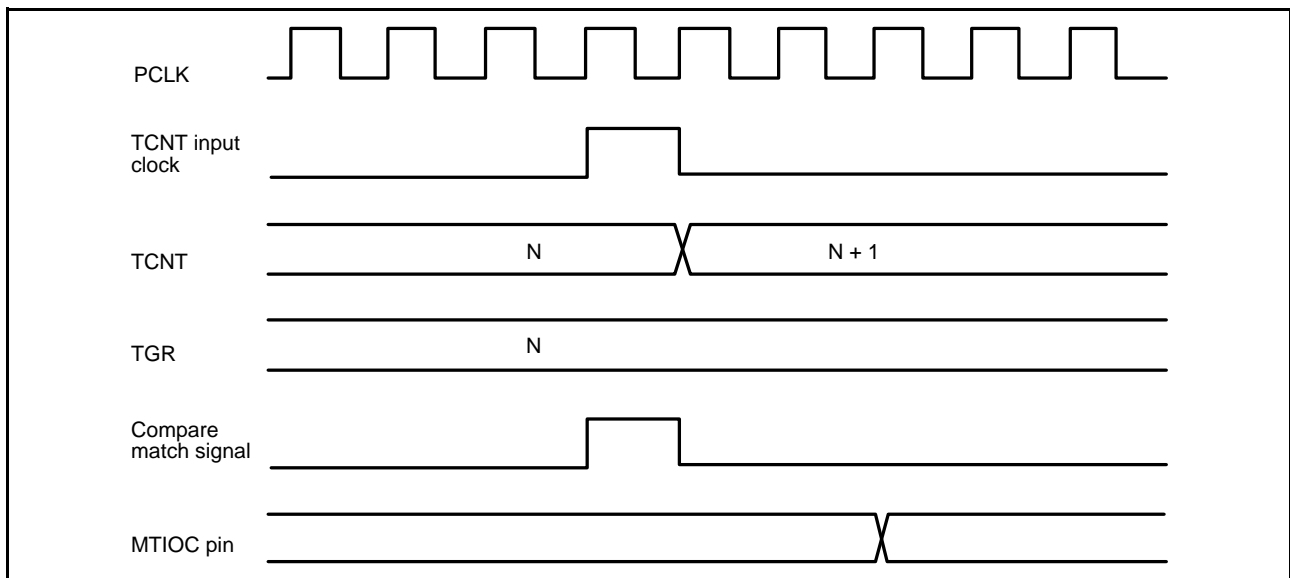


Figure 18.89 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode)

(3) Input Capture Signal Timing

Figure 18.90 shows the input capture signal timing.

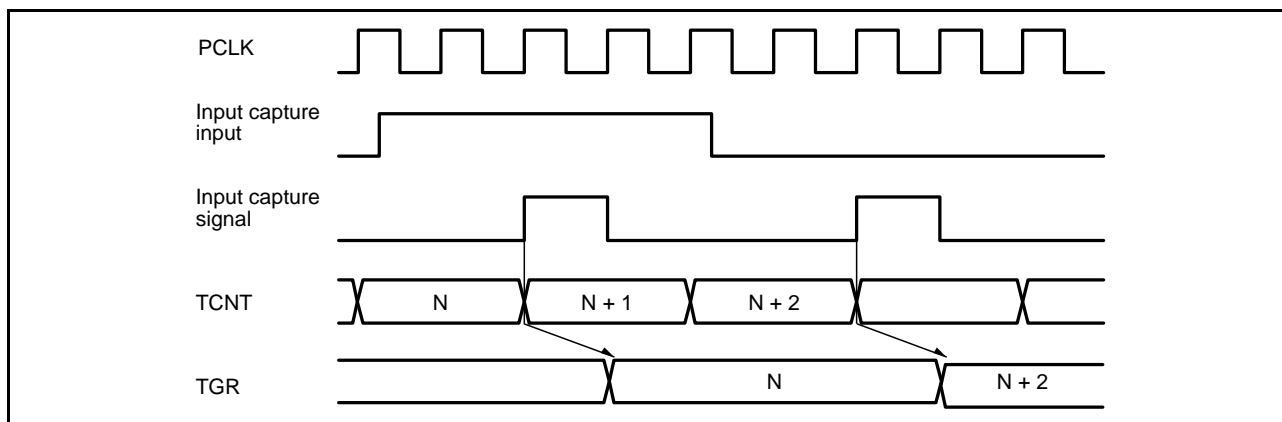


Figure 18.90 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 18.91 and Figure 18.92 show the timing when counter clearing on compare match is specified, and Figure 18.93 shows the timing when counter clearing on input capture is specified.

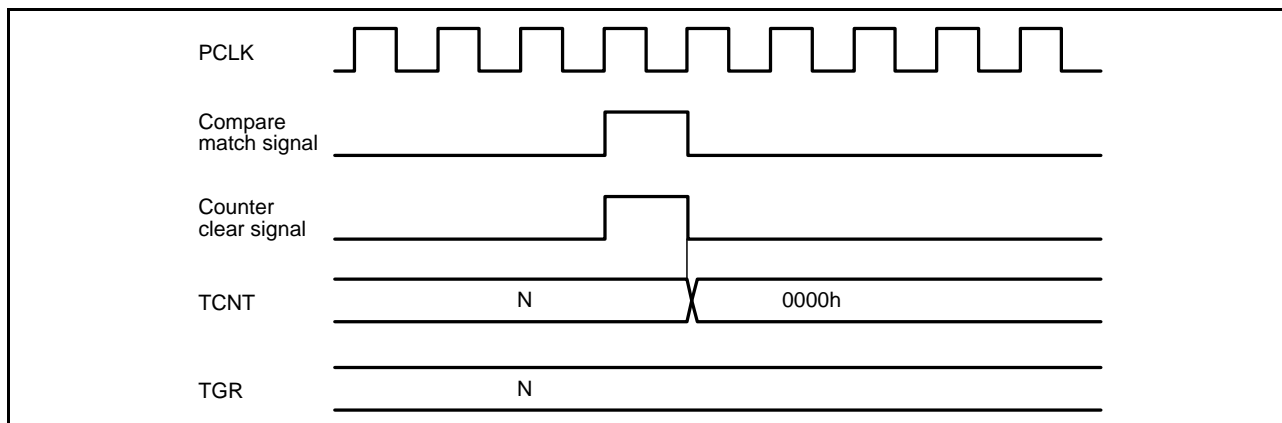


Figure 18.91 Counter Clear Timing (Compare Match) (Channels 0 to 4 or 6 to 10)

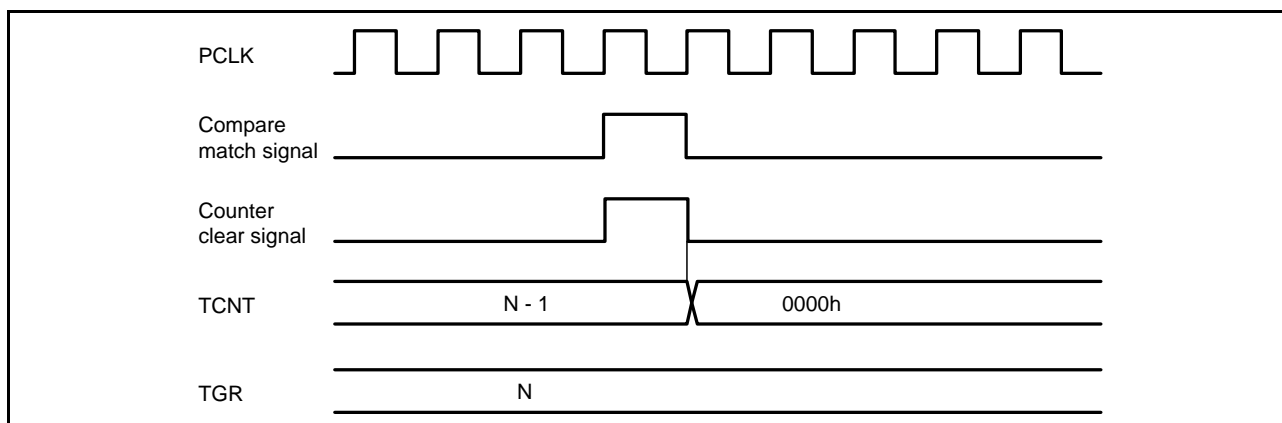


Figure 18.92 Counter Clear Timing (Compare Match) (Channel 5 or 11)

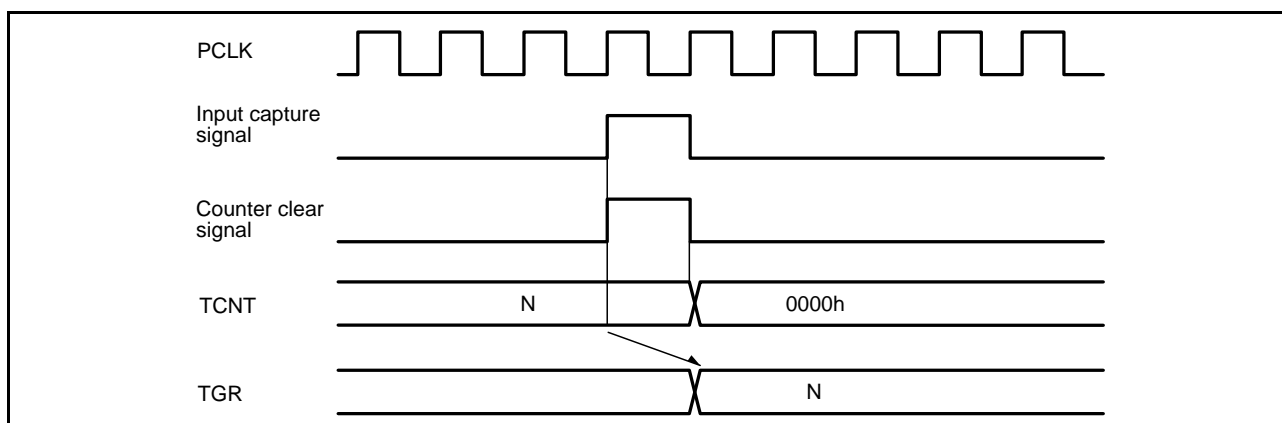


Figure 18.93 Counter Clear Timing (Compare Match) (Channels 0 to 5 or 6 to 11)

(5) Buffer Operation Timing

Figure 18.94 to Figure 18.96 show the timing in buffer operation.

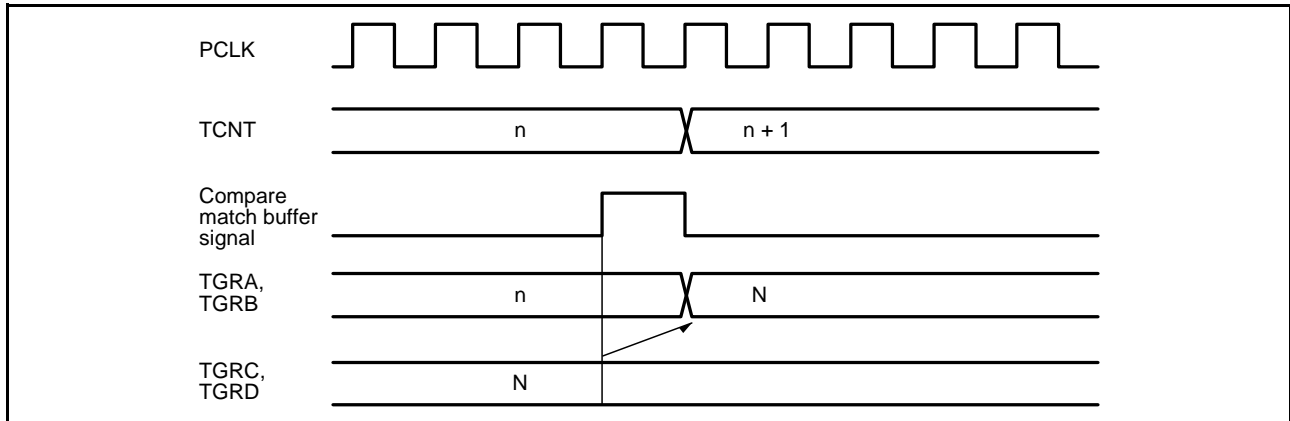


Figure 18.94 Buffer Operation Timing (Compare Match)

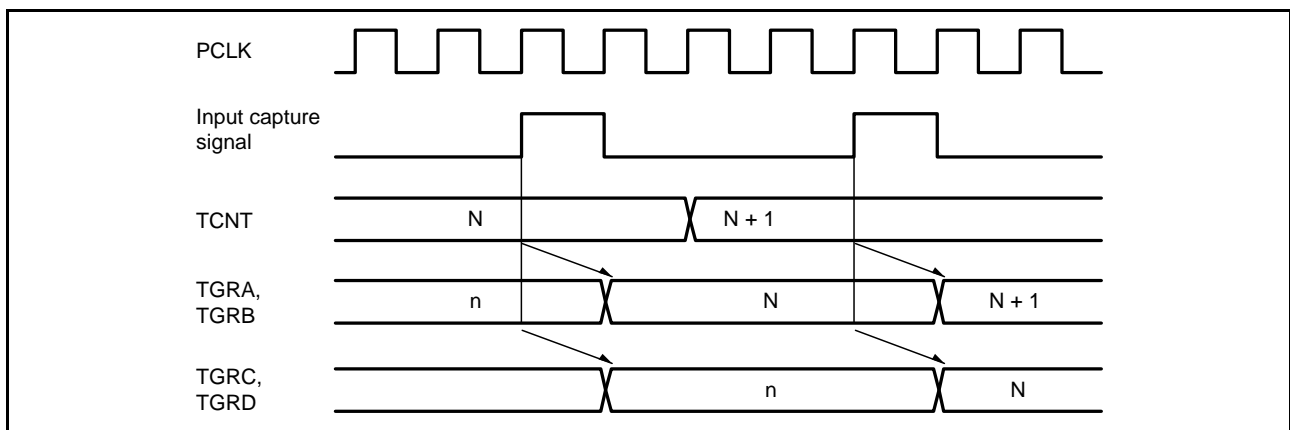


Figure 18.95 Buffer Operation Timing (Input Capture)

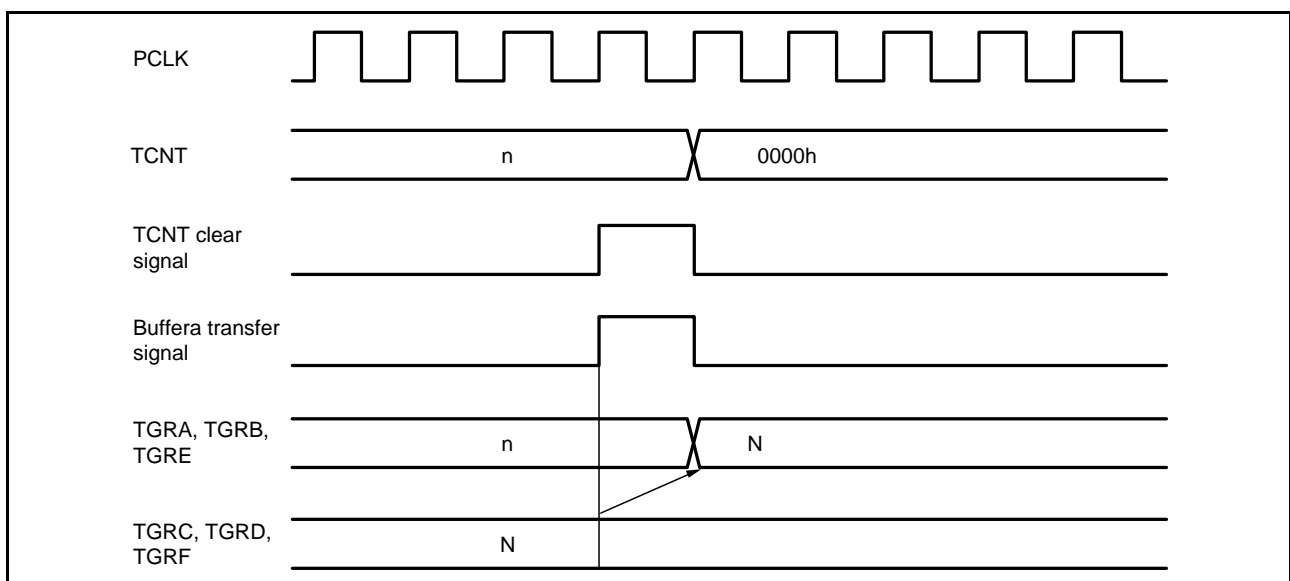


Figure 18.96 Buffer Operation Timing (when TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 18.97 to Figure 18.99 show the buffer transfer timing in complementary PWM mode.

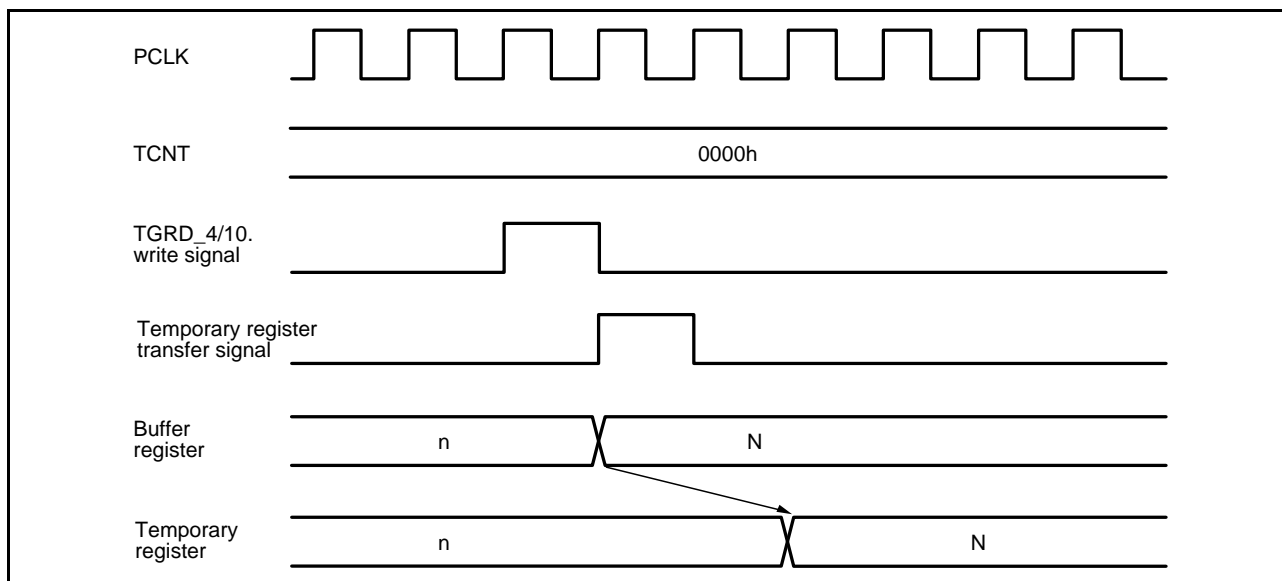


Figure 18.97 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

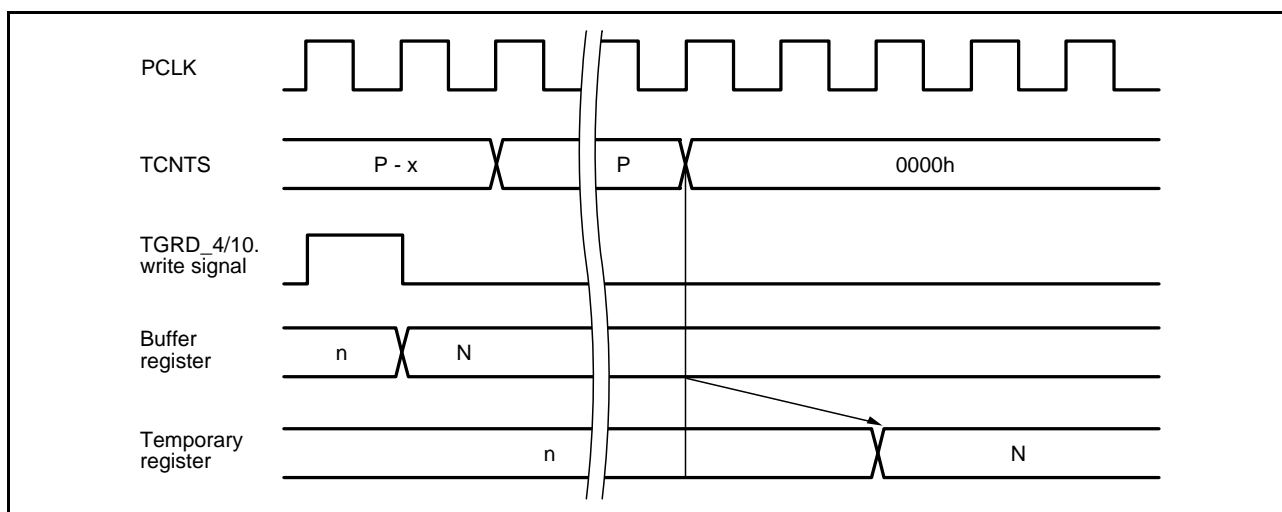


Figure 18.98 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

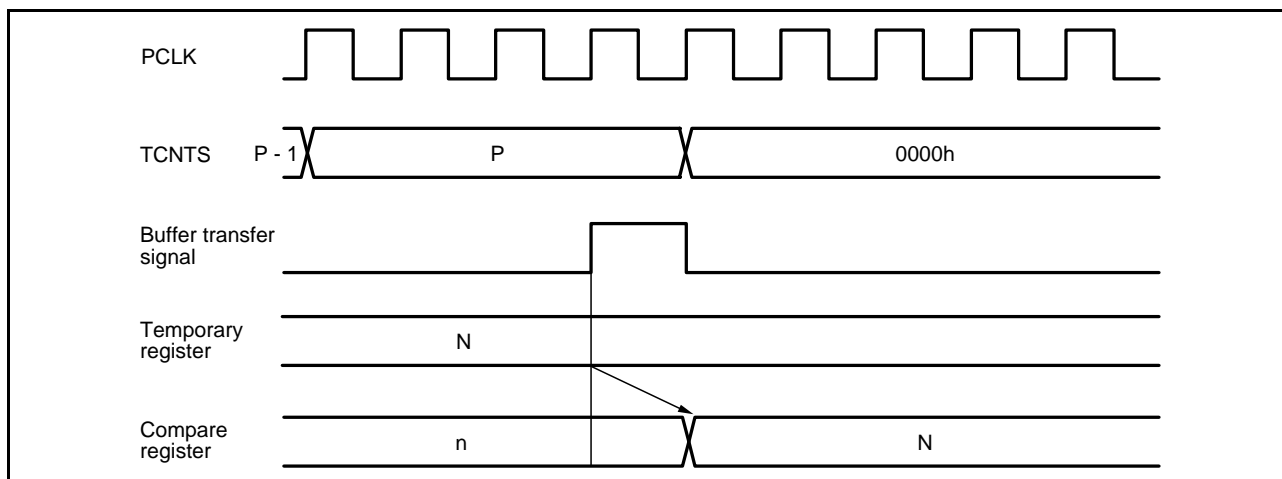


Figure 18.99 Transfer Timing from Temporary Register to Compare Register

18.5.2 Interrupt Signal Timing

(1) Timing for TGI Interrupt by Compare Match

Figure 18.100 and Figure 18.101 show the TGI interrupt request signal timing on compare match.

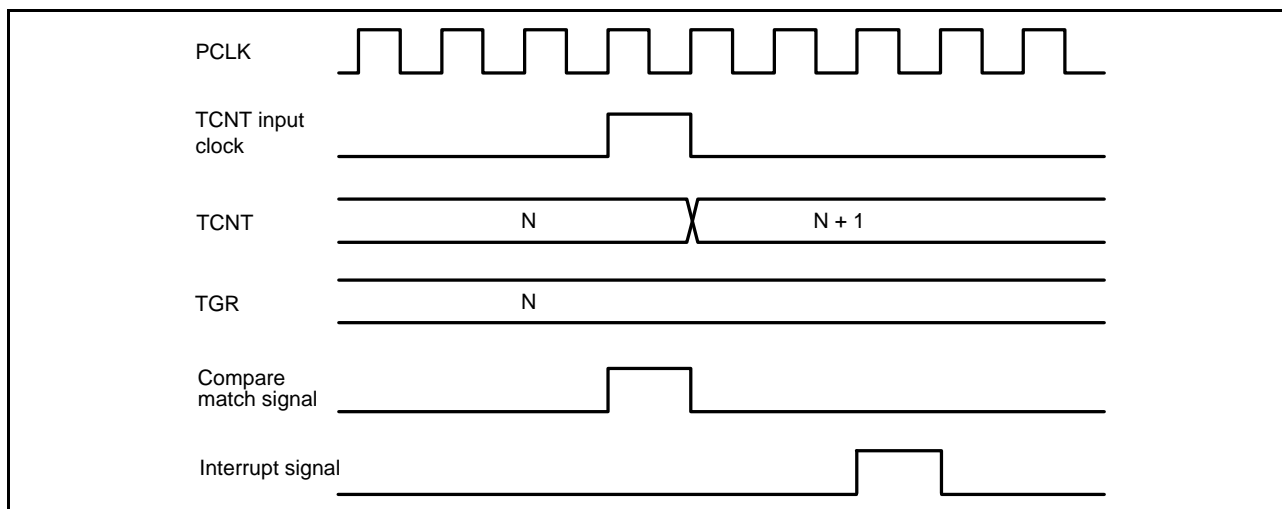


Figure 18.100 TGI Interrupt Timing (Compare Match) (Channels 0 to 4 or 6 to 10)

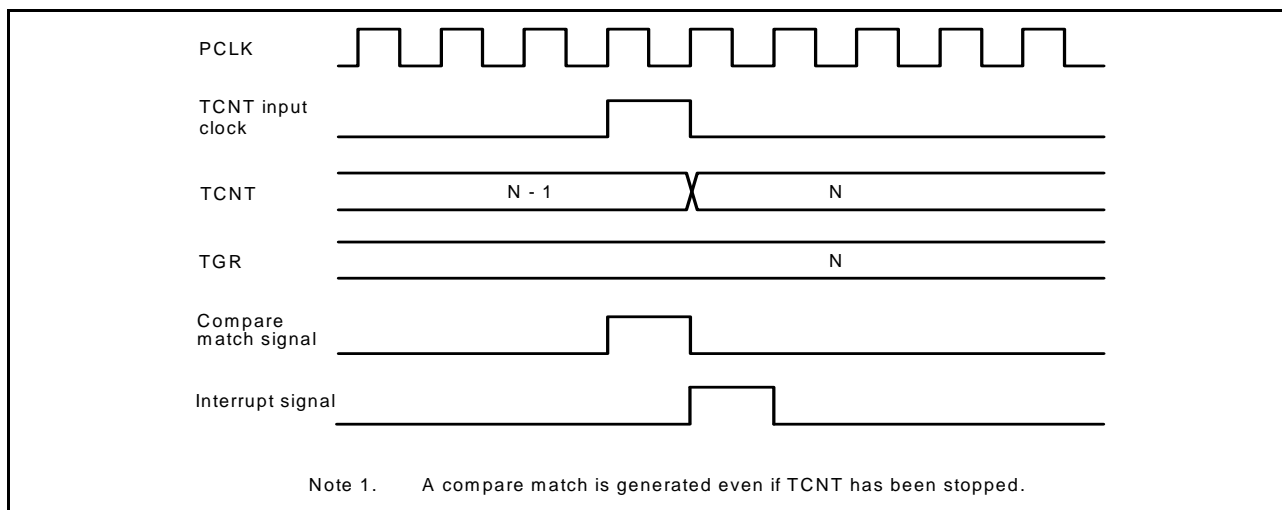


Figure 18.101 TGI Interrupt Timing (Compare Match) (Channel 5 or 11)

(2) Timing for TGI Interrupt Setting by Input Capture

Figure 18.102 and Figure 18.103 show TGI interrupt request signal timing on input capture.

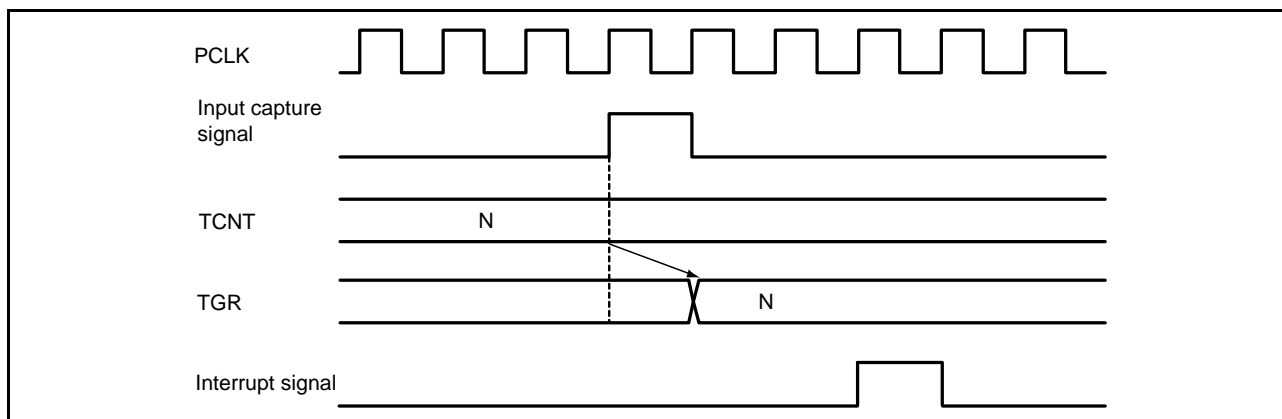


Figure 18.102 TGI Interrupt Timing (Input Capture) (Channels 0 to 4 or 6 to 10)

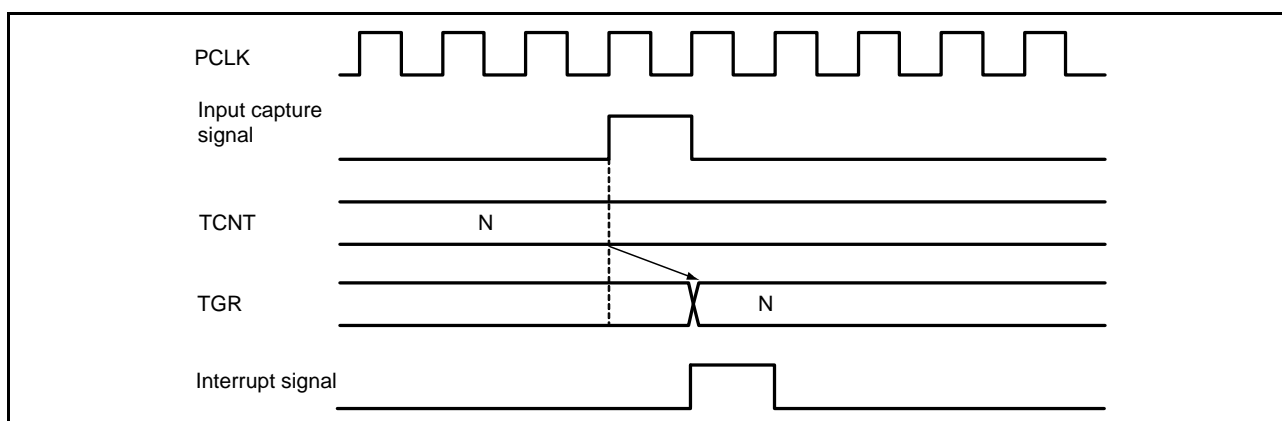


Figure 18.103 TGI Interrupt Timing (Input Capture) (Channel 5 or 11)

(3) TCIV and TCIU Interrupt Timing

Figure 18.104 shows the TCIV interrupt request signal timing on overflow.

Figure 18.105 shows the TCIU interrupt request signal timing on underflow.

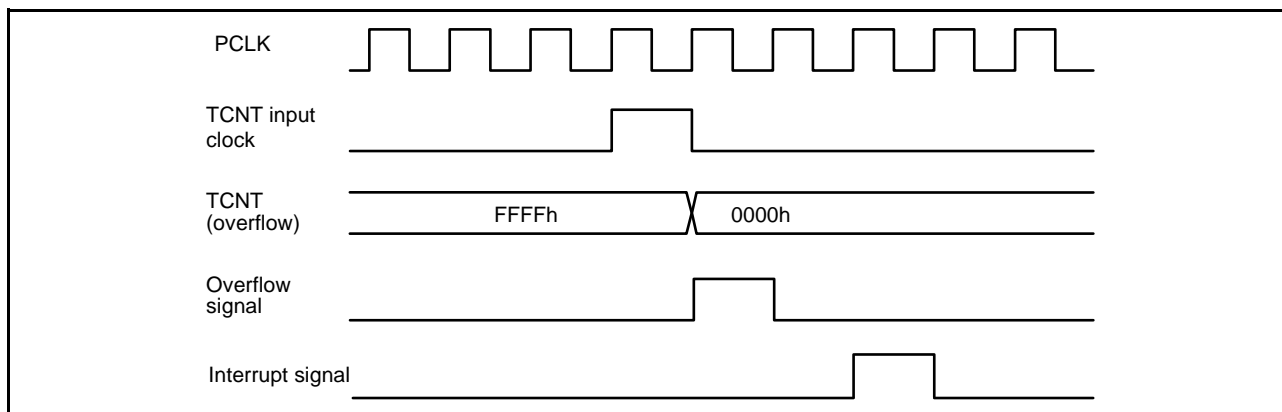


Figure 18.104 TCIV Interrupt Setting Timing

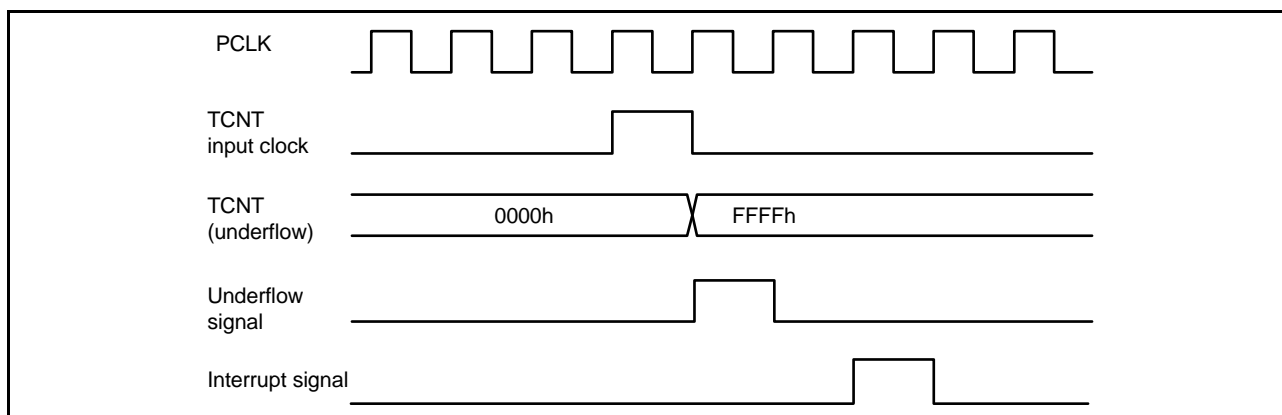


Figure 18.105 TCIU Interrupt Setting Timing

18.6 Usage Notes

18.6.1 Module Clock Stop Mode Setting

MTU operation can be disabled or enabled using the module stop control register. MTU operation is stopped with the initial setting. Register access is enabled by clearing the module clock stop mode. For details, refer to section 9, Low Power Consumption.

18.6.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states for single-edge detection, and at least 2.5 states 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 states, and the pulse width must be at least 2.5 states. Figure 18.106 shows the input clock conditions in phase counting mode.

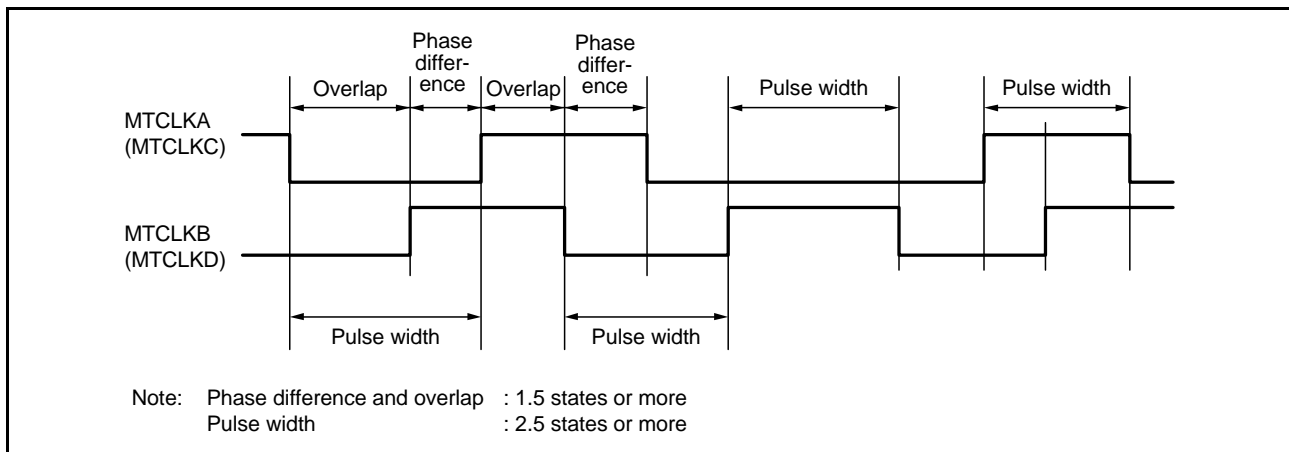


Figure 18.106 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode (Unit 0)

18.6.3 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 TCNU updates the matched count value). Consequently, the actual counter frequency is given by the following formula:

- Channels 0 to 4 and 6 to 10

$$f = \frac{PCLK}{(N+1)}$$

- Channels 5 and 11

$$f = \frac{PCLK}{N}$$

f: Counter frequency
PCLK: MTU clock operating frequency
N: TGR setting

18.6.4 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 18.107 shows the timing in this case.

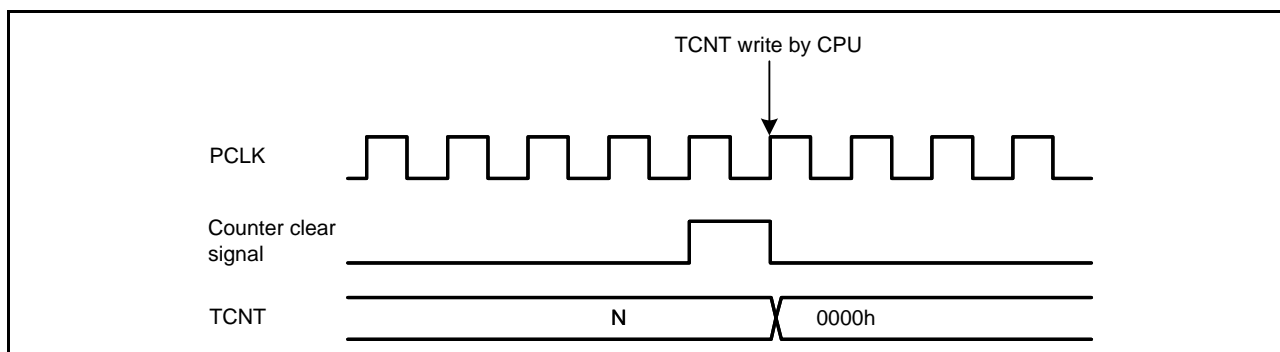


Figure 18.107 Contention between TCNT Write and Clear Operations

18.6.5 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 18.108 shows the timing in this case.

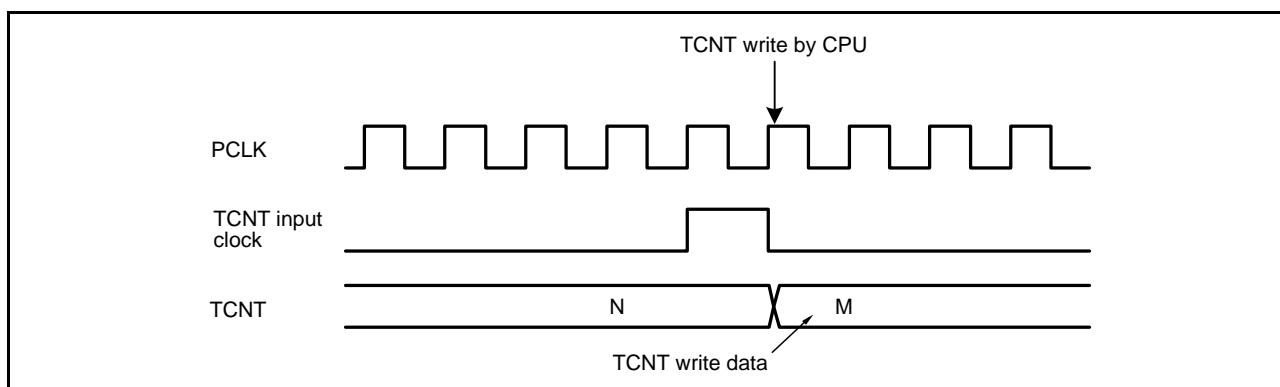


Figure 18.108 Contention between TCNT Write and Increment Operations

18.6.6 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 18.109 shows the timing in this case.

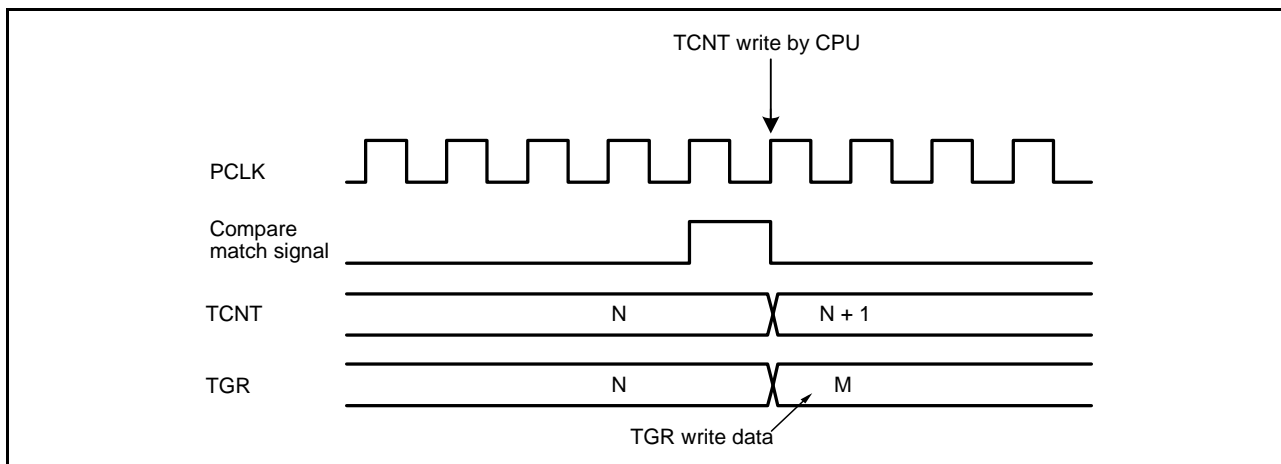


Figure 18.109 Contention between TGR Write Operation and Compare Match

18.6.7 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 18.110 shows the timing in this case.

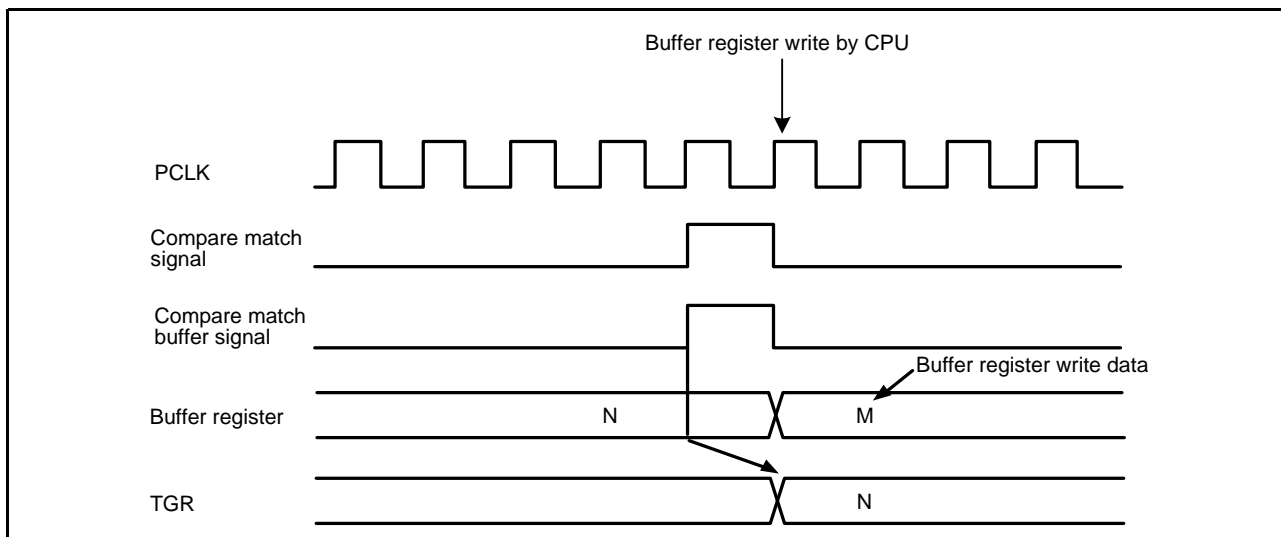


Figure 18.110 Contention between Buffer Register Write Operation and Compare Match

18.6.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the buffer transfer mode register (TBTM), if TCNT clearing occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 18.111 shows the timing in this case.

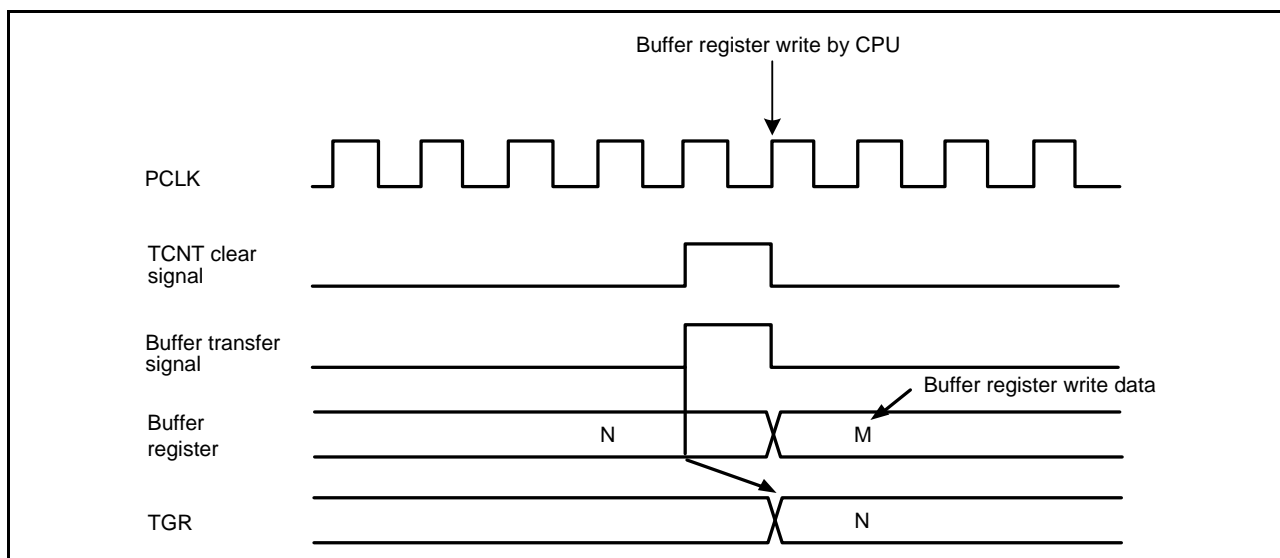


Figure 18.111 Contention between Buffer Register Write and TCNT Clear Operations

18.6.9 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 in channels 0 to 4 (or 6 to 10), and the data after input capture transfer in channel 5 (or 11).

Figure 18.112 and Figure 18.113 show the timing in this case.

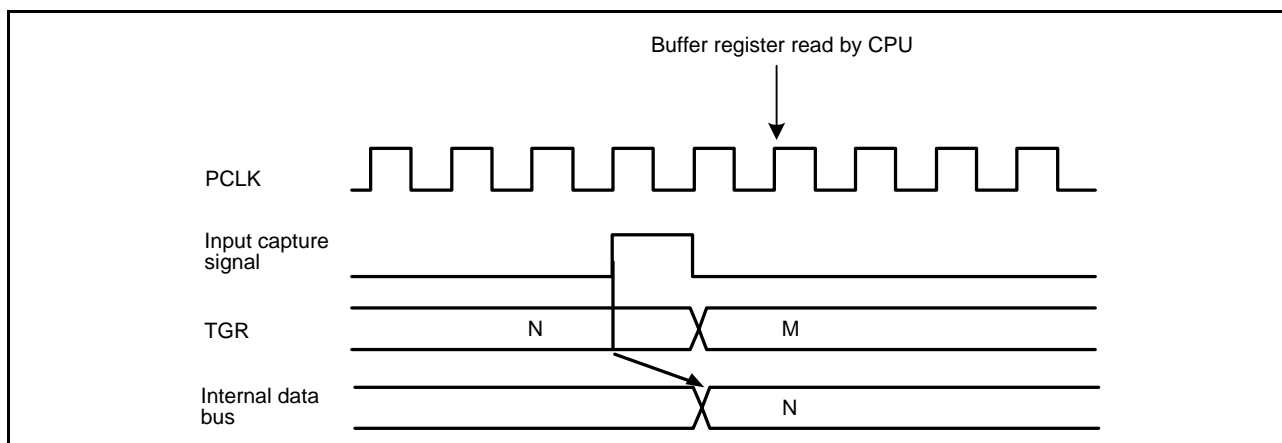


Figure 18.112 Contention between TGR Read Operation and Input Capture (Channels 0 to 4 or 6 to 10)

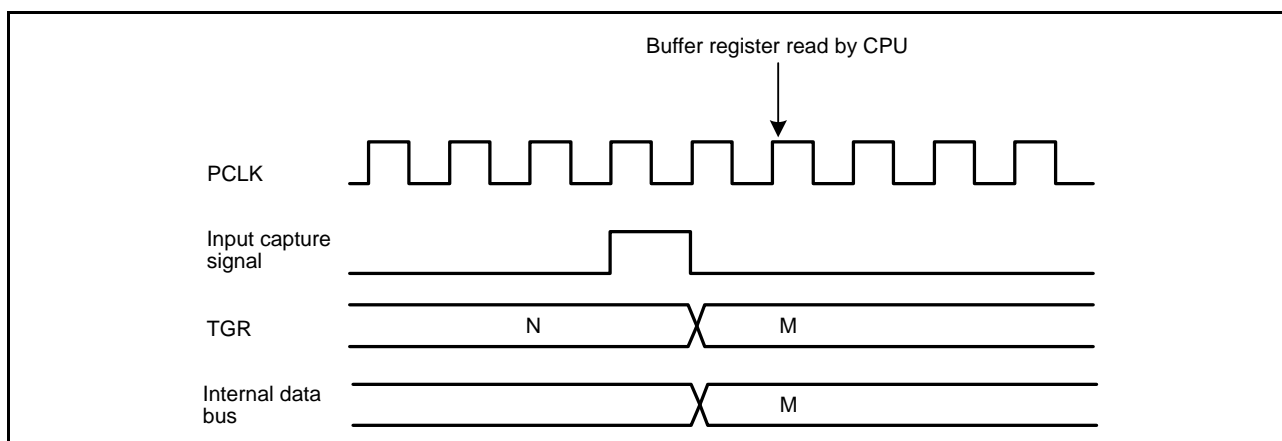


Figure 18.113 Contention between TGR Read Operation and Input Capture (Channel 5 or 11)

18.6.10 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 channels 0 to 4 (or 6 to 10). In channel 5 (or 11), the TGR write operation is performed and the input capture signal is generated.

Figure 18.114 and Figure 18.115 show the timing in this case.

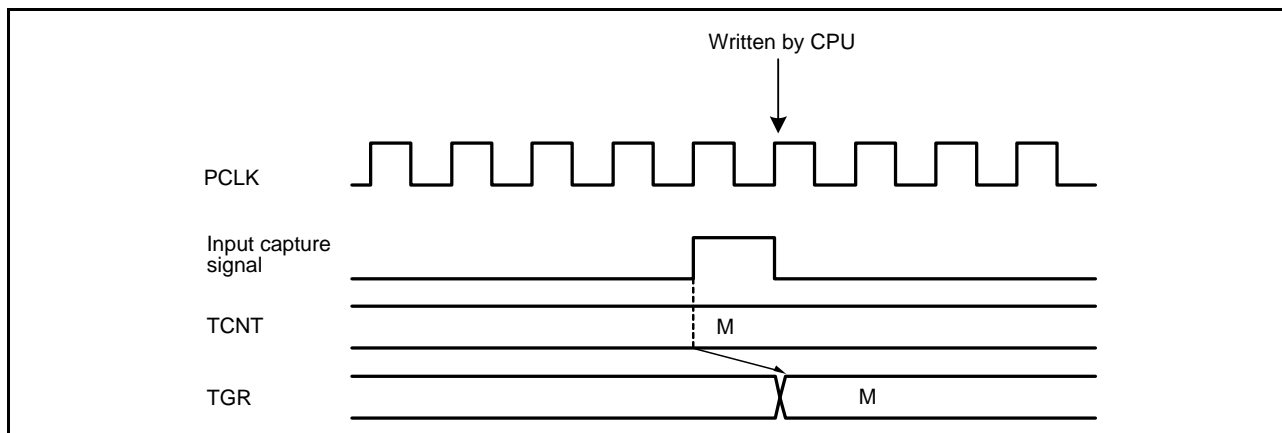


Figure 18.114 Contention between TGR Write Operation and Input Capture (Channels 0 to 4 or 6 to 10)

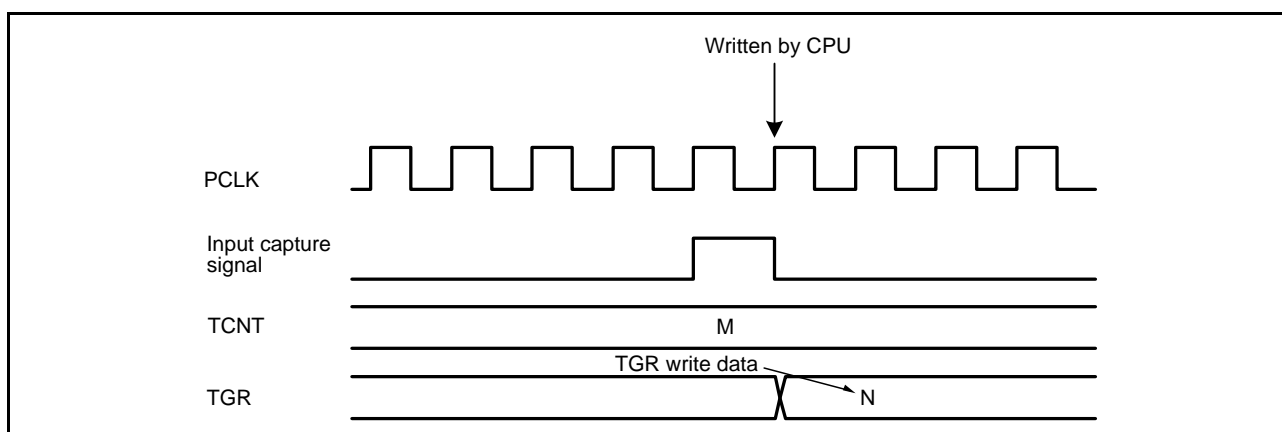


Figure 18.115 Contention between TGR Write Operation and Input Capture (Channel 5 or 11)

18.6.11 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 18.116 shows the timing in this case.

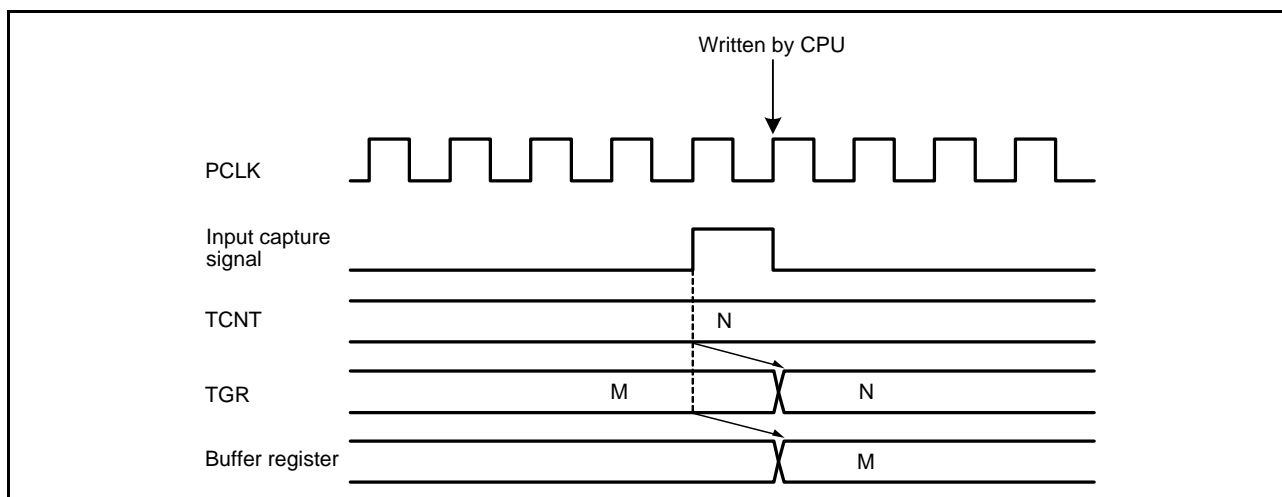


Figure 18.116 Contention between Buffer Register Write Operation and Input Capture

18.6.12 Contention between MTU_n.TCNT Write Operation and Overflow/Underflow in Cascaded Operation (n = 2 or 8)

With timer counters MTU_m.TCNT and MTU_n.TCNT in a cascade, when a contention occurs between MTU_m.TCNT counting (an MTU_n.TCNT overflow/underflow) and the MTU_n.TCNT write cycle, the MTU_n.TCNT write operation is performed and the MTU_m.TCNT count signal is disabled. In this case, if MTU_m.TGRA works as a compare match register and there is a match between the MTU_m.TGRA and MTU_m.TCNT values, a compare match signal is issued.

Furthermore, when the MTU_m.TCNT count clock is selected as the input capture source of channel 0 or 6, MTU₀.TGRA to MTU₀.TGRD (or MTU₆.TGRA to MTU₆.TGRD) work in input capture mode. In addition, when the MTU₀.TGRC (or MTU₆.TGRC) compare match/input capture is selected as the input capture source of MTU_m.TGRB, MTU_m.TGRB works in input capture mode. (n = 2 or 8, m = 1 or 7)

Figure 18.117 shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize channels 1 and 2 (or channels 7 and 8).

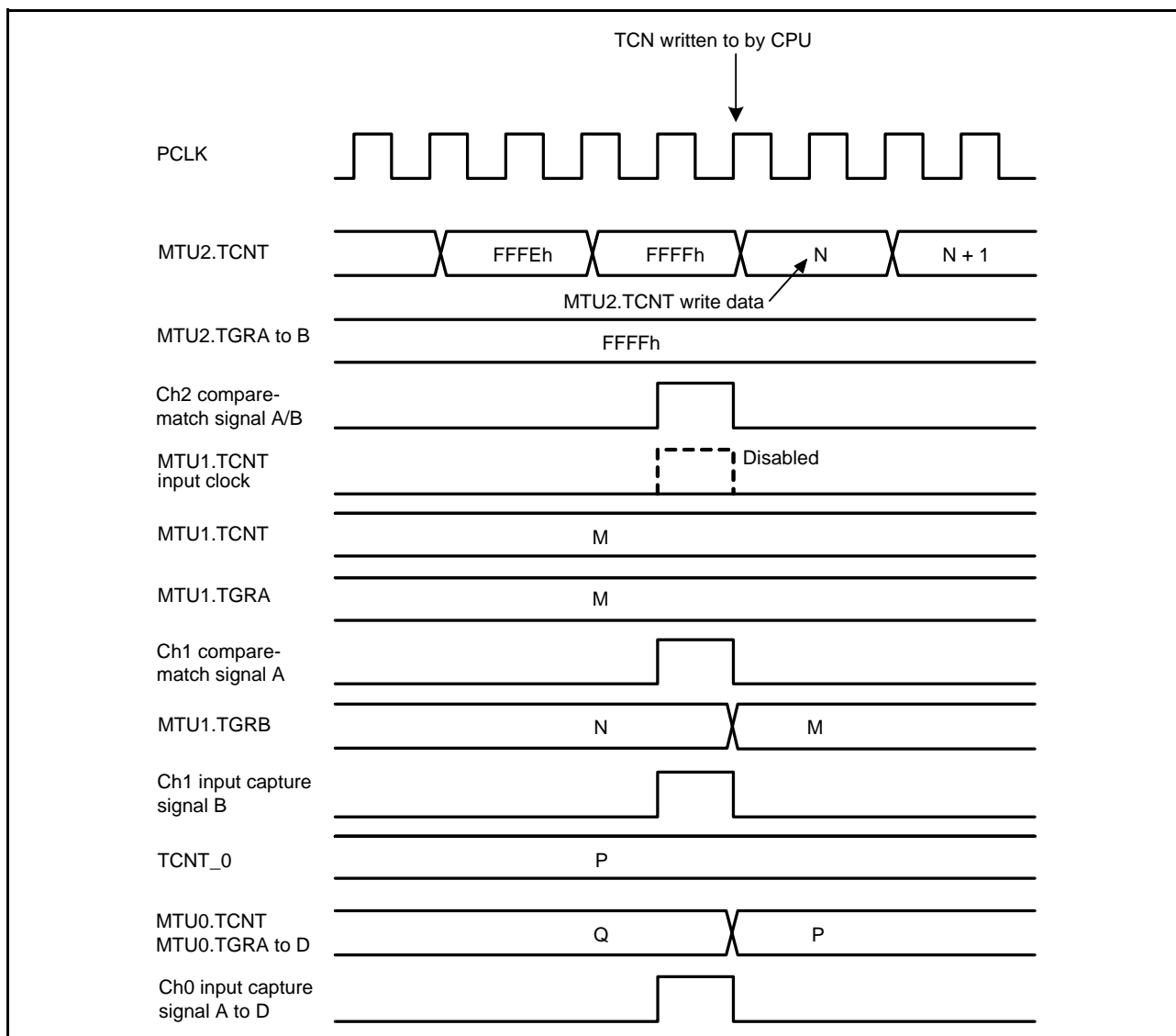


Figure 18.117 Contention between MTU₂.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

18.6.13 Counter Value when Stopped in Complementary PWM Mode

When counting operation in MTUn.TCNT and MTUm.TCNT is stopped in complementary PWM mode, the MTUn.TCNT value is set to the timer dead time register (TDDR) value and MTUm.TCNT is set to 0000h. (n = 3 or 9, m = 4 or 10)

When operation is restarted in complementary PWM mode, counting begins automatically from the initial setting state. Figure 18.118 shows this operation.

When counting begins in another operating mode, be sure to make initial settings in MTUn.TCNT and MTUm.TCNT.

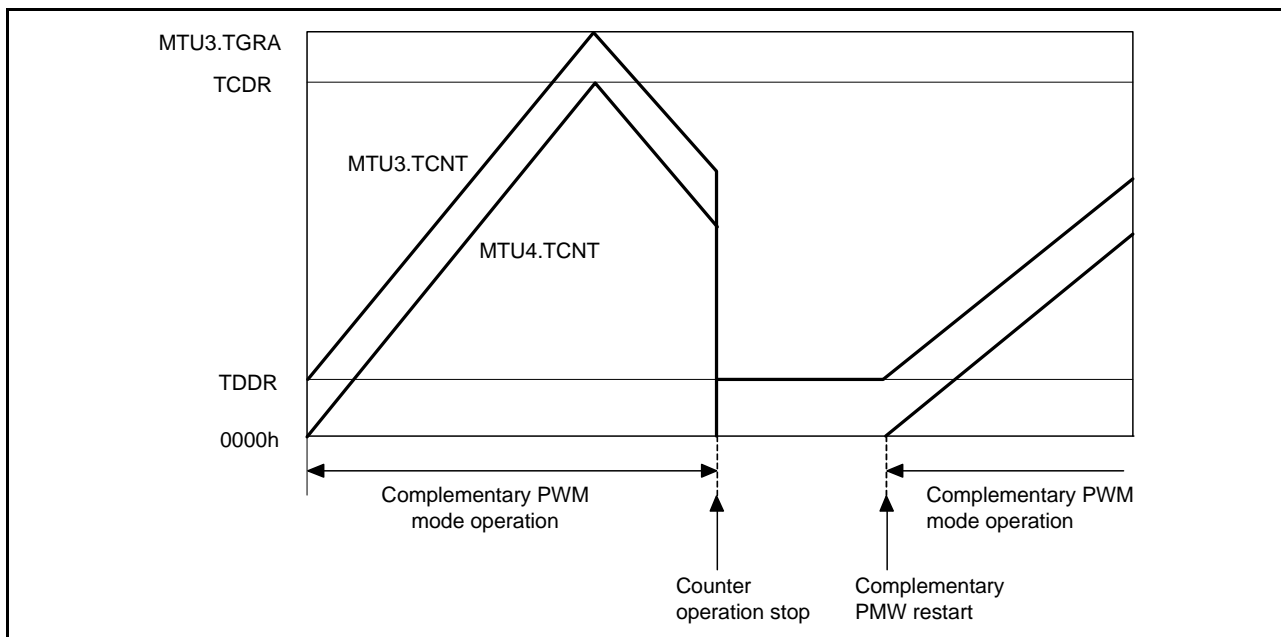


Figure 18.118 Counter Value when Stopped in Complementary PWM Mode (MTU3 and MTU4 Operation)

18.6.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM cycle set register (MTUn.TGRA), timer cycle data register (TCDR), and duty set registers (MTUn.TGRB, MTUm.TGRA, and MTUm.TGRB) in complementary PWM mode, be sure to use buffer operation. Also BFA and BFB bits in MTUm.TMDR should be set to 0. Setting the BFA bit in MTUm.TMDR to 1 disables MTIOCmC pin waveform output.

In complementary PWM mode, buffer operation in channels 3 and 4 (or channels 9 and 10) depends on the settings in bits BFA and BFB of MTUn.TMDR. When the BFA bit in MTUn.TMDR is set to 1, MTUn.TGRC functions as a buffer register for MTUn.TGRA. At the same time, MTUm.TGRC functions as a buffer register for MTUm.TGRA, and TCBR functions as a buffer register for TCDR. (n = 3 or 9, m = 4 or 10)

18.6.15 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the BFA and BFB bits in MTU_n.TMDR to 0. The MTIOC_mC pin cannot output waveforms if the BFA bit in MTU_n.TMDR is set to 1.

In reset-synchronized PWM mode, buffer operation in channels 3 and 4 (or channels 9 and 10) depends on the settings in the BFA and BFB bits of MTU_n.TMDR. For example, if the BFA bit in MTU_n.TMDR is set to 1, MTU_n.TGRC functions as a buffer register for MTU_n.TGRA. At the same time, MTU_m.TGRC functions as a buffer register for MTU_m.TGRA.

While the MTU_n.TGRC and MTU_n.TGRD are operating as buffer registers, the corresponding TGIC and TGID interrupt requests are never generated. (n = 3 or 9, m = 4 or 10)

Figure 18.119 shows an example of MTU3.TGR, MTU4.TGR, MTIOC3, and MTIOC4 operation with the BFA and BFB bits in MTU3.TMDR set to 1 and the BFA and BFB bits in MTU4.TMDR set to 0.

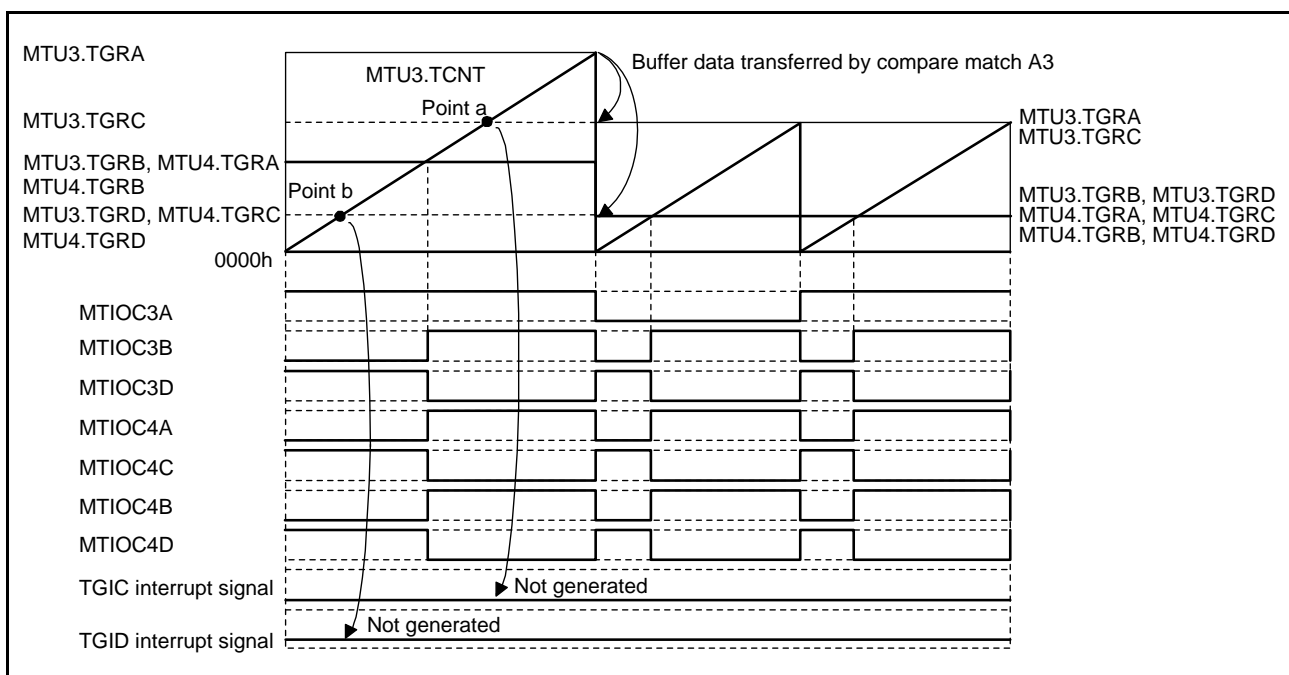


Figure 18.119 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

18.6.16 Overflow Flags in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTUn.TCNT and MTUm.TCNT start counting when the CST3 bit of TSTR is set to 1. In this state, the MTUm.TCNT count clock source and count edge are determined by the MTUn.TCR setting.

In reset-synchronized PWM mode, with cycle register MTUn.TGRA set to FFFFh and the MTUn.TGRA compare match selected as the counter clearing source, MTUn.TCNT and MTUm.TCNT count up to FFFFh, then a compare match occurs with MTUn.TGRA, and MTUn.TCNT and MTUm.TCNT are both cleared (n = 3 or 9, m = 4 or 10). In this case, the corresponding TCIV interrupt request is not generated.

Figure 18.120 shows an operation example in reset-synchronized PWM mode with cycle register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match specified for the counter clearing source without synchronous operation setting.

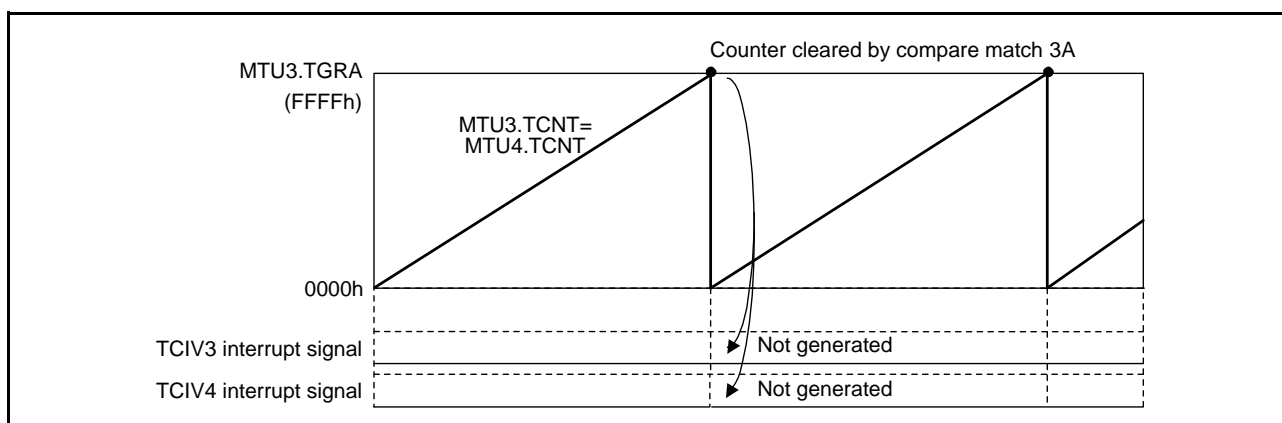


Figure 18.120 Overflow Flags in Reset-Synchronized PWM Mode

18.6.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, TCNT clearing takes precedence and the corresponding TCIV interrupt is not generated.

Figure 18.121 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

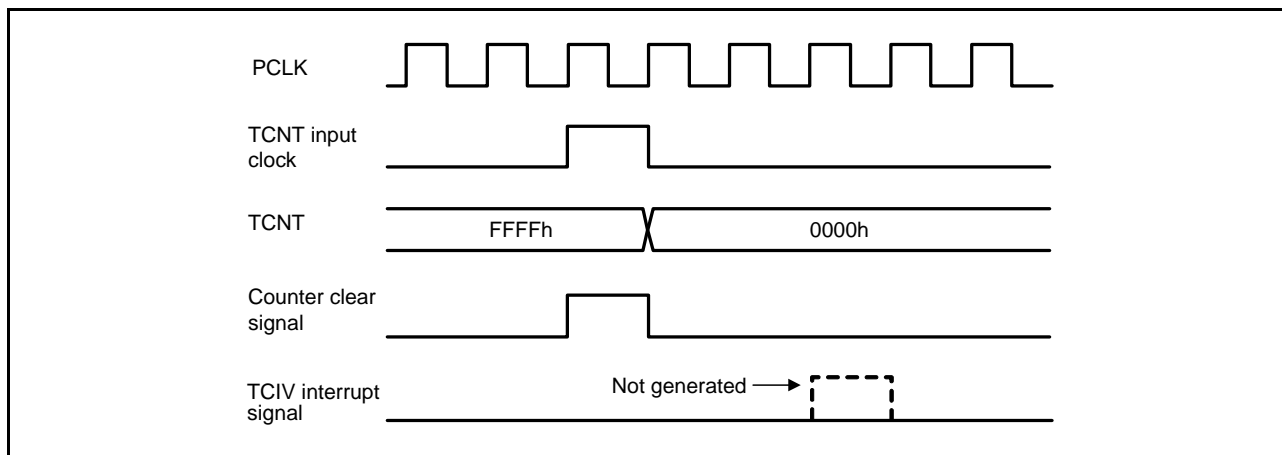


Figure 18.121 Contention between Overflow and Counter Clearing

18.6.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT counts up or down in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence and no interrupt request is generated.

Figure 18.122 shows the operation timing when there is contention between TCNT write operation and overflow.

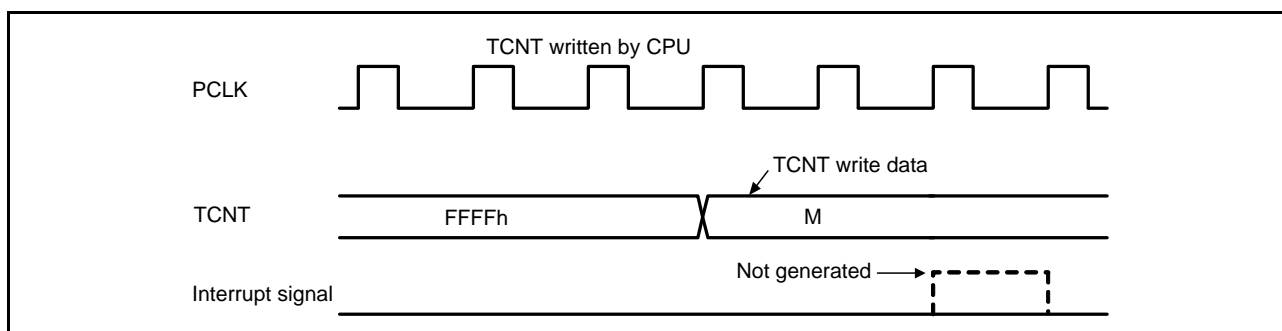


Figure 18.122 Contention between TCNT Write Operation and Overflow

18.6.19 Note on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from normal operation or PWM mode 1 to reset-synchronized PWM mode in channels 3 and 4 (or channels 9 and 10), if the counter is stopped while the output pins (MTIOCnB, MTIOCnD, MTIOCmA, MTIOCmC, MTIOCmB, and MTIOCmD) are held high 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 operation to reset-synchronized PWM mode, write 11h to MTUn.TIORH, MTUn.TIORL, MTUm.TIORH, and MTUm.TIORL to initialize the output pin state to low, then set the registers to the initial value (00h) before making the mode transition. (n = 3 or 9, m = 4 or 10)

When making a transition from PWM mode 1 to reset-synchronized PWM mode, switch to normal operation, initialize the output pin state to low, and then set the registers to the initial value (00h) before making the transition to reset-synchronized PWM mode.

18.6.20 Output Levels in Complementary PWM Mode and Reset-Synchronous PWM Mode

When the MTU3 or MTU4 module is in complementary PWM mode or reset-synchronous PWM mode, the levels output for the PWM waveform are set by the OLSN and OLSP bits in timer output control register 1 (TOCR1). Set TIOR to 00h when operation is in complementary PWM mode or reset-synchronous PWM mode.

18.6.21 Interrupts in the Module-Stop State

When an MTU2 is placed in the module-stop state and an interrupt is currently being requested, clearing of the interrupt factor for the CPU or the activating factor for the DMAC or DTC is not possible.

Accordingly, disable interrupts and so on before placing the module in the module-stop state.

18.6.22 Simultaneous Input Capture in MTUn.TCNT and MTUm.TCNT in Cascade Connection

When timer counters 1 and 2 (or 7 and 8) (MTU1.TCNT and MTU2.TCNT or MTU7.TCNT and MTU8.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 MTIOCnA and MTIOCmA or to MTIOCnB and MTIOCmB. This is because the input timing of MTIOCnA and MTIOCmA or of MTIOCnB and MTIOCmB may not be the same when external input-capture signals input into MTUn.TCNT and MTUm.TCNT are taken in synchronization with the internal clock. (n = 1 or 7, m = 2 or 8)

For example, MTU1.TCNT (the counter for upper 16 bits) does not capture the count-up value by an overflow from MTU2.TCNT (the counter for lower 16 bits) but captures the count value before the up-counting. In this case, the values of MTU1.TCNT = FFF1h and MTU2.TCNT = 0000h should be transferred to MTU1.TGRA and MTU2.TGRA or to MTU1.TGRB and MTU2.TGRB, but the values of MTU1.TCNT = FFF0h and MTU2.TCNT = 0000h are erroneously transferred.

The MTU has a new function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT (or MTU7.TCNT and MTU8.TCNT) with a single input-capture input as the trigger. This function allows reading of the 32-bit counter such that MTU1.TCNT and MTU2.TCNT (or MTU7.TCNT and MTU8.TCNT) are captured at the same time. For details, see section 18.2.8, Timer Input Capture Control Register (TICCR).

18.6.23 Notes when Complementary PWM Mode Output Protection Functions are not Used

The complementary PWM mode output protection functions are initially enabled. If the functions are not used, the POECR1 and POECR2 registers of POE should be set to 00h.

18.6.24 Preventing Abnormal Operation of Synchronous Clearing in Complementary PWM Mode

The phenomena listed below appear if condition 1 or 2 is satisfied while output-waveform control is enabled (TWCR.WRE = 1) during synchronous counter-clearing in complementary PWM mode.

- Dead time on the PWM output pin is shortened (or eliminated).
- The active level is output on the PWM inverse-output pin beyond the interval for output of the active level.

Condition 1: Due to the interval for suppression of initial output (10), the PWM output was synchronously cleared during the dead-time interval (see figure 18.123).

Condition 2: Due to the intervals for suppression of initial output (10) and (11), synchronous clearing proceeds while any relation from among $MTU3.TGRB \leq TDDR$, $MTU4.TGRA \leq TDDR$, and $MTU4.TGRB \leq TDDR$ is satisfied (see figure 18.124).

Use the method below to avoid these phenomena.

- Ensure that synchronous clearing proceeds while the values of all comparison registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) are set to at least twice the value of the dead-time data register (TDDR).

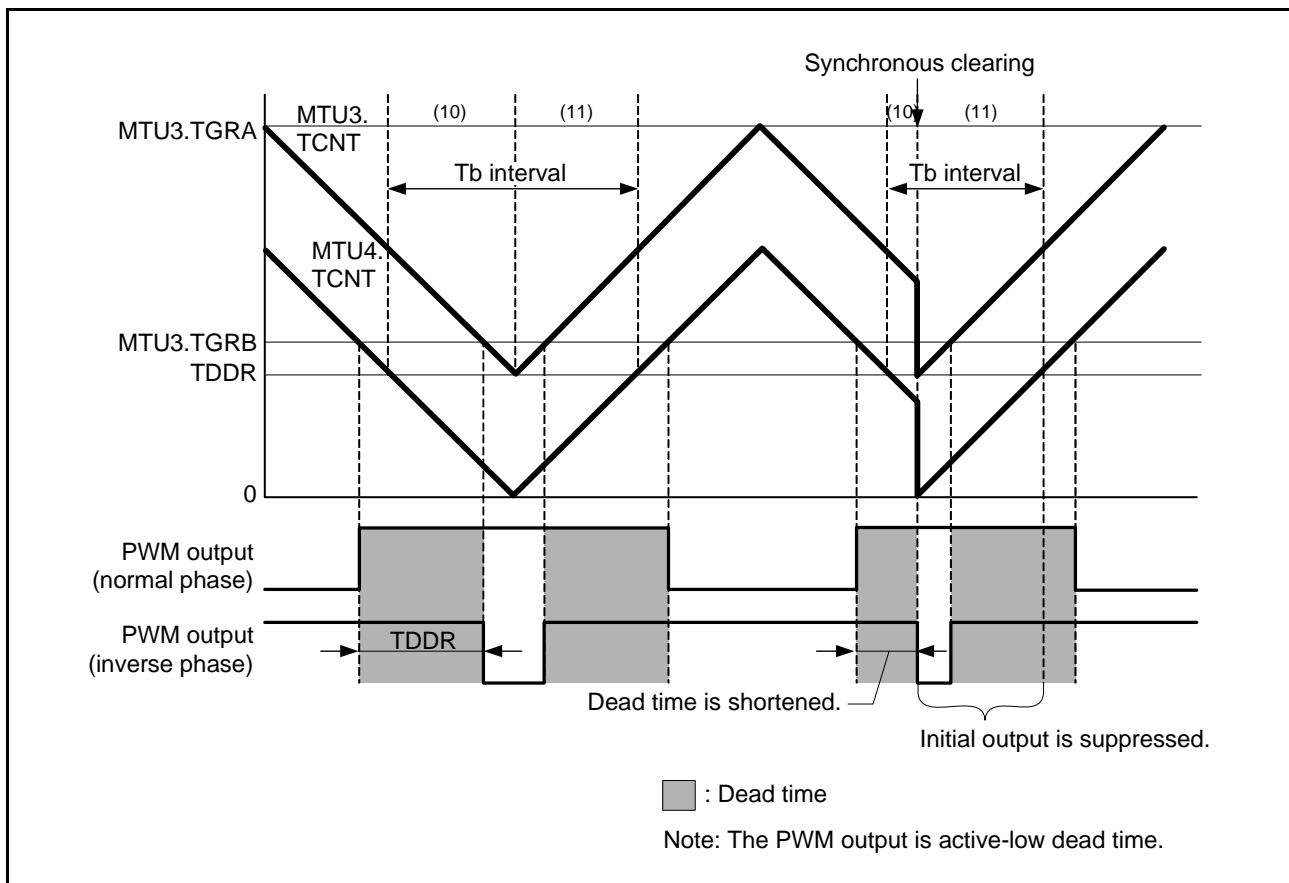


Figure 18.123 Example of Synchronous Clearing (when Condition 1 Holds)

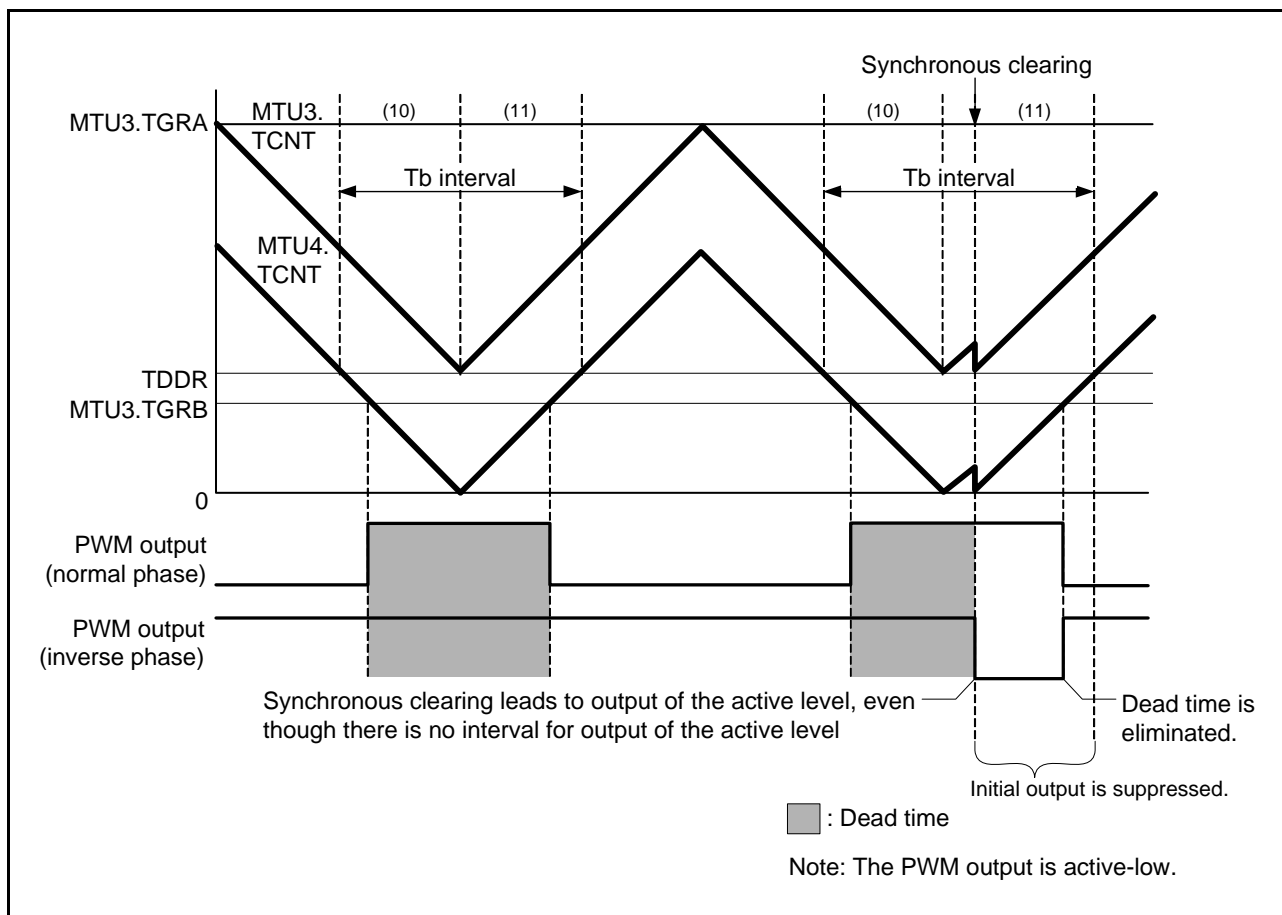


Figure 18.124 Example of Synchronous Clearing (when Condition 2 Holds)

18.7 MTU Output Pin Initialization

18.7.1 Operating Modes

The MTU has the following six operating modes. Waveforms can be output in any of these modes.

- Normal mode (channels 0 to 4 or channels 6 to 10)
- PWM mode 1 (channels 0 to 4 or channels 6 to 10)
- PWM mode 2 (channels 0 to 2 or channels 6 to 8)
- Phase counting modes 1 to 4 (channels 1 and 2 or channels 7 and 8)
- Complementary PWM mode (channels 3 and 4 or channels 9 and 10)
- Reset-synchronized PWM mode (channels 3 and 4 or channels 9 and 10)

This section describes how to initialize the MTU output pins in each of these modes.

18.7.2 Operation in Case of Re-Setting Due to Error during Operation

If an error occurs during MTU operation, MTU output should be cut off by the system. The output can be cut off by setting the non-active level to be output to the general I/O port by the data direction register (DDR) and the data register (DR) in the I/O port in advance, prohibiting the MTU pin output by setting the pin to function as the general output port, and then outputting the non-active level to the pin. For motor driving pins, output can also be cut by hardware, using port output enable (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 shown in Table 18.62.

Table 18.62 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)

[Legend]

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

18.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation

- When making a transition to a mode (Normal, PWM1, PWM2, or PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of TIOR setting.
- In PWM mode 1, since a waveform is not output to the MTIOCNB (MTIOCnD) pins (n = channel number), setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 1.
- In PWM mode 2, since a waveform is not output to the cycle register pins, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 2.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, setting TIOR will not initialize the buffer register pins. If initialization is required, clear buffer mode, carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR will not initialize the MTIOCNc (n = channel number) pin. To initialize the MTIOCNc (n = channel number) pin, clear buffer mode, carry out initialization, then set buffer mode again.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, switch to normal mode, perform initialization with TIOR, restore TIOR to its initial value, and temporarily disable output in channels 3 and 4 (or channels 9 and 10) with the timer output master enable register (TOER). After that, operate the MTU in accordance with the mode setting procedure (TOCR setting, TMDR setting, and TOER setting).

Note: Channel number is substituted for “n” indicated in this section.

Pin initialization procedures are described below for the numbered combinations in Table 18.62. The active level is assumed to be low.

(1) Operation when Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 18.125 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

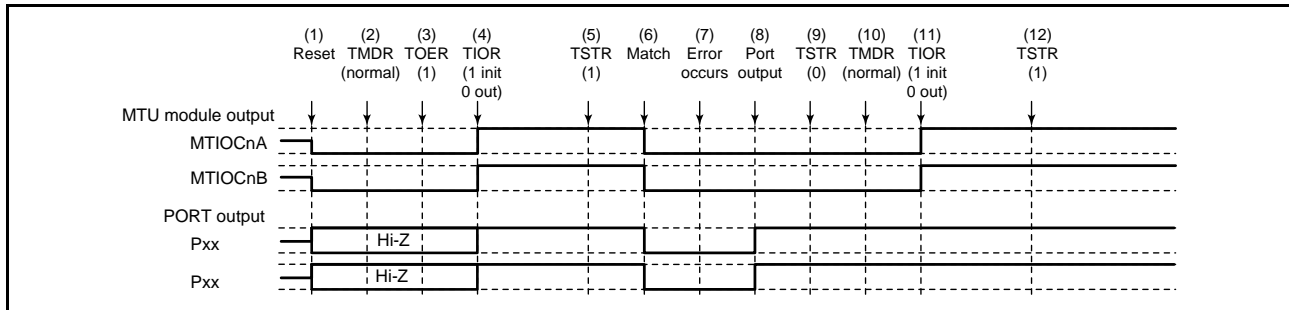


Figure 18.125 Error Occurrence in Normal Mode, Recovery in Normal Mode

- (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 channels 3 and 4 (or channels 9 and 10), enable output with TOER before initializing the pins with TIOR.
- (4) Initialize the pins with TIOR. (In the example, the initial output is high, and low is output on compare match occurrence.)
- (5) Start count operation by setting TSTR.
- (6) Output goes low on compare match occurrence.
- (7) An error occurs.
- (8) Disable the MTU output and select port output with TIOR and output the inverse of the active level.
- (9) Stop count operation by setting TSTR.
- (10) This step is not necessary when restarting in normal mode.
- (11) Initialize the pins with TIOR.
- (12) Restart operation by setting TSTR.

(2) Operation when Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 18.126 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

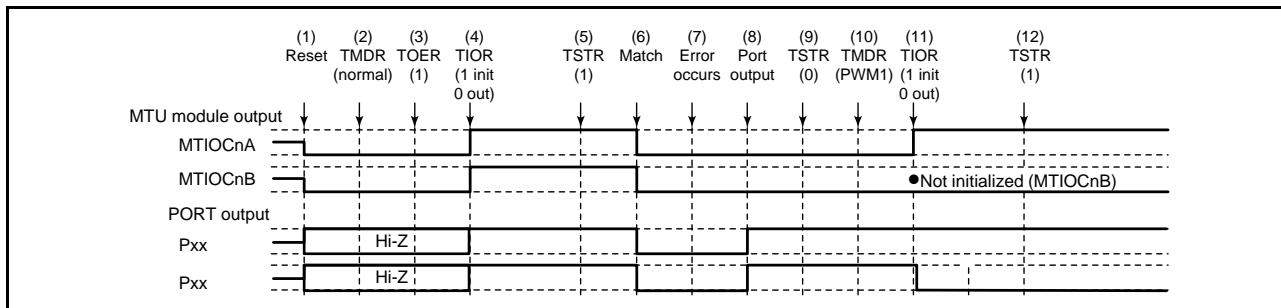


Figure 18.126 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 18.125.

(10) Set PWM mode 1.

(11) Initialize the pins with TIOR. (In PWM mode 1, the MTIOcN B side is not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 1.)

(12) Restart operation by setting TSTR.

(3) Operation when Error Occurs in Normal Mode and Operation is Restarted in PWM mode 2

Figure 18.127 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

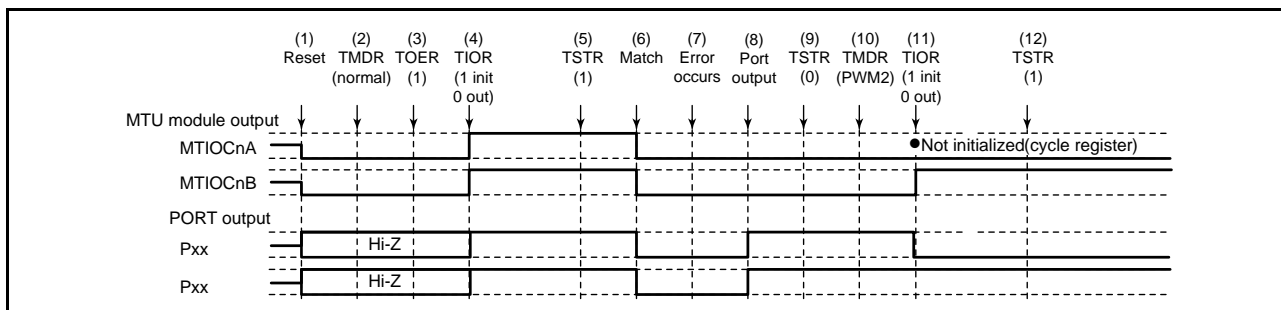


Figure 18.127 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 18.125.

(10) Set PWM mode 2.

(11) 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.)

(12) Restart operation by setting TSTR.

Note: PWM mode 2 can only be selected for channels 0 to 2 or 6 to 8, and therefore TOER setting is not necessary.

(4) Operation when Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 18.128 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

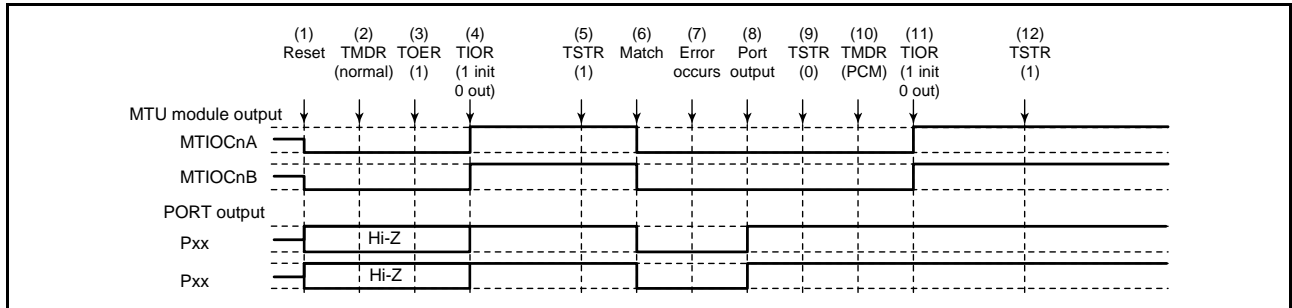


Figure 18.128 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 18.125.

(10) Set the phase counting mode.

(11) Initialize the pins with TIOR.

(12) Restart operation by setting TSTR.

Note: The phase counting mode can only be selected for channels 1 and 2, and therefore TOER setting is not necessary.

(5) Operation when Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 18.129 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

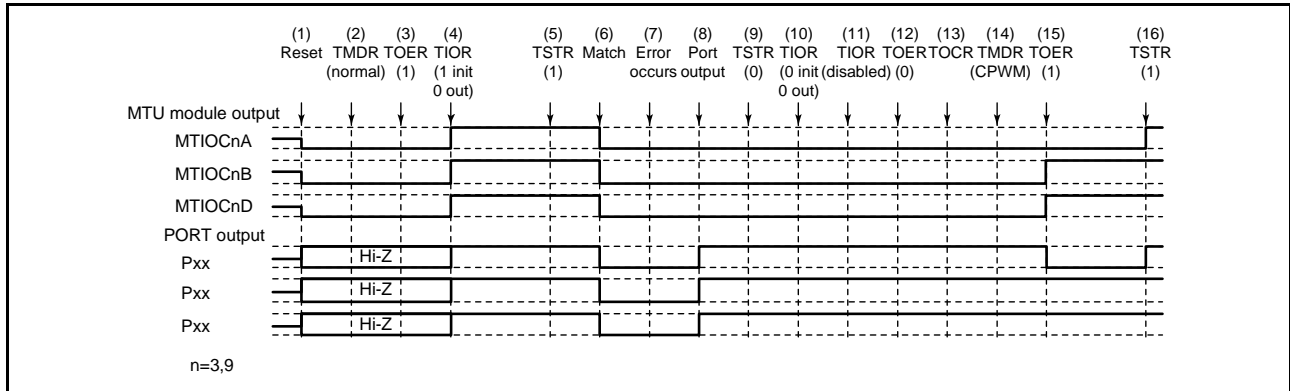


Figure 18.129 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

(1) to (9) are the same as in Figure 18.125.

- (10) Initialize the normal mode waveform generation section with TIOR.
- (11) Disable operation of the normal mode waveform generation section with TIOR.
- (12) Disable output in channels 3 and 4 (or channels 9 and 10) 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 channels 3 and 4 (or channels 9 and 10) with TOER.
- (16) Restart operation by setting TSTR.

(6) Operation when Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 18.130 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

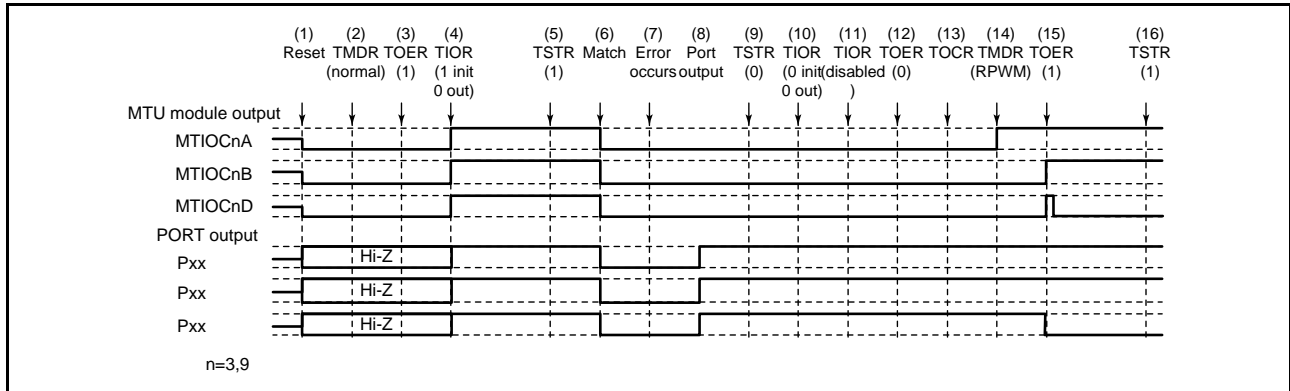


Figure 18.130 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (12) are the same as in Figure 18.125.

(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 channels 3 and 4 (or channels 9 and 10) with TOER.

(16) Restart operation by setting TSTR.

(7) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 18.131 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

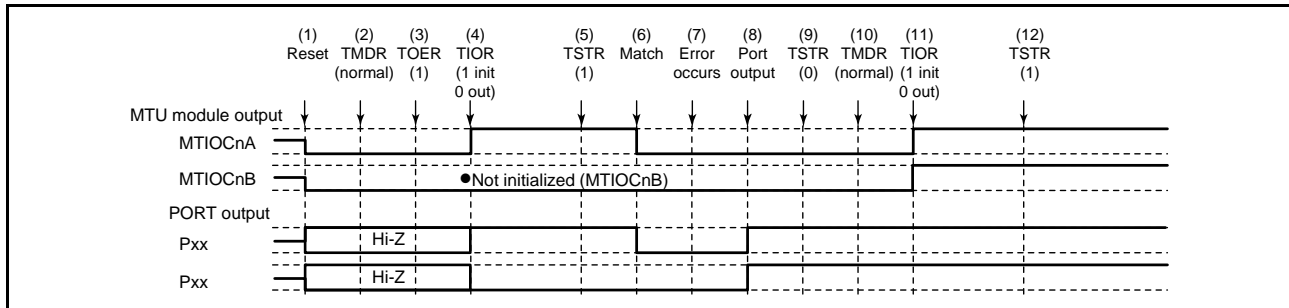


Figure 18.131 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

- (1) After a reset, the MTU output goes low and the ports enter high-impedance state.
- (2) Set PWM mode 1.
- (3) For channels 3 and 4 (or channels 9 and 10), enable output with TOER before initializing the pins with TIOR.
- (4) Initialize the pins with TIOR. (In the example, the initial output is high, and low level is output on compare match occurrence. In PWM mode 1, the MTIOCnB side is not initialized.)
- (5) Start count operation by setting TSTR.
- (6) Output goes low on compare match occurrence.
- (7) An error occurs.
- (8) Disable the MTU output and select port output with TIOR and output the inverse of the active level.
- (9) Stop count operation by setting TSTR.
- (10) Set normal mode.
- (11) Initialize the pins with TIOR.
- (12) Restart operation by setting TSTR.

(8) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

Figure 18.132 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

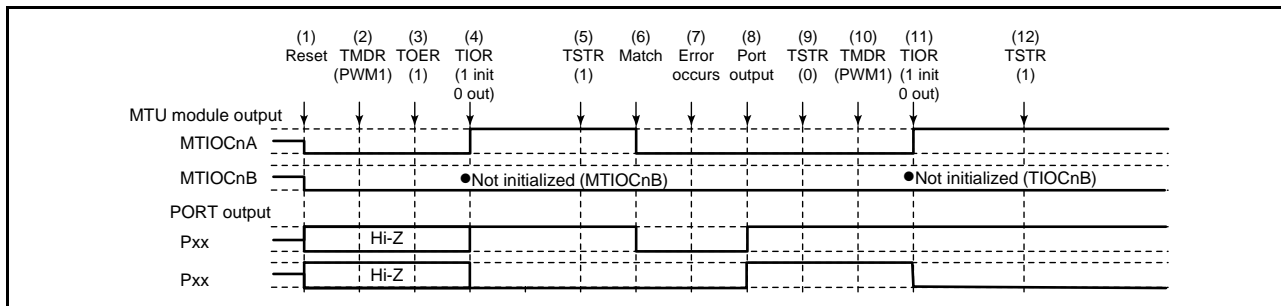


Figure 18.132 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 18.131.

(10) This step is not necessary when restarting in PWM mode 1.

(11) Initialize the pins with TIOR. (In PWM mode 1, the MTIOCnB side is not initialized.)

(12) Restart operation by setting TSTR.

(9) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

Figure 18.133 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

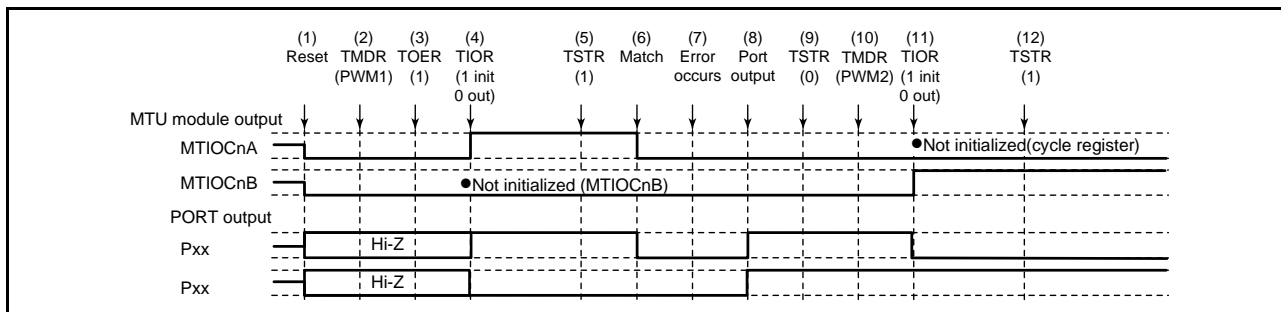


Figure 18.133 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 18.131.

(10) Set PWM mode 2.

(11) Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)

(12) Restart operation by setting TSTR.

Note: PWM mode 2 can only be selected for channels 0 to 2 (or 6 to 8), 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 18.134 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

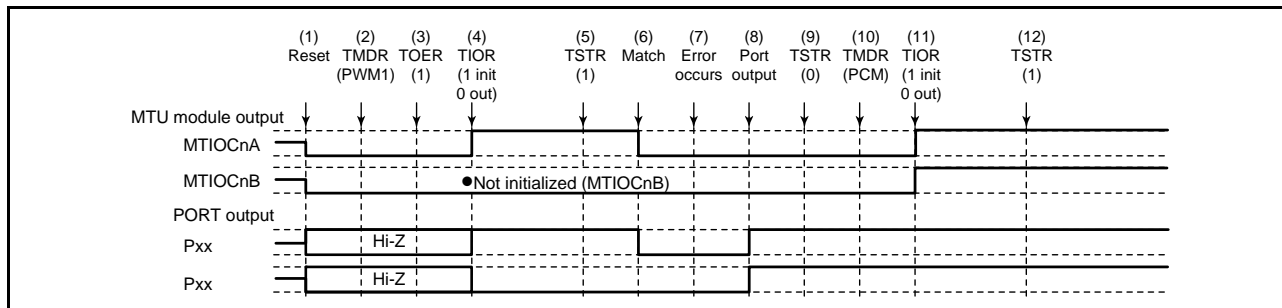


Figure 18.134 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 18.131.

(10) Set the phase counting mode.

(11) Initialize the pins with TIOR.

(12) Restart operation by setting TSTR.

Note: The phase counting mode can only be selected for channels 1 and 2 (or 7 and 8), 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 18.135 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

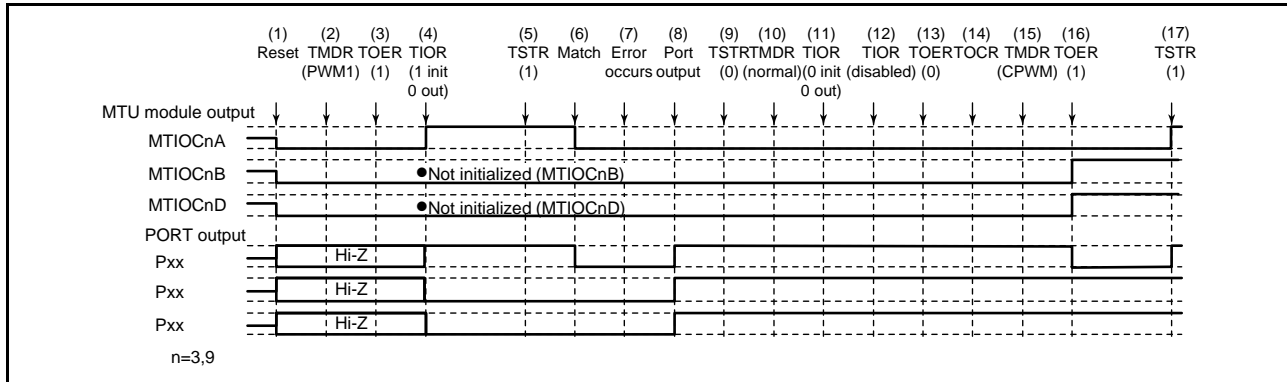


Figure 18.135 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

(1) to (9) are the same as in Figure 18.131.

(10) Set normal mode to initialize the normal mode waveform generation section.

(11) Initialize the PWM mode 1 waveform generation section with TIOR.

(12) Disable operation of the PWM mode 1 waveform generation section with TIOR

(13) Disable output in channels 3 and 4 (or channels 9 and 10) 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 channels 3 and 4 (or channels 9 and 10) with TOER.

(17) Restart operation by setting TSTR.

(12) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 18.136 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

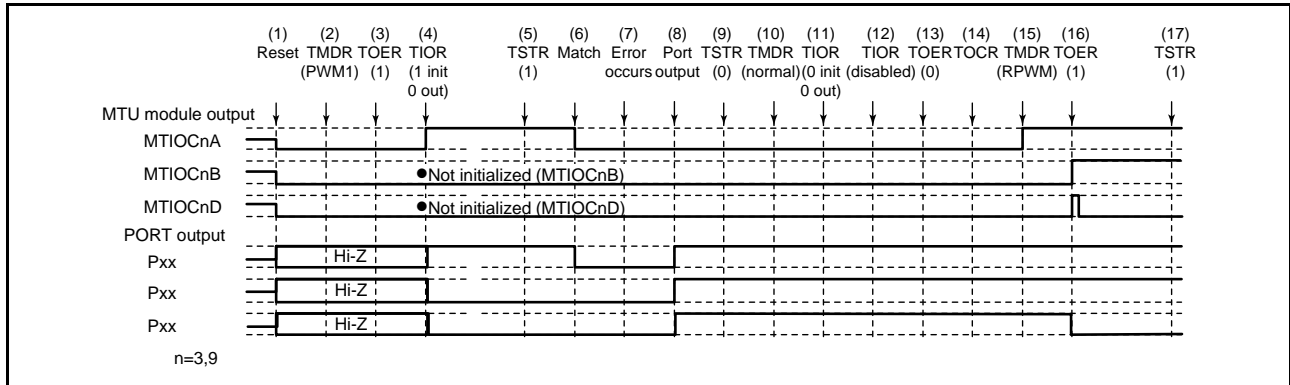


Figure 18.136 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

(1) to (13) are the same as in Figure 18.135.

(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 channels 3 and 4 (or channels 9 and 10) with TOER.

(17) Restart operation by setting TSTR.

(13) Operation when Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 18.137 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

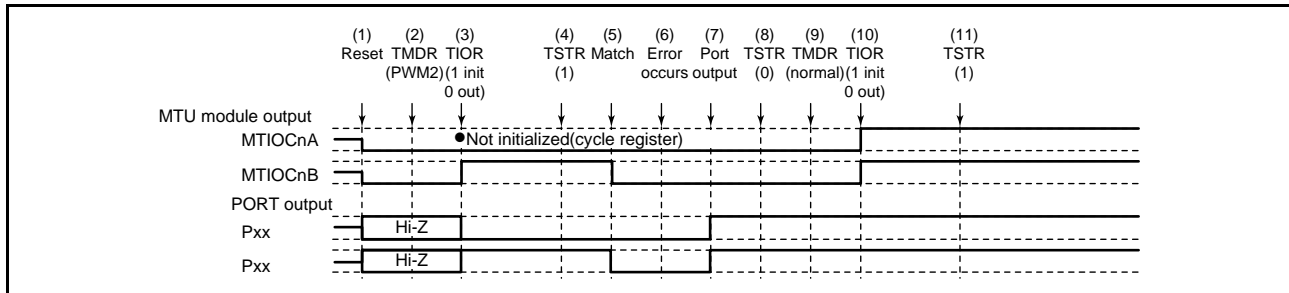


Figure 18.137 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

- (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 high, and 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) Start count operation by setting TSTR.
- (5) Output goes low on compare match occurrence.
- (6) An error occurs.
- (7) Disable the MTU output and select port output with TIOR and output the inverse of the active level.
- (8) Stop count operation by setting TSTR.
- (9) Set normal mode.
- (10) Initialize the pins with TIOR.
- (11) Restart operation by setting TSTR.

(14) Operation when Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 18.138 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

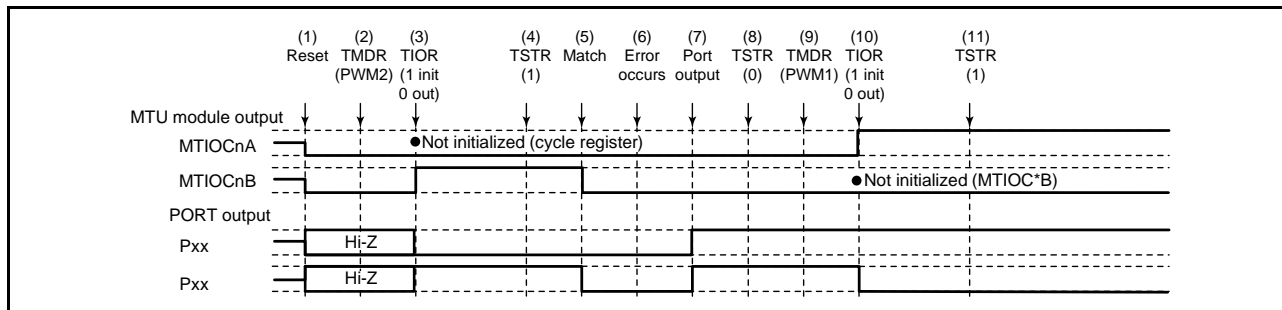


Figure 18.138 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

(1) to (8) are the same as in Figure 18.137.

(9) Set PWM mode 1.

(10) Initialize the pins with TIOR. (In PWM mode 1, the MTIOCnB side is not initialized.)

(11) Restart operation by setting TSTR.

(15) Operation when Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 18.139 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

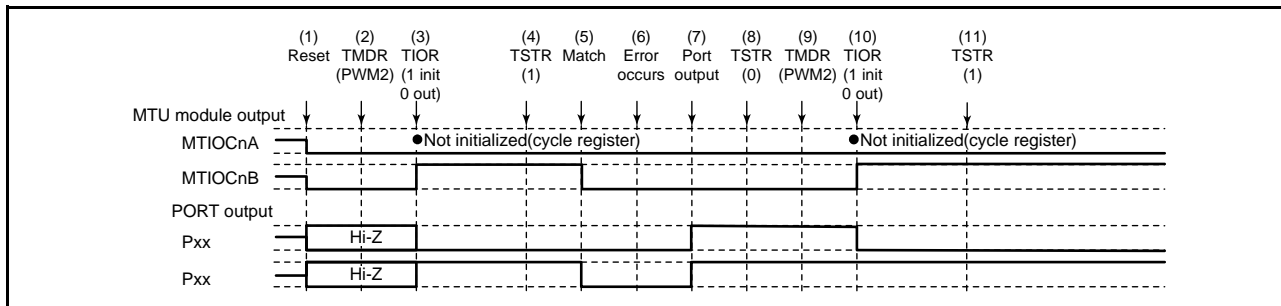


Figure 18.139 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

(1) to (8) are the same as in Figure 18.137.

(9) This step is not necessary when restarting in PWM mode 2.

(10) Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)

(11) Restart operation by setting TSTR.

(16) Operation when Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 18.140 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

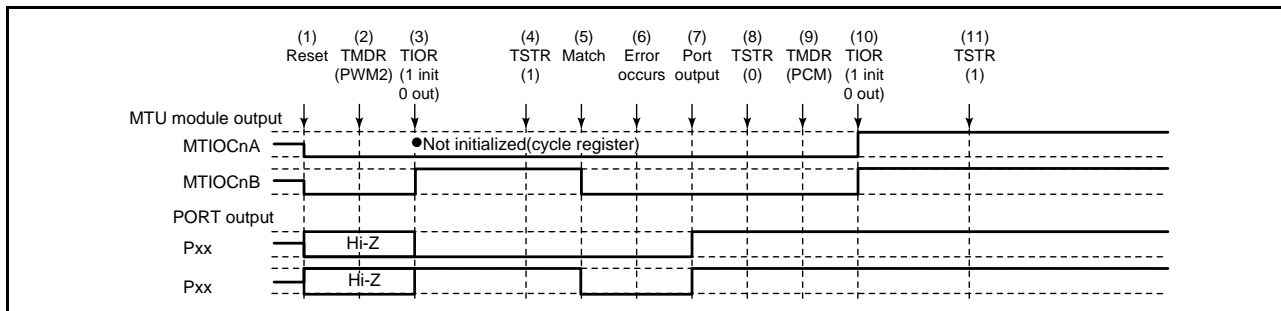


Figure 18.140 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

(1) to (8) are the same as in Figure 18.137.

(9) Set the phase counting mode.

(10) Initialize the pins with TIOR.

(11) Restart operation by setting TSTR.

(17) Operation when Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 18.141 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

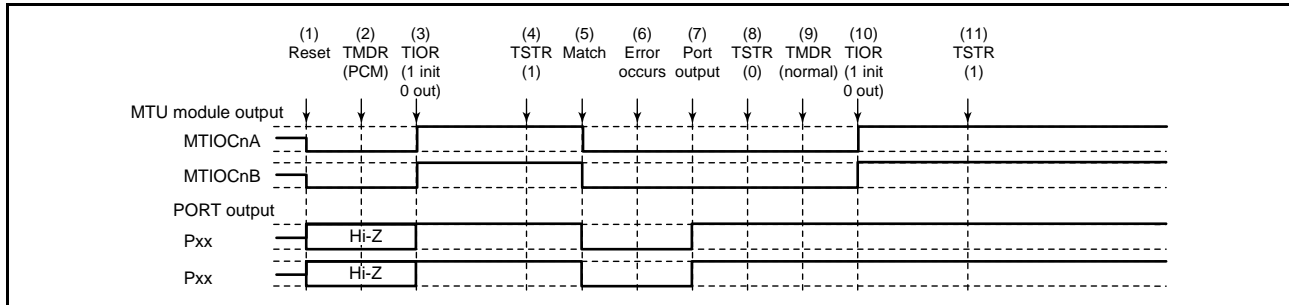


Figure 18.141 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

- (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 high, and low level is output on compare match occurrence.)
- (4) Start count operation by setting TSTR.
- (5) Output goes low on compare match occurrence.
- (6) An error occurs.
- (7) Select port output and output the inverse of the active level.
- (8) Stop count operation by setting TSTR.
- (9) Set normal mode.
- (10) Initialize the pins with TIOR.
- (11) Restart operation by setting TSTR.

(18) Operation when Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 18.142 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

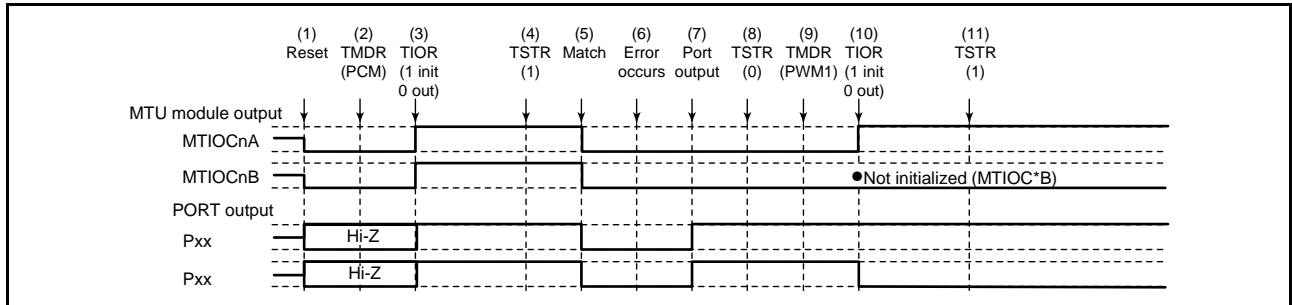


Figure 18.142 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

(1) to (8) are the same as in Figure 18.141.

(9) Set PWM mode 1.

(10) Initialize the pins with TIOR. (In PWM mode 1, the MTIOCnB side is not initialized.)

(11) Restart operation by setting TSTR.

(19) Operation when Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 18.143 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

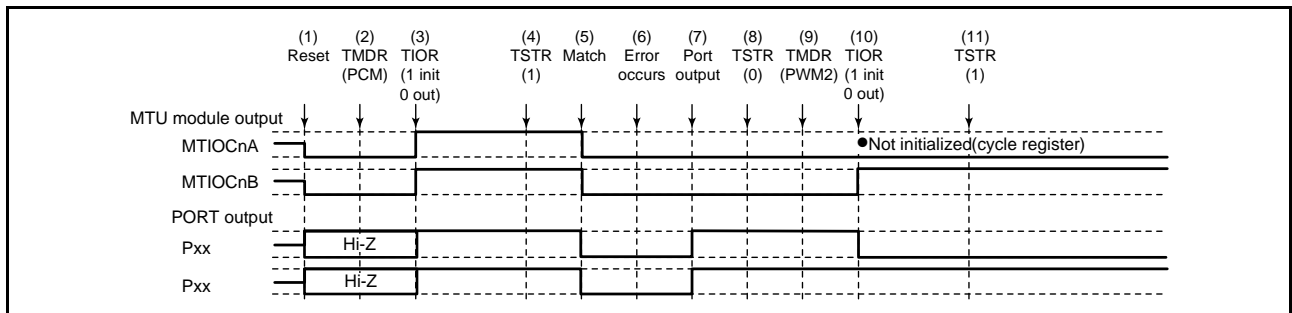


Figure 18.143 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

(1) to (8) are the same as in Figure 18.141.

(9) Set PWM mode 2.

(10) Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)

(11) Restart operation by setting TSTR.

(20) Operation when Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 18.144 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

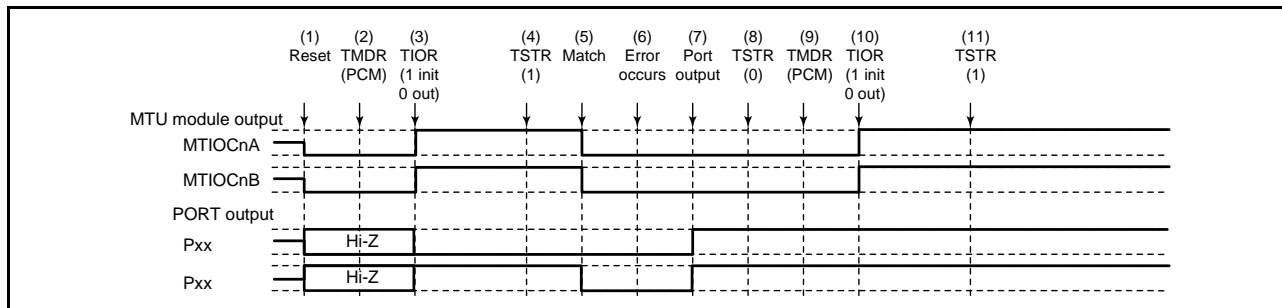


Figure 18.144 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

(1) to (8) are the same as in Figure 18.141.

(9) This step is not necessary when restarting in phase counting mode.

(10) Initialize the pins with TIOR.

(11) Restart operation by setting TSTR.

(21) Operation when Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 18.145 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

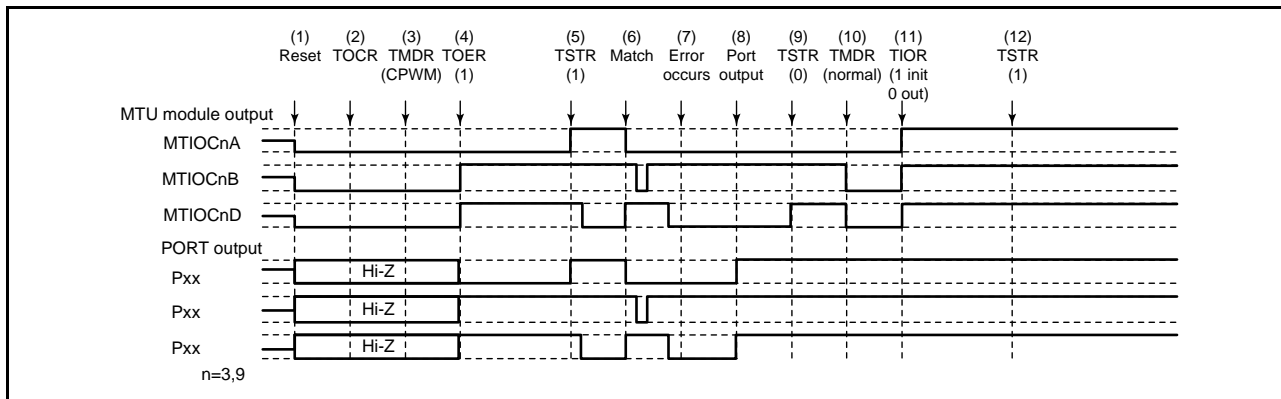


Figure 18.145 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

- (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 channels 3 and 4 (or channels 9 and 10) with TOER.
- (5) Start count operation by setting TSTR.
- (6) The complementary PWM waveform is output on compare match occurrence.
- (7) An error occurs.
- (8) Disable the MTU output and select port output with TIOR and output the inverse of the active level.
- (9) Stop count operation by setting TSTR. (MTU output becomes the initial complementary PWM output value).
- (10) Set normal mode (MTU output goes low).
- (11) Initialize the pins with TIOR.
- (12) Restart operation by setting TSTR.

(22) Operation when Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 18.146 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

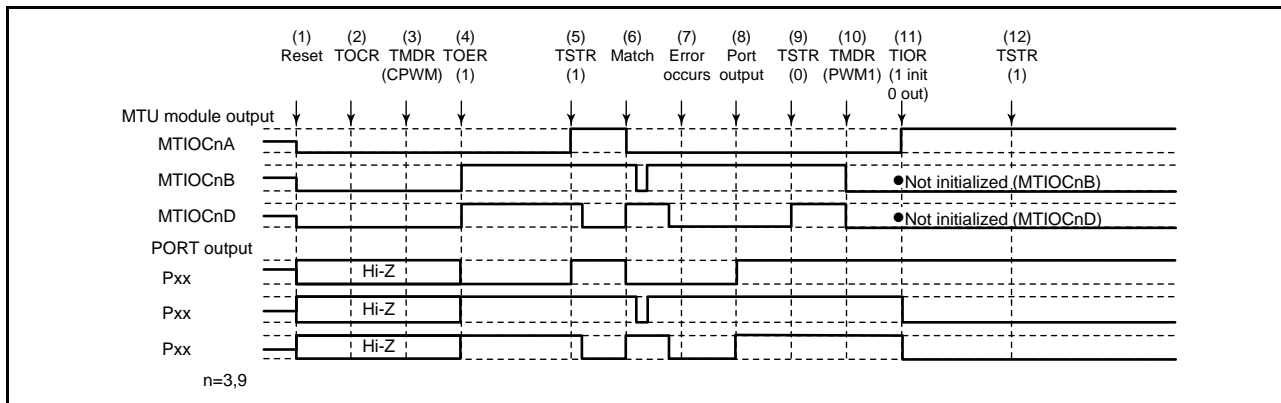


Figure 18.146 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 18.145.

(10) Set PWM mode 1 (MTU output goes low).

(11) Initialize the pins with TIOR. (In PWM mode 1, the MTIOCnB side is not initialized.)

(12) Restart operation by setting TSTR.

(23) Operation when Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 18.147 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 ratio at the time of stopping the counter).

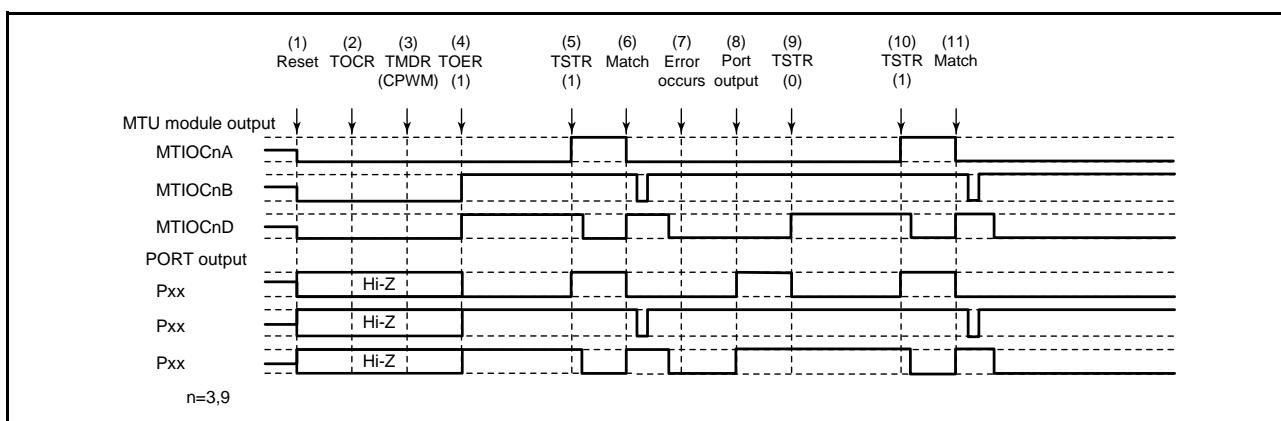


Figure 18.147 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

(1) to (9) are the same as in Figure 18.145.

(10) Restart operation by setting TSTR.

(11) 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 18.148 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 ratio).

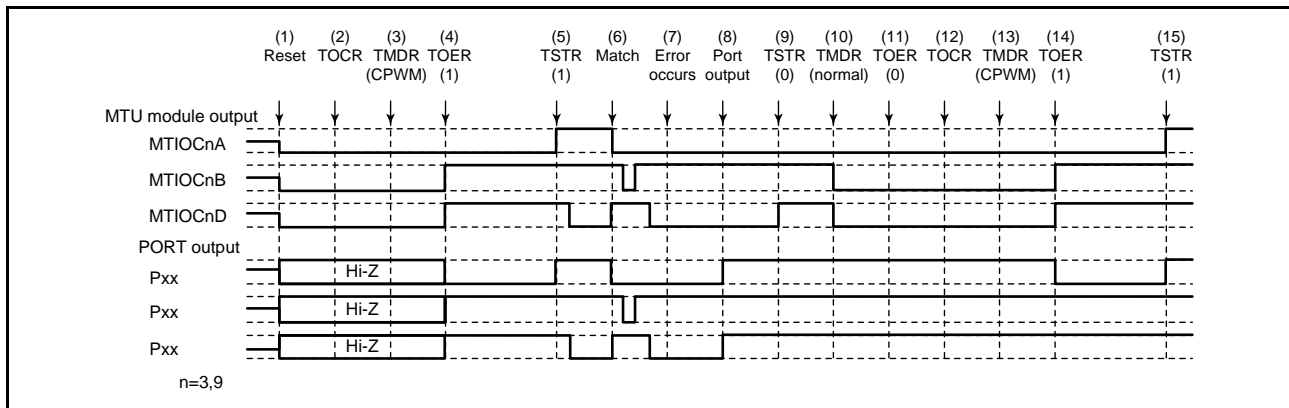


Figure 18.148 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

(1) to (9) are the same as in Figure 18.145.

(10) Set normal mode and make new settings (MTU output goes low).

(11) Disable output in channels 3 and 4 (or channels 9 and 10) with TOER.

(12) Select the complementary PWM output level and enable or disable cyclic output with TOCR.

(13) Set complementary PWM mode.

(14) Enable output in channels 3 and 4 (or channels 9 and 10) with TOER.

(15) Restart operation by setting TSTR.

(25) Operation when Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 18.149 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

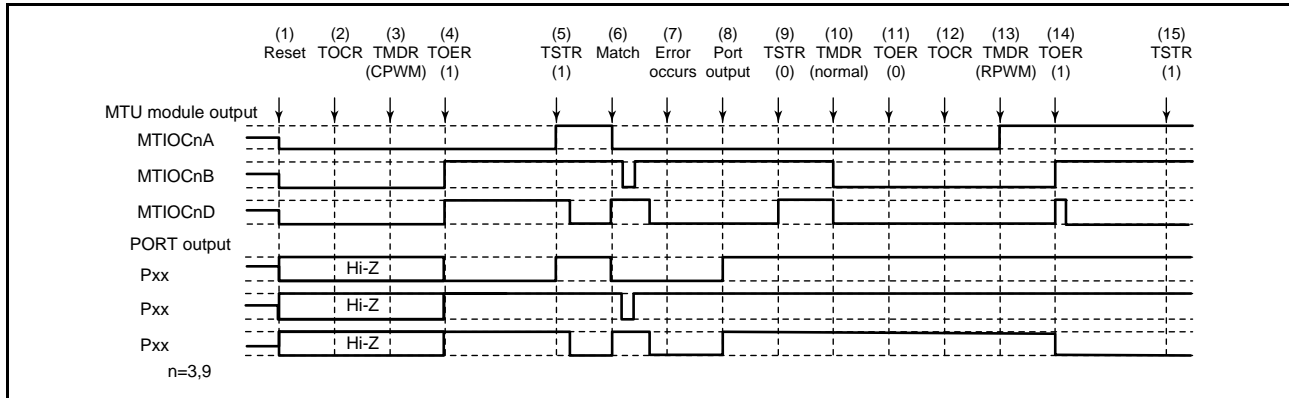


Figure 18.149 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (9) are the same as in Figure 18.145.

(10) Set normal mode (MTU output goes low).

(11) Disable output in channels 3 and 4 (or channels 9 and 10) with TOER.

(12) Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.

(13) Set reset-synchronized PWM mode.

(14) Enable output in channels 3 and 4 (or channels 9 and 10) with TOER.

(15) Restart operation by setting TSTR.

(26) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 18.150 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

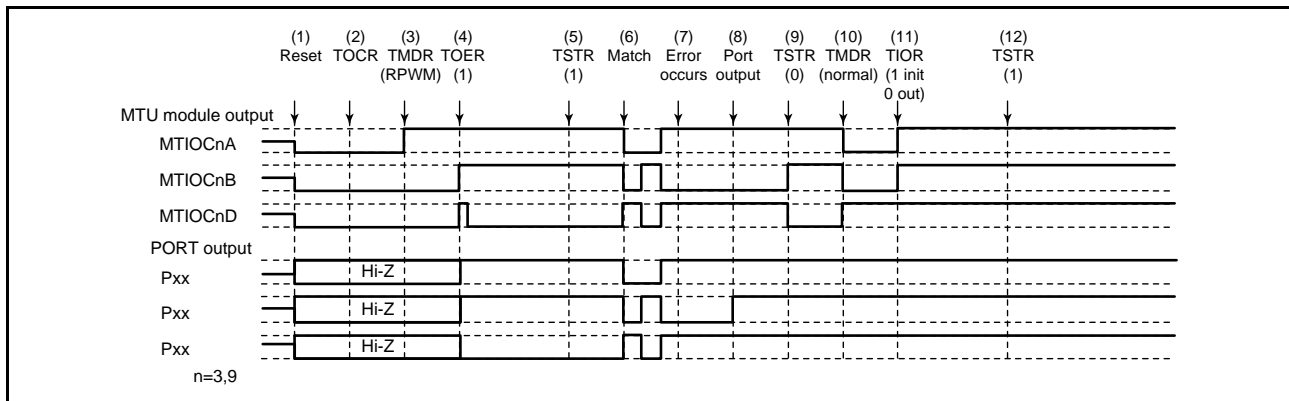


Figure 18.150 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

- (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 channels 3 and 4 (or channels 9 and 10) with TOER.
- (5) Start count operation by setting TSTR.
- (6) The reset-synchronized PWM waveform is output on compare match occurrence.
- (7) An error occurs.
- (8) Disable the MTU output and select port output with TIOR and output the inverse of the active level.
- (9) Stop count operation by setting TSTR. (MTU output becomes the initial reset-synchronized PWM output value.)
- (10) Set normal mode (positive-phase MTU output goes low, and negative-phase output goes high).
- (11) Initialize the pins with TIOR.
- (12) Restart operation by setting TSTR.

(27) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 18.151 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

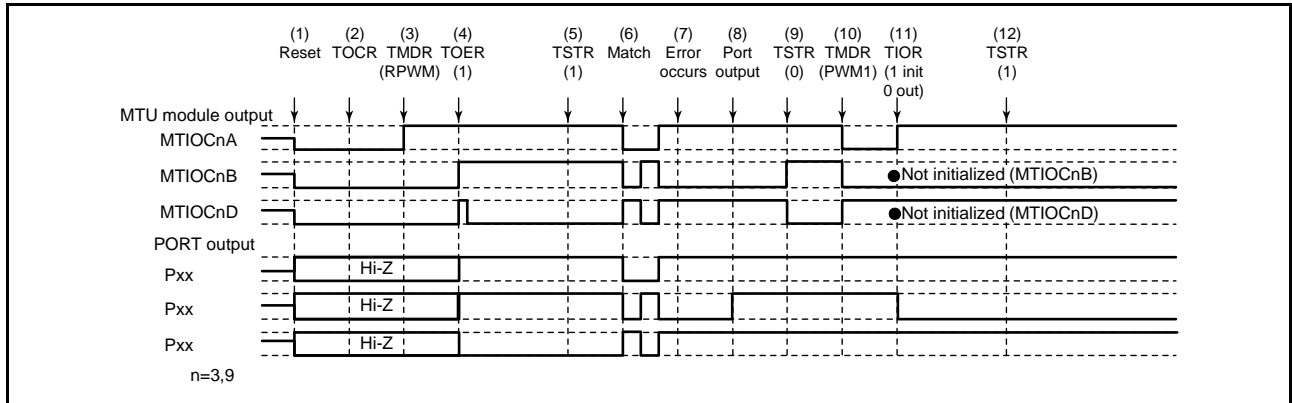


Figure 18.151 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 18.150.

(10) Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).

(11) Initialize the pins with TIOR. (In PWM mode 1, the MTIOCnB side is not initialized.)

(12) Restart operation by setting TSTR.

(28) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 18.152 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

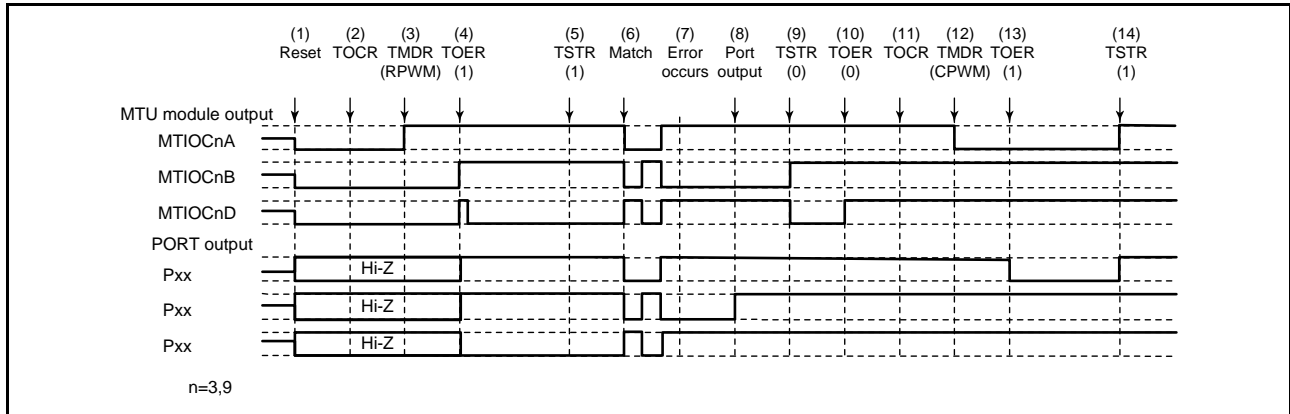


Figure 18.152 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

(1) to (9) are the same as in Figure 18.150.

(10) Disable output in channels 3 and 4 (or channels 9 and 10) with TOER

(11) Select the complementary PWM output level and enable or disable cyclic output with TOCR.

(12) Set complementary PWM mode (MTU cyclic output pin goes low).

(13) Enable output in channels 3 and 4 (or channels 9 and 10) with TOER.

(14) 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 18.153 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.

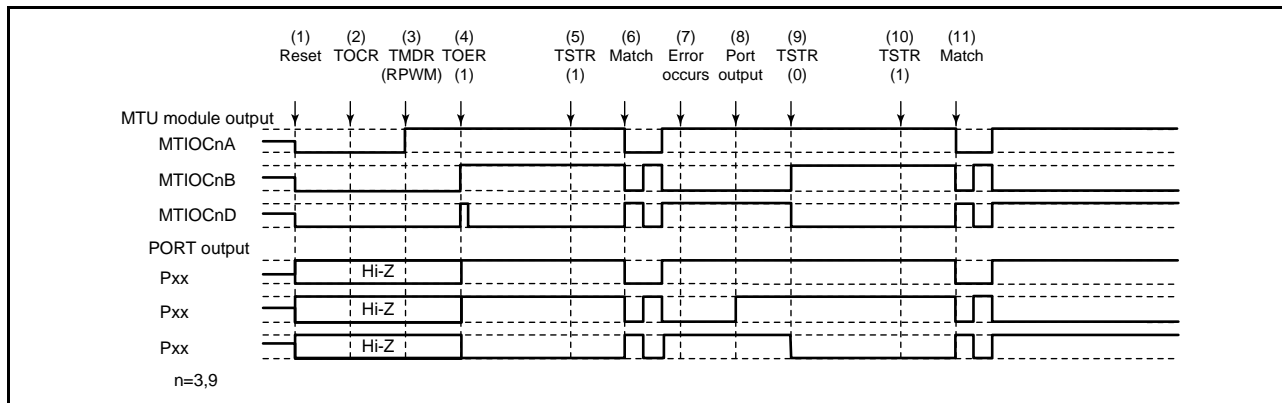


Figure 18.153 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (9) are the same as in Figure 18.150.

(10) Restart operation by setting TSTR.

(11) The reset-synchronized PWM waveform is output on compare match occurrence.

19. Port Output Enable 2 (POE2)

The port output enable (POE) can be used to place the pins multiplexed with the MTU complementary PWM output pins and the pins multiplexed with MTU0 or MTU6 pins (MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D; or MTIOC6A, MTIOC6B, MTIOC6C, and MTIOC6D) in high-impedance state, depending on the input change on POE0# to POE9# pins and the output status of the pins multiplexed with the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC9B, MTIOC9D, MTIOC10A, MTIOC10B, MTIOC10C, and MTIOC10D), or by modifying register settings. It can also simultaneously generate interrupt requests.

19.1 Overview

Table 19.1 lists the specifications of the POE, and Figure 19.1 shows a block diagram of the POE.

Table 19.1 POE Specifications

Item	Description
Function	<ul style="list-style-type: none"> Each of the POE0# to POE9# input pins can be set for falling edge, $PCLK/8 \times 16$, $PCLK/16 \times 16$, or $PCLK/128 \times 16$ low-level sampling. Pins for the MTU complementary PWM output and MTU0 or MTU6 can be placed in high-impedance state by POE0# to POE9# pin falling-edge or low-level sampling. Pins for the MTU complementary PWM output and MTU0 or MTU6 can be placed in high-impedance state when the oscillation-stop detection circuit in the clock pulse generator detects stopped oscillation. Pins for the MTU can be placed in high-impedance state when output levels of the MTU complementary PWM output pins are compared and simultaneous active-level output continues for one cycle or more. Pins for the MTU complementary PWM output and MTU0 or MTU6 can be placed in high-impedance state by modifying the POE register settings. Interrupts can be generated by input-level sampling or output-level comparison results.

The POE has input level detection circuits, output level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in Figure 19.1. In addition to control by the POE, pins for complementary PWM output and MTU0 or MTU6 can be placed in high-impedance state when the oscillator stops.

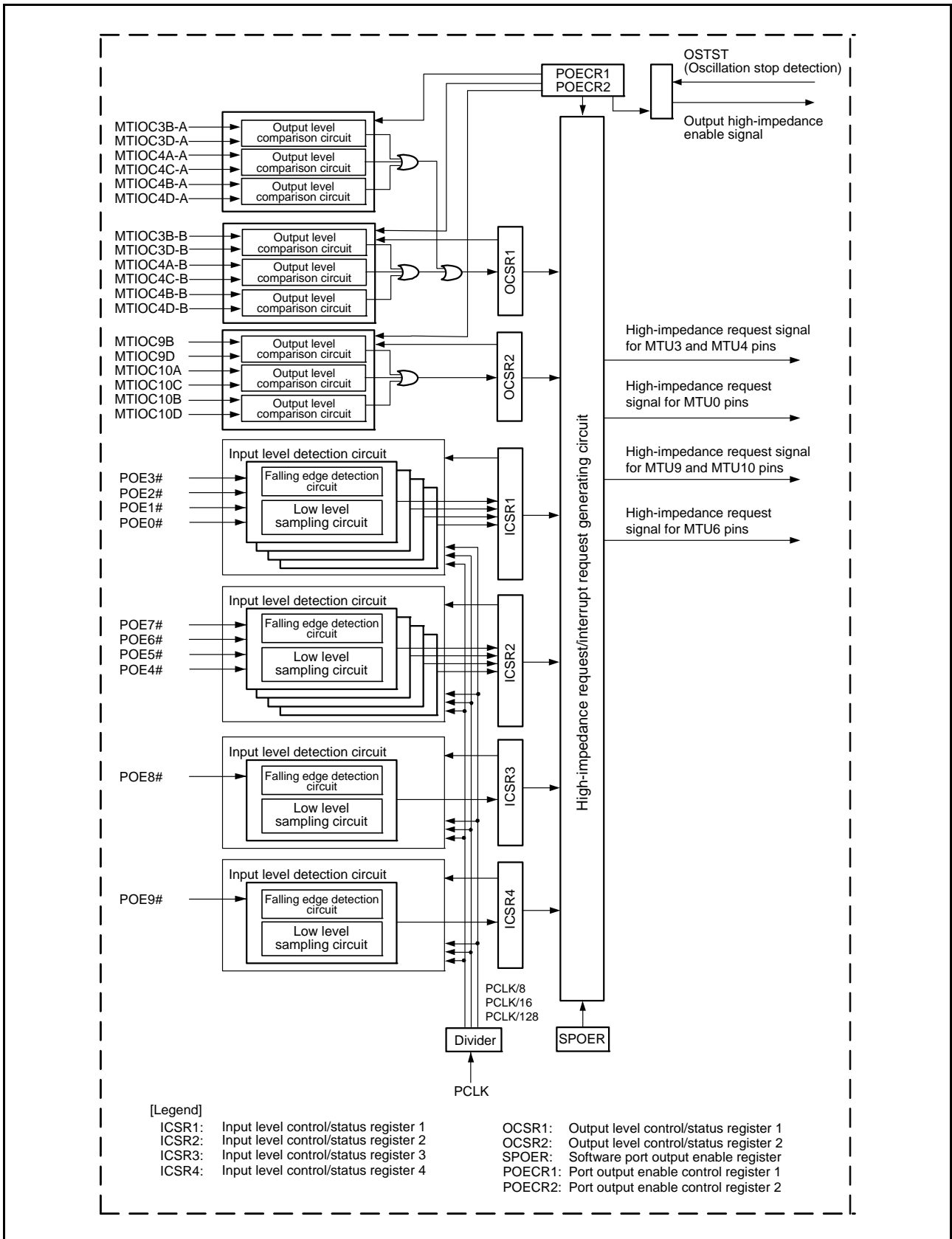


Figure 19.1 POE Block Diagram

Table 19.2 shows input/output pins to be used by the POE.

Table 19.2 POE Input/Output Pins

Pin Name	I/O	Description
POE0# to POE3#	Input	Inputs request signals to place the MTU3 and MTU4 pins for MTU complementary PWM output in high-impedance state.
POE4# to POE7#	Input	Inputs request signals to place the MTU9 and MTU10 pins for MTU complementary PWM output in high-impedance state.
POE8#	Input	Inputs request signals to place the pins for MTU0 in high-impedance state.
POE9#	Input	Inputs request signals to place the pins for MTU6 in high-impedance state.

Table 19.3 shows output-level comparisons with pin combinations.

Table 19.3 Pin Combinations

Pin Combination	I/O	Description
MTIOC3B-A and MTIOC3D-A	Output	The MTU3 and MTU4 pins for MTU complementary PWM output are placed in high-impedance state when the pins simultaneously output an active level (low when the OLSP bit in TOCR1 of the MTUA is 0 with the TOCS bit in TOCR1 of the MTUA cleared to 0 or high when the OLSP bit is 1, or low when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits in TOCR2 of the MTUA are 0 with the TOCS bit in TOCR1 of the MTUA set to 1 or high when these bits are 1) for one or more cycles of the peripheral clock (PCLK).
MTIOC4A-A and MTIOC4C-A	Output	
MTIOC4B-A and MTIOC4D-A	Output	
MTIOC3B-B and MTIOC3D-B	Output	
MTIOC4A-B and MTIOC4C-B	Output	Pin combinations for output comparison and high-impedance control can be selected by POE registers.
MTIOC4B-B and MTIOC4D-B	Output	
MTIOC9B and MTIOC9D	Output	The MTU9 and MTU10 pins for MTU complementary PWM output are placed in high-impedance state when the pins simultaneously output an active level (low when the OLSP bit in TOCR1 of the MTUB is 0 with the TOCS bit in TOCR1 of the MTUB cleared to 0 or high when the OLSP bit is 1, or low when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits in TOCR2 of the MTUB are 0 with the TOCS bit in TOCR1 of the MTUB set to 1 and high when these bits are 1) for one or more cycles of the peripheral clock (PCLK).
MTIOC10A and MTIOC10C	Output	
MTIOC10B and MTIOC10D	Output	

19.2 Register Descriptions

Table 19.4 is the list of POE registers. The POE registers are initialized by a reset.

Table 19.4 POE Register Configuration

Register Name	Symbol	Value after Reset	Address	Access Size
Input level control/status register 1	ICSR1	0000h	0008 8900h	16
Output level control/status register 1	OCSR1*	0000h	0008 8902h	16
Input level control/status register 2	ICSR2	0000h	0008 8904h	16
Output level control/status register 2	OCSR2	0000h	0008 8906h	16
Input level control/status register 3	ICSR3	0000h	0008 8908h	16
Software port output enable register	SPOER	00h	0008 890Ah	8
Port output enable control register 1	POECR1	00h	0008 890Bh	8
Port output enable control register 2	POECR2	7070h	0008 890Ch	16
Input level control/status register 4	ICSR4	0000h	0008 890Eh	16

Note : * In the 100-pin LQFP version, do not set the OCSR1.OCE1 bit to 1. Otherwise, the operation is not guaranteed.

19.2.1 Input Level Control/Status Register 1 (ICSR1)

Address: 0008 8900h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POE3F	POE2F	POE1F	POE0F	—	—	—	PIE1	POE3M[1:0]	POE2M[1:0]	POE1M[1:0]	POE0M[1:0]				
Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE0M[1:0]	POE0 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE0# input. 0 1: Accepts a request when POE0# input has been sampled 16 times at PCLK/8 clock pulses and all are low. 1 0: Accepts a request when POE0# input has been sampled 16 times at PCLK/16 clock pulses and all are low. 1 1: Accepts a request when POE0# input has been sampled 16 times at PCLK/128 clock pulses and all are low.	R/W*1
b3, b2	POE1M[1:0]	POE1 Mode Select	b3 b2 0 0: Accepts a request on the falling edge of POE1# input. 0 1: Accepts a request when POE1# input has been sampled 16 times at PCLK/8 clock pulses and all are low. 1 0: Accepts a request when POE1# input has been sampled 16 times at PCLK/16 clock pulses and all are low. 1 1: Accepts a request when POE1# input has been sampled 16 times at PCLK/128 clock pulses and all are low.	R/W*1
b5, b4	POE2M[1:0]	POE2 Mode Select	b5 b4 0 0: Accepts a request on the falling edge of POE2# input. 0 1: Accepts a request when POE2# input has been sampled 16 times at PCLK/8 clock pulses and all are low. 1 0: Accepts a request when POE2# input has been sampled 16 times at PCLK/16 clock pulses and all are low. 1 1: Accepts a request when POE2# input has been sampled 16 times at PCLK/128 clock pulses and all are low.	R/W*1
b7, b6	POE3M[1:0]	POE3 Mode Select	b7 b6 0 0: Accepts a request on the falling edge of POE3# input. 0 1: Accepts a request when POE3# input has been sampled 16 times at PCLK/8 clock pulses and all are low. 1 0: Accepts a request when POE3# input has been sampled 16 times at PCLK/16 clock pulses and all are low. 1 1: Accepts a request when POE3# input has been sampled 16 times at PCLK/128 clock pulses and all are low.	R/W*1
b8	PIE1	Port Interrupt Enable 1	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b11 to b9	—	Reserved	These bits are always 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

Can be modified only once after a reset.

Note 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

ICSR1 selects the input modes for the POE0# to POE3# pins, controls the enable/disable of interrupts, and indicates status.

POE0M[1:0] Bits (POE0 Mode Select)

These bits select the input mode of the POE0# pin.

POE1M[1:0] Bits (POE1 Mode Select)

These bits select the input mode of the POE1# pin.

POE2M[1:0] Bits (POE2 Mode Select)

These bits select the input mode of the POE2# pin.

POE3M[1:0] Bits (POE3 Mode Select)

These bits select the input mode of the POE3# pin.

PIE1 Bit (Port Interrupt Enable 1)

This bit enables or disables interrupt requests when any one of the POE0F to POE3F bits of the ICSR1 is set to 1.

POE0F Flag (POE0 Flag)

This flag indicates that a high impedance request has been input to the POE0# pin.

[Clearing conditions]

- By writing 0 to POE0F after reading POE0F = 1

[Setting condition]

- When the input set by POE0M[1:0] occurs at the POE0# pin

POE1F Flag (POE1 Flag)

This flag indicates that a high impedance request has been input to the POE1# pin.

[Clearing conditions]

- By writing 0 to POE1F after reading POE1F = 1

[Setting condition]

- When the input set by POE1M[1:0] occurs at the POE1# pin

POE2F Flag (POE2 Flag)

This flag indicates that a high impedance request has been input to the POE2# pin.

[Clearing conditions]

- By writing 0 to POE2F after reading POE2F = 1

[Setting condition]

- When the input set by POE2M[1:0] occurs at the POE2# pin

POE3F Flag (POE3 Flag)

This flag indicates that a high impedance request has been input to the POE3# pin.

[Clearing conditions]

- By writing 0 to POE3F after reading POE3F = 1

[Setting condition]

- When the input set by POE3M[1:0] occurs at the POE3# pin

19.2.2 Output Level Control/Status Register 1 (OCSR1)

Address: 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 always read as 0. The write value should be 0.	RW
b8	OIE1	Output Short Interrupt Enable 1	0: Interrupt requests disabled. 1: Interrupt requests enabled.	R/W
b9	OCE1	Output Short High-Impedance Enable 1	0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.	R/W*1
b14 to b10	—	Reserved	These bits are always read as 0. The write value should be 0.	RW
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.

Note 3. In the 100-pin LQFP version, do not set the OCSR1.OCE1 bit to 1. Otherwise, the operation is not guaranteed.

OCSR1 controls the enable/disable of output level comparison and interrupts, and indicates status.

OIE1 Bit (Output Short Interrupt Enable 1)

This bit enables or disables interrupt requests when the OSF1 bit in OCSR1 is set to 1.

OCE1 Bit (Output Short High-Impedance Enable 1)

This bit specifies whether to place the pins in high-impedance state when the OSF1 bit in OCSR1 is set to 1.

OSF1 Flag (Output Short Flag 1)

This flag indicates that any one of the three pairs of two-phase MTU3 and MTU4 outputs for MTU complementary PWM output to be compared has simultaneously become an active level.

[Clearing condition]

- By writing 0 to OSF1 after reading OSF1 = 1

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level

19.2.3 Input Level Control/Status Register 2 (ICSR2)

Address: 0008 8904h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POE7F	POE6F	POE5F	POE4F	—	—	—	PIE2	POE7M[1:0]	POE6M[1:0]	POE5M[1:0]	POE4M[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	POE4M[1:0]	POE4 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE4# input. 0 1: Accepts a request when POE4# input has been sampled 16 times at PCLK/8 clock pulses and all are low. 1 0: Accepts a request when POE4# input has been sampled 16 times at PCLK/16 clock pulses and all are low. 1 1: Accepts a request when POE4# input has been sampled 16 times at PCLK/128 clock pulses and all are low.	R/W*1
b3, b2	POE5M[1:0]	POE5 Mode Select	b3 b2 0 0: Accepts a request on the falling edge of POE5# input. 0 1: Accepts a request when POE5# input has been sampled 16 times at PCLK/8 clock pulses and all are low. 1 0: Accepts a request when POE5# input has been sampled 16 times at PCLK/16 clock pulses and all are low. 1 1: Accepts a request when POE5# input has been sampled 16 times at PCLK/128 clock pulses and all are low.	R/W*1
b5, b4	POE6M[1:0]	POE6 Mode Select	b5 b4 0 0: Accepts a request on the falling edge of POE6# input. 0 1: Accepts a request when POE6# input has been sampled 16 times at PCLK/8 clock pulses and all are low. 1 0: Accepts a request when POE6# input has been sampled 16 times at PCLK/16 clock pulses and all are low. 1 1: Accepts a request when POE6# input has been sampled 16 times at PCLK/128 clock pulses and all are low.	R/W*1
b7, b6	POE7M[1:0]	POE7 Mode Select	b7 b6 0 0: Accepts a request on the falling edge of POE7# input. 0 1: Accepts a request when POE7# input has been sampled 16 times at PCLK/8 clock pulses and all are low. 1 0: Accepts a request when POE7# input has been sampled 16 times at PCLK/16 clock pulses and all are low. 1 1: Accepts a request when POE7# input has been sampled 16 times at PCLK/128 clock pulses and all are low.	R/W*1
b8	PIE2	Port Interrupt Enable 2	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b11 to b9	—	Reserved	These bits are always read as 0. The write value should be 0.	RW
b12	POE4F	POE4 Flag	0: Indicates that a high impedance request has not been input to the POE4# pin. 1: Indicates that a high impedance request has been input to the POE4# pin.	R/(W)*2
b13	POE5F	POE5 Flag	0: Indicates that a high impedance request has not been input to the POE5# pin. 1: Indicates that a high impedance request has been input to the POE5# pin.	R/(W)*2
b14	POE6F	POE6 Flag	0: Indicates that a high impedance request has not been input to the POE6# pin. 1: Indicates that a high impedance request has been input to the POE6# pin.	R/(W)*2
b15	POE7F	POE7 Flag	0: Indicates that a high impedance request has not been input to the POE7# pin. 1: Indicates that a high impedance request has been input to the POE7# 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.

ICSR2 selects the input modes for the POE4# to POE7# pins, controls the enable/disable of interrupts, and indicates status.

POE4M[1:0] Bits (POE4 Mode Select)

These bits select the input mode of the POE4# pin.

POE5M[1:0] Bits (POE5 Mode Select)

These bits select the input mode of the POE5# pin.

POE6M[1:0] Bits (POE6 Mode Select)

These bits select the input mode of the POE6# pin.

POE7M[1:0] Bits (POE7 Mode Select)

These bits select the input mode of the POE7# pin.

PIE2 Bit (Port Interrupt Enable 2)

This bit enables/disables interrupt requests when any one of the POE4F to POE7F bits of the ICSR2 is set to 1.

POE4F Flag (POE4 Flag)

This flag indicates that a high impedance request has been input to the POE4# pin.

[Clearing conditions]

- By writing 0 to POE4F after reading POE4F = 1

[Setting condition]

- When the input set by POE4M[1:0] occurs at the POE4# pin

POE5F Flag (POE5 Flag)

This flag indicates that a high impedance request has been input to the POE5# pin.

[Clearing conditions]

- By writing 0 to POE5F after reading POE5F = 1

[Setting condition]

- When the input set by POE5M[1:0] occurs at the POE5# pin

POE6F Flag (POE6 Flag)

This flag indicates that a high impedance request has been input to the POE6# pin.

[Clearing conditions]

- By writing 0 to POE6F after reading POE6F = 1

[Setting condition]

- When the input set by POE6M[1:0] occurs at the POE6# pin

POE7F Flag (POE7 Flag)

This flag indicates that a high impedance request has been input to the POE7# pin.

[Clearing conditions]

- By writing 0 to POE7F after reading POE7F = 1

[Setting condition]

- When the input set by POE7M[1:0] occurs at the POE7# pin

19.2.4 Output Level Control/Status Register 2 (OCSR2)

Address: 0008 8906h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSF2	—	—	—	—	—	OCE2	OIE2	—	—	—	—	—	—	—	—
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 always read as 0. The write value should be 0.	RW
b8	OIE2	Output Short Interrupt Enable 2	0: Interrupt requests disabled. 1: Interrupt requests enabled.	R/W
b9	OCE2	Output Short High-Impedance Enable 2	0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.	R/W*1
b14 to b10	—	Reserved	These bits are always read as 0. The write value should be 0.	RW
b15	OSF2	Output Short Flag 2	0: Indicates that outputs have not simultaneously become an active level. 1: Indicates that outputs have simultaneously become an active level.	R/(W)*2

Note 1. Can be modified only once after a reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

OCSR2 controls the enable/disable of output level comparison and interrupts, and indicates status.

OIE2 Bit (Output Short Interrupt Enable 1)

This bit enables or disables interrupt requests when the OSF2 bit in OCSR2 is set to 1.

OCE2 Bit (Output Short High-Impedance Enable 2)

This bit specifies whether to place the pins in high-impedance state when the OSF2 bit in OCSR2 is set to 1.

OSF2 Flag (Output Short Flag 2)

This flag indicates that any one of the three pairs of two-phase MTU9 and MTU10 outputs for MTU complementary PWM output to be compared has simultaneously become an active level.

[Clearing condition]

- By writing 0 to OSF2 after reading OSF2 = 1

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level

19.2.5 Input Level Control/Status Register 3 (ICSR3)

Address: 0008 8908h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	POE8F	—	—	POE8E	PIE3	—	—	—	—	—	—	—	POE8M[1:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE8M[1:0]	POE8 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE8# input 0 1: Accepts a request when POE8# input has been sampled 16 times at PCLK/8 clock pulses and all are low. 1 0: Accepts a request when POE8# input has been sampled 16 times at PCLK/16 clock pulses and all are low. 1 1: Accepts a request when POE8# input has been sampled 16 times at PCLK/128 clock pulses and all are low.	R/W*1
b7 to b2	—	Reserved	These bits are always read as 0. The write value should be 0.	RW
b8	PIE3	Port Interrupt Enable 3	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE8E	POE8 High-Impedance Enable	0: Does not place the MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D pins in high-impedance state. 1: Places the MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D pins in high-impedance state.	R/W*1
b11, b10	—	Reserved	These bits are always read as 0. The write value should be 0.	RW
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 always read as 0. The write value should be 0.	RW

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.

ICSR3 selects the POE8# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE8M[1:0] Bits (POE8 Mode Select)

These bits select the input mode of the POE8# pin.

PIE3 Bit (Port Interrupt Enable 3)

This bit enables or disables interrupt requests when the POE8F bit is set to 1.

POE8E Bit (POE8 High-Impedance Enable)

This bit specifies whether to place the corresponding pin in high-impedance state when the POE8F bit is set to 1.

POE8F Flag (POE8 Flag)

This flag indicates that a high impedance request has been input to the POE8# pin.

[Clearing condition]

- By writing 0 to POE8F after reading POE8F = 1

[Setting condition]

- When the input set by POE8M[1:0] occurs at the POE8# pin

19.2.6 Input Level Control/Status Register 4 (ICSR4)

Address: 0008 890Eh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	POE9F	—	—	POE9E	PIE4	—	—	—	—	—	—	POE9M[1:0]	—
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
b1, b0	POE9M[1:0]	POE9 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE9# input 0 1: Accepts a request when POE9# input has been sampled 16 times at PCLK/8 clock pulses and all are low. 1 0: Accepts a request when POE9# input has been sampled 16 times at PCLK/16 clock pulses and all are low. 1 1: Accepts a request when POE9# input has been sampled 16 times at PCLK/128 clock pulses and all are low.	R/W*1
b7 to b2	—	Reserved	These bits are always read as 0. The write value should be 0.	RW
b8	PIE4	Port Interrupt Enable 4	0: Interrupt requests disabled. 1: Interrupt requests enabled.	R/W
b9	POE9E	POE9 High-Impedance Enable	0: Does not place the MTIOC6A, MTIOC6B, MTIOC6C, and MTIOC6D pins in high-impedance state. 1: Places the MTIOC6A, MTIOC6B, MTIOC6C, and MTIOC6D pins in high-impedance state.	R/W*1
b11, b10	—	Reserved	These bits are always read as 0. The write value should be 0.	RW
b12	POE9F	POE9 Flag	0: Indicates that a high impedance request has not been input to the POE9# pin. 1: Indicates that a high impedance request has been input to the POE9# pin.	R/(W)*2
b15 to b13	—	Reserved	These bits are always read as 0. The write value should be 0.	RW

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.

ICSR4 selects the POE9# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE9M[1:0] Bits (POE9 Mode Select)

These bits select the input mode of the POE9# pin.

PIE4 Bit (Port Interrupt Enable 4)

This bit enables or disables interrupt requests when the POE9F bit is set to 1.

POE9E Bit (POE9 High-Impedance Enable)

This bit specifies whether to place the corresponding pin in high-impedance state when the POE9F bit is set to 1.

POE9F Flag (POE9 Flag)

This flag indicates that a high impedance request has been input to the POE9# pin.

[Clearing condition]

- By writing 0 to POE9F after reading POE9F = 1

[Setting condition]

- When the input set by POE9M[1:0] occurs at the POE9# pin

19.2.7 Software Port Output Enable Register (SPOER)

Address: 0008 890Ah

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	CH6HIZ	CH910HIZ	CH0HIZ	CH34HIZ

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 state. 1: Places the pins in high-impedance state.	R/W
b1	CH0HIZ	MTU0 Output High-Impedance Enable	0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.	R/W
b2	CH910HIZ	MTU9 and MTU10 Output High-Impedance Enable	0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.	R/W
b3	CH6HIZ	MTU6 Output High-Impedance Enable	0: Does not place the pins in high-impedance state. 1: Places the pins in high-impedance state.	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should be 0.	RW

SPOER controls high-impedance state of the pins.

CH34HIZ Bit (MTU3 and MTU4 Output High-Impedance Enable)

This bit specifies whether to place the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) in high-impedance state.

[Clearing conditions]

- Reset
- By writing 0 to CH34HIZ after reading CH34HIZ = 1

[Setting condition]

- By writing 1 to CH34HIZ

CH0HIZ Bit (MTU0 Output High-Impedance Enable)

This bit specifies whether to place the MTU0 pin in high-impedance state.

[Clearing conditions]

- Reset
- By writing 0 to CH0HIZ after reading CH0HIZ = 1

[Setting condition]

- By writing 1 to CH0HIZ

CH910HIZ Bit (MTU9 and MTU10 Output High-Impedance Enable)

This bit specifies whether to place the MTU complementary PWM output pins (MTIOC9B, MTIOC9D, MTIOC10A, MTIOC10B, MTIOC10C, MTIOC10D) in high-impedance state.

[Clearing conditions]

- Reset
- By writing 0 to CH910HIZ after reading CH910HIZ = 1

[Setting condition]

- By writing 1 to CH910HIZ

CH6HIZ Bit (MTU6 Output High-Impedance Enable)

This bit specifies whether to place the MTU6 pin in high-impedance state.

[Clearing conditions]

- Reset
- By writing 0 to CH6HIZ after reading CH6HIZ = 1

[Setting condition]

- By writing 1 to CH6HIZ

19.2.8 Port Output Enable Control Register 1 (POECR1)

Address: 0008 890Bh

b7	b6	b5	b4	b3	b2	b1	b0
PE7ZE	PE6ZE	PE5ZE	PE4ZE	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 state. 1: Places the pin in high-impedance state.	R/W*
b1	PE1ZE	MTIOC0B High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*
b2	PE2ZE	MTIOC0C High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*
b3	PE3ZE	MTIOC0D High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*
b4	PE4ZE	MTIOC6A High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*
b5	PE5ZE	MTIOC6B High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*
b6	PE6ZE	MTIOC6C High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*
b7	PE7ZE	MTIOC6D High-Impedance Enable	0: Does not place the pin in high-impedance state. 1: Places the pin in high-impedance state.	R/W*

Note : Can be modified only once after a reset.

POECR1 controls high-impedance state of the pins.

PE0ZE Bit (MTIOC0A High-Impedance Enable)

This bit specifies whether to place the PE34/MTIOC0A pin for the MTU0 in high-impedance state when any of the POE8F flag in ICSR3, CH0HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

PE1ZE Bit (MTIOC0B High-Impedance Enable)

This bit specifies whether to place the P15/MTIOC0B pin for the MTU0 in high-impedance state when any of the POE8F flag in ICSR3, CH0HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

PE2ZE Bit (MTIOC0C High-Impedance Enable)

This bit specifies whether to place the P32/MTIOC0C pin for the MTU0 in high-impedance state when any of the POE8F flag in ICSR3, CH0HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

PE3ZE Bit (MTIOC0D High-Impedance Enable)

This bit specifies whether to place the P33/MTIOC0D pin for the MTU0 in high-impedance state when any of the POE8F flag in ICSR3, CH0HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

PE4ZE Bit (MTIOC6A High-Impedance Enable)

This bit specifies whether to place the PA0/MTIOC6A pin for the MTU6 in high-impedance state when any of the POE9F flag in ICSR3, CH6HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

PE5ZE Bit (MTIOC6B High-Impedance Enable)

This bit specifies whether to place the PA1/MTIOC6B pin for the MTU6 in high-impedance state when any of the POE9F flag in ICSR3, CH6HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

PE6ZE Bit (MTIOC6C High-Impedance Enable)

This bit specifies whether to place the PA2/MTIOC6C pin for the MTU6 in high-impedance state when any of the POE9F flag in ICSR3, CH6HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

PE7ZE Bit (MTIOC6D High-Impedance Enable)

This bit specifies whether to place the PA3/MTIOC6D pin for the MTU6 in high-impedance state when any of the POE9F flag in ICSR3, CH6HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

19.2.9 Port Output Enable Control Register 2 (POECR2)

Address: 0008 890Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	P1CZEA	P2CZEA	P3CZEA	—	P1CZEB	P2CZEB	P3CZEB	—	P4CZE	P5CZE	P6CZE	—	—	—	—
Value after reset:	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0 to b3	—	Reserved	These bits are always read as 0. The write value should be 0.	RW
b4	P6CZE	MTU Port 6 High-Impedance Enable	0: Does not compare the output level or place the pins in high-impedance state. 1: Compare the output level and places the pins in high-impedance state.	R/W*
b5	P5CZE	MTU Port 5 High-Impedance Enable	0: Does not compare the output level or place the pins in high-impedance state. 1: Compare the output level and places the pins in high-impedance state.	R/W*
b6	P4CZE	MTU Port 4 High-Impedance Enable	0: Does not compare the output level or place the pins in high-impedance state. 1: Compare the output level and places the pins in high-impedance state.	R/W*
b7	—	Reserved	This bit is always read as 0. The write value should be 0.	RW
b8	P3CZEB	MTU Port 3 High-Impedance Enable B	0: Does not compare the output level or place the pins in high-impedance state. 1: Compare the output level and places the pins in high-impedance state.	R/W*
b9	P2CZEB	MTU Port 2 High-Impedance Enable B	0: Does not compare the output level or place the pins in high-impedance state. 1: Compare the output level and places the pins in high-impedance state.	R/W*
b10	P1CZEB	MTU Port 1 High-Impedance Enable B	0: Does not compare the output level or place the pins in high-impedance state. 1: Compare the output level and places the pins in high-impedance state.	R/W*
b11	—	Reserved	This bit is always read as 0. The write value should be 0.	RW
b12	P3CZEA	MTU Port 3 High-Impedance Enable A	0: Does not compare the output level or place the pins in high-impedance state. 1: Compare the output level and places the pins in high-impedance state.	R/W*
b13	P2CZEA	MTU Port 2 High-Impedance Enable A	0: Does not compare the output level or place the pins in high-impedance state. 1: Compare the output level and places the pins in high-impedance state.	R/W*
b14	P1CZEA	MTU Port 1 High-Impedance Enable A	0: Does not compare the output level or place the pins in high-impedance state. 1: Compare the output level and places the pins in high-impedance state.	R/W*
b15	—	Reserved	This bit is always read as 0. The write value should be 0.	RW

Note : Can be modified only once after a reset.

POECR2 controls high-impedance state of the pins.

P6CZE Bit (MTU Port 6 High-Impedance Enable)

This bit specifies whether to place the PB6/MTIOC10B and PB7/MTIOC10D pins for the MTU complementary PWM output in high-impedance state when any one of the OSF2 flag in OCSR2, POE4F, POE5F, POE6F, and POE7F flags in ICSR2, and CH910HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

P5CZE Bit (MTU Port 5 High-Impedance Enable)

This bit specifies whether to place the PB4/MTIOC10A and PB5/MTIOC10C pins for the MTU complementary PWM output in high-impedance state when any one of the OSF2 flag in OCSR2, POE4F, POE5F, POE6F, and POE7F flags in ICSR2, and CH910HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

P4CZE Bit (MTU Port 4 High-Impedance Enable)

This bit specifies whether to place the PB2/MTIOC9B and PB3/MTIOC9D pins for the MTU complementary PWM output in high-impedance state when any one of the OSF2 flag in OCSR2, POE4F, POE5F, POE6F, and POE7F flags in ICSR2, and CH910HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

P3CZEB Bit (MTU Port 3 High-Impedance Enable B)

This bit specifies whether to place the P54/MTIOC4B-B and P55/MTIOC4D-B pins for the MTU complementary PWM output in high-impedance state when any one of the OSF1 flag in OCSR1, POE0F, POE1F, POE2F, and POE3F flags in ICSR1, and CH34HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

P2CZEB Bit (MTU Port 2 High-Impedance Enable B)

This bit specifies whether to place the P82/MTIOC4A-B and P83/MTIOC4C-B pins for the MTU complementary PWM output in high-impedance state when any one of the OSF1 flag in OCSR1, POE0F, POE1F, POE2F, and POE3F flags in ICSR1, and CH34HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

P1CZEB Bit (MTU Port 1 High-Impedance Enable B)

This bit specifies whether to place the P80/MTIOC3B-B and P81/MTIOC3D-B pins for the MTU complementary PWM output in high-impedance state when any one of the OSF1 flag in OCSR1, POE0F, POE1F, POE2F, and POE3F flags in ICSR1, and CH34HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

P3CZEA Bit (MTU Port 3 High-Impedance Enable A)

This bit specifies whether to place the P30/MTIOC4B-A and P31/MTIOC4D-A pins for the MTU complementary PWM output in high-impedance state when any one of the OSF1 flag in OCSR1, POE0F, POE1F, POE2F, and POE3F flags in ICSR1, and CH34HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

P2CZEA Bit (MTU Port 2 High-Impedance Enable A)

This bit specifies whether to place the P24/MTIOC4A-A and P25/MTIOC4C-A pins for the MTU complementary PWM output in high-impedance state when any one of the OSF1 flag in OCSR1, POE0F, POE1F, POE2F, and POE3F flags in ICSR1, and CH34HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

P1CZEA Bit (MTU Port 1 High-Impedance Enable A)

This bit specifies whether to place the P22/MTIOC3B-A and P23/MTIOC3D-A pins for the MTU complementary PWM output in high-impedance state when any one of the OSF1 flag in OCSR1, POE0F, POE1F, POE2F, and POE3F flags in ICSR1, and CH34HIZ bit in SPOER, and OSTST bit in NMISR of the ICU is set to 1.

19.3 Operation

Table 19.5 shows the target pins for high-impedance control and conditions to place the pins in high-impedance state.

Table 19.5 Target Pins and Conditions for High-Impedance Control

Pins	Conditions	Detailed Conditions
MTU3 pins (P22/ MTIOC3B-A and P23/ MTIOC3D-A)	POE0# to POE3# input level detection, P22/MTIOC3B and P23/MTIOC3D output level comparison, SPOER setting, or detection of stopped oscillation	P1CZEA • ((POE3F+POE2F+POE1F+POE0F) + (OSF1 • OCE1) + (CH34HIZA) + (OSTST))
MTU4 pins (P24/ MTIOC4A-A and P25/ MTIOC4C-A)	POE0# to POE3# input level detection, P24/MTIOC4A and P25/MTIOC4C output level comparison, SPOER setting, or detection of stopped oscillation	P2CZEA • ((POE3F+POE2F+POE1F+POE0F) + (OSF1 • OCE1) + (CH34HIZ) + (OSTST))
MTU4 pins (P30/ MTIOC4B-A and P31/ MTIOC4D-A)	POE0# to POE3# input level detection, P30/MTIOC4B and P31/MTIOC4D output level comparison, SPOER setting, or detection of stopped oscillation	P3CZEA • ((POE3F+POE2F+POE1F+POE0F) + (OSF1 • OCE1) + (CH34HIZ) + (OSTST))
MTU3 pins (P80/ MTIOC3B-B and P81/ MTIOC3D-B)	POE0# to POE3# input level detection, P80/MTIOC3B and P81/MTIOC3D output level comparison, SPOER setting, or detection of stopped oscillation	P1CZEB • ((POE3F+POE2F+POE1F+POE0F) + (OSF1 • OCE1) + (CH34HIZ) + (OSTST))
MTU4 pins (P82/ MTIOC4A-B and P83/ MTIOC4C-B)	POE0# to POE3# input level detection, P82/MTIOC4A and P83/MTIOC4C output level comparison, SPOER setting, or detection of stopped oscillation	P2CZEB • ((POE3F+POE2F+POE1F+POE0F) + (OSF1 • OCE1) + (CH34HIZ) + (OSTST))
MTU4 pins (P54/ MTIOC4B-B and P55/ MTIOC4D-B)	POE0# to POE3# input level detection, P54/MTIOC4B and P55/MTIOC4D output level comparison, SPOER setting, or detection of stopped oscillation	P3CZEB • ((POE3F+POE2F+POE1F+POE0F) + (OSF1 • OCE1) + (CH34HIZ) + (OSTST))
MTU9 pins (PB2/ MTIOC9B and PB3/ MTIOC9D)	POE4# to POE7# input level detection, PB2/MTIOC9B and PB3/MTIOC9D output level comparison, SPOER setting, or detection of stopped oscillation	P4CZE • ((POE4F+POE5F+POE6F+POE7F) + (OSF2 • OCE2) + (CH910HIZ) + (OSTST))
MTU10 pins (PB4/ MTIOC10A and PB5/ MTIOC10C)	POE4# to POE7# input level detection, PB4/MTIOC10A and PB5/MTIOC10C output level comparison, SPOER setting, or detection of stopped oscillation	P5CZE • ((POE4F+POE5F+POE6F+POE7F) + (OSF2 • OCE2) + (CH910HIZ) + (OSTST))
MTU10 pins (PB6/ MTIOC10B and PB7/ MTIOC10D)	POE4# to POE7# input level detection, PB6/MTIOC10B and PB7/MTIOC10D output level comparison, SPOER setting, or detection of stopped oscillation	P6CZE • ((POE4F+POE5F+POE6F+POE7F) + (OSF2 • OCE2) + (CH910HIZ) + (OSTST))
MTU0 pin (P34/ MTIOC0A)	POE8# input level detection, SPOER setting, or detection of stopped oscillation	PE0ZE • ((POE8F • POE8E) + (CH0HIZ) + (OSTST))
MTU0 pin (P15/ MTIOC0B)	POE8# input level detection, SPOER setting, or detection of stopped oscillation	PE1ZE • ((POE8F • POE8E) + (CH0HIZ) + (OSTST))
MTU0 pin (P32/ MTIOC0C)	POE8# input level detection, SPOER setting, or detection of stopped oscillation	PE2ZE • ((POE8F • POE8E) + (CH0HIZ) + (OSTST))
MTU0 pin (P33/ MTIOC0D)	POE8# input level detection, SPOER setting, or detection of stopped oscillation	PE3ZE • ((POE8F • POE8E) + (CH0HIZ) + (OSTST))
MTU6 pin (PA0/ MTIOC6A)	POE9# input level detection, SPOER setting, or detection of stopped oscillation	PE4ZE • ((POE9F • POE9E) + (CH6HIZ) + (OSTST))
MTU6 pin (PA1/ MTIOC6B)	POE9# input level detection, SPOER setting, or detection of stopped oscillation	PE5ZE • ((POE9F • POE9E) + (CH6HIZ) + (OSTST))
MTU6 pin (PA2/ MTIOC6C)	POE9# input level detection, SPOER setting, or detection of stopped oscillation	PE6ZE • ((POE9F • POE9E) + (CH6HIZ) + (OSTST))
MTU6 pin (PA3/ MTIOC6D)	POE9# input level detection, SPOER setting, or detection of stopped oscillation	PE7ZE • ((POE9F • POE9E) + (CH6HIZ) + (OSTST))

19.3.1 Input Level Detection Operation

If the input conditions set by ICSR1 to ICSR4 occur on the POE0# to POE9# pins, the pins for the MTU complementary PWM output, and MTU0 and MTU6 are placed in high-impedance state. Note however, that these MTU pins also enter high-impedance state even when general output function, MTU unit 0 function, or MTU unit 1 function is not selected for these pins.

(1) Falling Edge Detection

When a change from high to low is input to the POE0# to POE9# pins, the pins for the MTU complementary PWM output, and MTU0 and MTU6 are placed in high-impedance state. Figure 19.2 shows a sample timing after the level changes in input to the POE0# to POE9# pins until the respective pins enter high-impedance state.

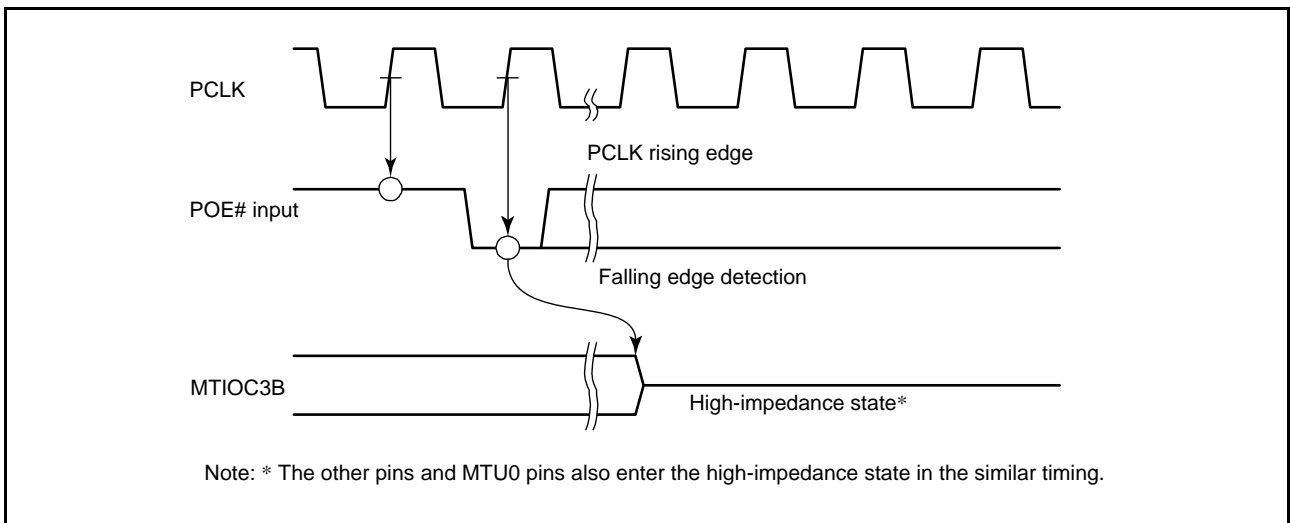


Figure 19.2 Falling Edge Detection

(2) Low-Level Detection

Figure 19.3 shows the low-level detection operation. Sixteen continuous low levels are sampled with the sampling clock selected by ICSR1 to ICSR4. If even one high level is detected during this interval, the low level is not accepted.

The timing when pins for the MTU complementary PWM output, and MTU0 and MTU6 enter the high-impedance state after the sampling clock is input is the same in both falling-edge detection and in low-level detection.

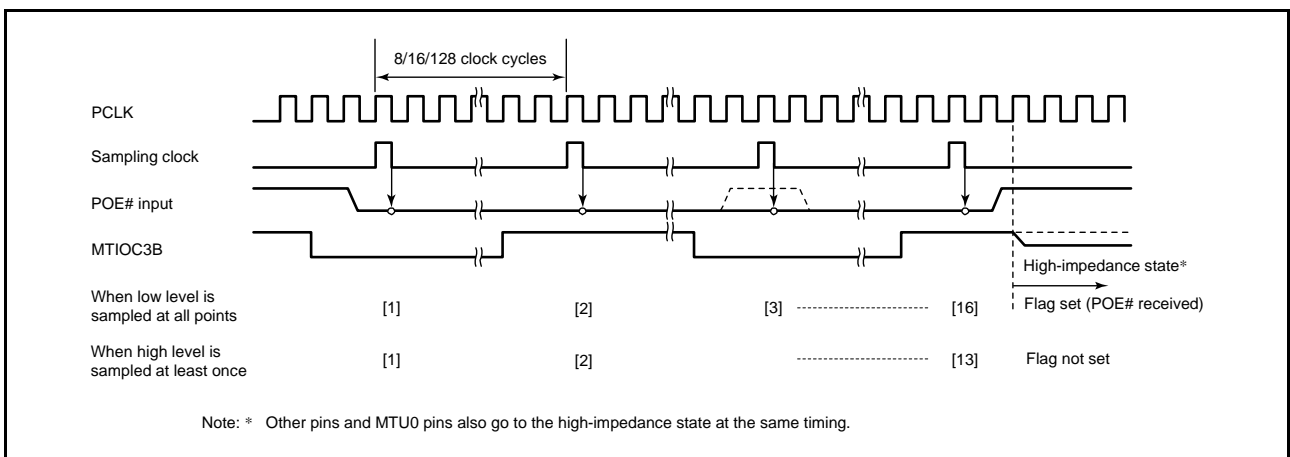


Figure 19.3 Low-Level Detection Operation

19.3.2 Output-Level Compare Operation

Figure 19.4 shows an example of the output-level compare operation for the combination of MTIOC3B and MTIOC3D. The operation is the same for the other pin combinations.

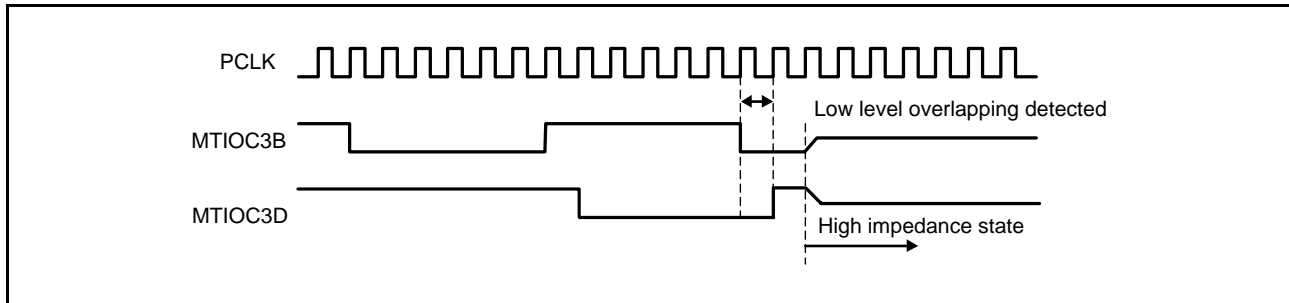


Figure 19.4 Output-Level Compare Operation

19.3.3 High-Impedance Control Using Registers

The high-impedance state of the MTU pins (MTU0, MTU6, MTU3, MTU4, MTU9, and MTU10) can be directly controlled using the software port output enable register (SPOER).

Setting the CH34HIZ bit in SPOER to 1 places the MTU3 and MTU4 pins specified by port output enable control register 2 (POE2CR) in the high-impedance state.

The high-impedance state of other pins can also be controlled by setting the appropriate bits in SPOER.

19.3.4 High-Impedance Control through Detection of Stopped Oscillation

When the oscillation-stop detection circuit in the clock pulse generator detects stopped oscillation, the MTU complementary PWM output pins specified by port output enable control register 2 (POE2CR) and MTU0 and MTU6 pins specified by port output enable control register 1 (POE1CR) can be placed in the high-impedance state.

19.3.5 Release from High-Impedance State

MTU pins which have entered high-impedance state due to input-level detection can be released from the state either by returning them to their initial state with a reset, or by clearing all of the POE3F to POE9F flags in ICSR1 to ICSR4.

However, note that when low-level sampling is selected with the POE0M[1:0] to POE9M[1:0] bits in ICSR1 to ICSR4, just writing 0 to a flag is ignored (the flag is not cleared); flags can be cleared by writing 0 to it only after a high level is input to the POE0# to POE9# pins and is sampled.

MTU pins which have entered high-impedance state due to output-level detection can be released from the state either by returning them to their initial state with a reset, or by clearing the OCF1 and OCF2 flags in OCSR1 and OCSR2.

However, note that just writing 0 to a flag is ignored (the flag is not cleared); flags can be cleared only after an inactive level is output from the MTU complementary PWM pins. Inactive-level outputs can be achieved by setting the MTU unit 0 and unit 1 internal registers.

19.4 Interrupts

The POE issues a request to generate an interrupt when the specified condition is satisfied during input level detection or output level comparison. Table 19.6 shows the interrupt sources and their conditions.

Table 19.6 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OE11	Output enable interrupt 1	POE3F, POE2F, POE1F, POE0F, OSF1	PIE1 • (POE0F+POE1F+POE2F+POE3F) + OIE1 • OSF1
OE12	Output enable interrupt 2	POE8F	PIE3 • POE8F
OE13	Output enable interrupt 3	POE4F, POE5F, POE6F, POE7F, OSF2	PIE2 • (POE4F+POE5F+POE6F+POE7F) + OIE2 • OSF2
OE14	Output enable interrupt 4	POE9F	PIE4 • POE9F

19.5 Usage Notes

When the POE is used, do not make a transition to software standby mode or deep software standby mode. In these modes, the POE stops and thus the high-impedance state of pins cannot be controlled.

When the POE is not to be used, write 00h to the port output enable control registers 1 and 2 (POECCR1 and POECCR2), respectively.

20. Programmable Pulse Generator (PPG)

The programmable pulse generator (PPG) generates pulse outputs by using the multi-function timer pulse unit (MTU) as a time base.

The RX62N/RX621 Group has two PPG units, each of which controls up to 16 pulse output pins. The pulse outputs from the PPGs are divided into 4-bit groups that can operate all simultaneously and independently.

20.1 Overview

Table 20.1 lists the specifications of the PPG and Table 20.2 lists PPG functions.

Figure 20.1 and Figure 20.2 show block diagrams of the PPGs.

Table 20.1 Specifications of PPG

Item	Specifications
Number of output bits	Up to 32 bits
Pulse output	<ul style="list-style-type: none"> • Two units, each capable of output through four pin groups • Output trigger signals are selectable. • Non-overlapping operation is possible. • Inverted output is selectable.
Output data transfer	Can operate together with the DTC and DMACA (When MTU interrupt is in use)
Power consumption reducing function	Module stop state can be set for each unit.

Table 20.2 List of PPG Functions

Item		PPG0	PPG1
PPG output trigger	MTU (unit 0) channels 0 to 3 (MTU0 to MTU3)	Compare match	√
		Input capture	√
	MTU (unit 1) channels 6 to 9 (MTU6 to MTU9)	Compare match	—
		Input capture	—
Non-overlapping operation		√	√
Output data transfer	DTC	√	√
	DMACA	√	√
Selecting inverted output		√	√
Setting the module stop state*		The MSTPA11 bit in MSTPCRA	The MSTPA10 bit in MSTPCRA

[Legend] √: Possible
 —: Not possible

Note : * For details, see section 9, Low Power Consumption.

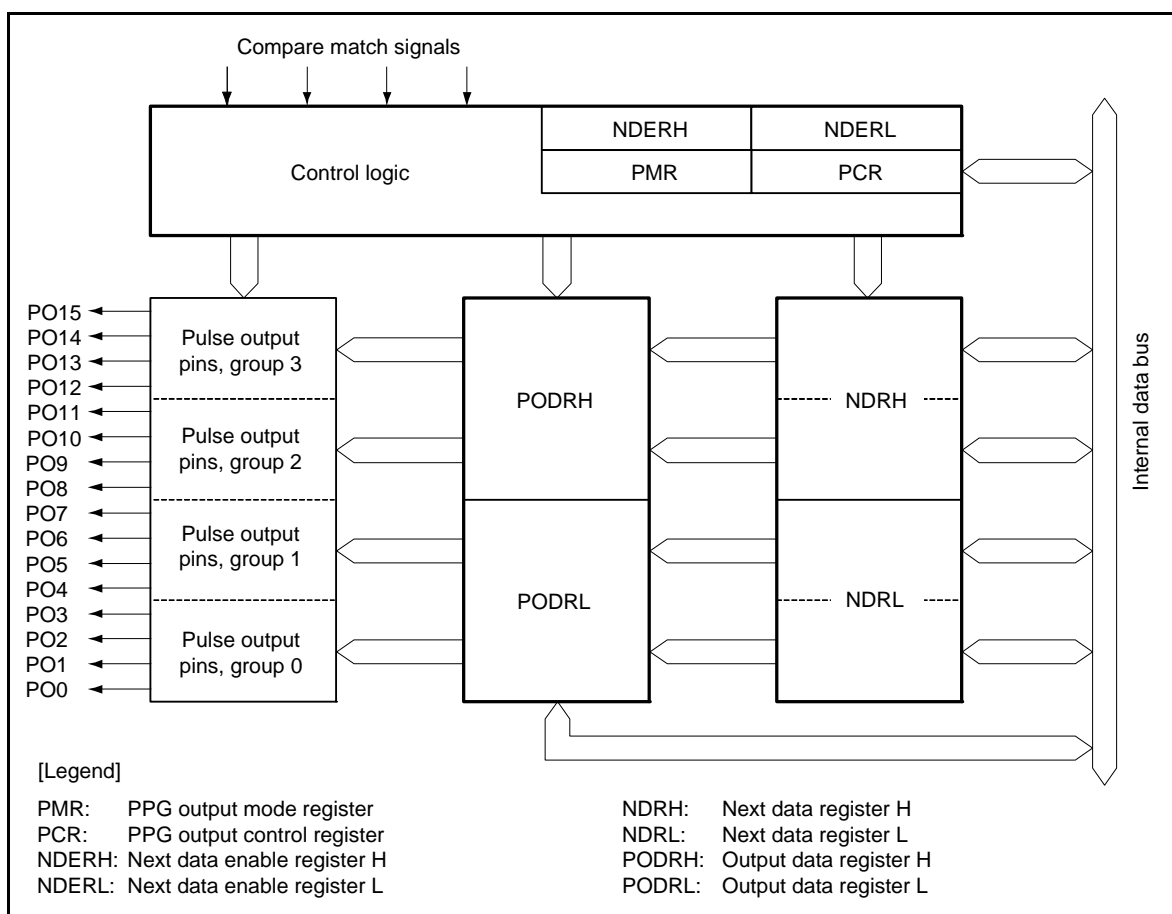


Figure 20.1 Block Diagram of PPG (Unit 0)

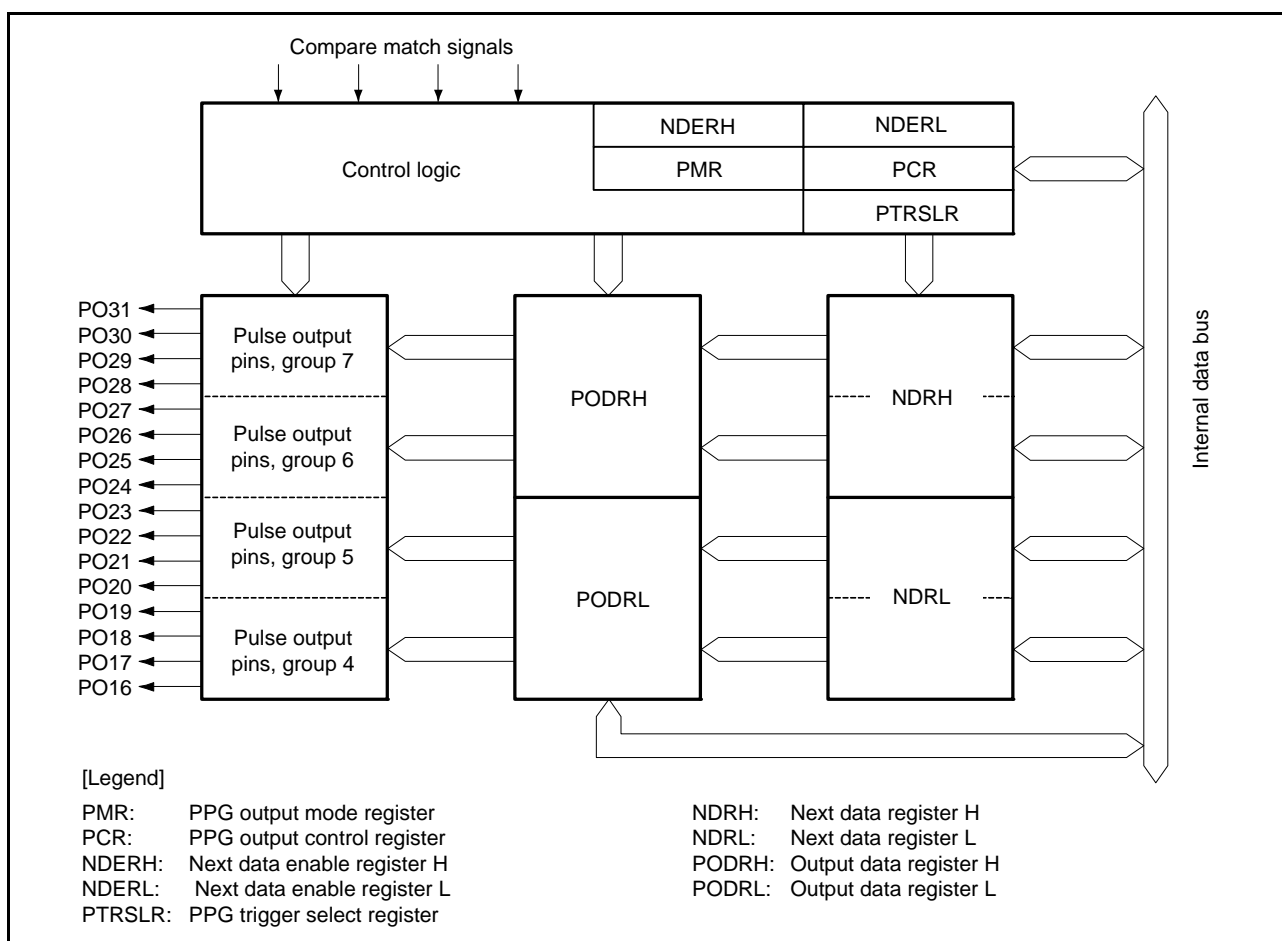


Figure 20.2 Block Diagram of PPG (Unit 1)

Table 20.3 lists the pin configuration of the PPG.

Table 20.3 Pin Configuration of PPG

Unit	Pin Name	I/O	Function	
PPG0	PO0	Output	Group 0 pulse output	
	PO1	Output		
	PO2	Output		
	PO3	Output		
	PO4	Output	Group 1 pulse output	
	PO5	Output		
	PO6	Output		
	PO7	Output		
	PO8	Output	Group 2 pulse output	
	PO9	Output		
	PO10	Output		
	PO11	Output		
	PPG1	PO12	Output	Group 3 pulse output
		PO13	Output	
		PO14	Output	
PO15		Output		
PO16		Output	Group 4 pulse output	
PO17		Output		
PO18		Output		
PO19		Output		
PO20		Output	Group 5 pulse output	
PO21		Output		
PO22	Output			
PO23	Output			
PO24	Output	Group 6 pulse output		
PO25	Output			
PO26	Output			
PO27	Output			
PO28	Output	Group 7 pulse output		
PO29	Output			
PO30	Output			
PO31	Output			

20.2 Register Descriptions

Table 20.4 lists the registers of the PPG.

Table 20.4 Registers of PPG

Unit	Register Name	Symbol	Value after Reset	Address	Access Size
PPG0	PPG output control register	PCR	FFh	0008 81E6h	8
	PPG output mode register	PMR	F0h	0008 81E7h	8
	Next data transfer enable register H	NDERH	00h	0008 81E8h	8
	Next data transfer enable register L	NDERL	00h	0008 81E9h	8
	Output data register H	PODRH	00h	0008 81EAh	8
	Output data register L	PODRL	00h	0008 81EBh	8
	Next data register H	NDRH	00h	0008 81ECh*1	8
	Next data register L	NDRL	00h	0008 81EDh*2	8
	Next data register H2	NDRH2	00h	0008 81EEh*1	8
	Next data register L2	NDRL2	00h	0008 81EFh*2	8
PPG1	PPG trigger select register	PTRSLR	01h	0008 81F0h	8
	PPG output control register	PCR	FFh	0008 81F6h	8
	PPG output mode register	PMR	F0h	0008 81F7h	8
	Next data transfer enable register H	NDERH	00h	0008 81F8h	8
	Next data transfer enable register L	NDERL	00h	0008 81F9h	8
	Output data register H	PODRH	00h	0008 81FAh	8
	Output data register L	PODRL	00h	0008 81FBh	8
	Next data register H	NDRH	00h	0008 81FCh*3	8
	Next data register L	NDRL	00h	0008 81FDh*4	8
	Next data register H2	NDRH2	00h	0008 81FEh*3	8
Next data register L2	NDRL2	00h	0008 81FFh*4	8	

Note 1. When pulse output groups 2 and 3 have the same output trigger by PPG0.PCR settings, the PPG0.NDRH address is 0008 81ECh. When they have different output triggers, the PPG0.NDRH2 addresses corresponding to groups 2 and 3 are 0008 81EEh and 0008 81ECh, respectively.

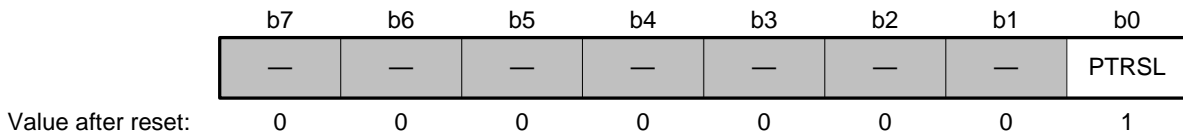
Note 2. When pulse output groups 0 and 1 have the same output trigger by PPG0.PCR settings, the PPG0.NDRL address is 0008 81EDh. When they have different output triggers, the PPG0.NDRL2 addresses corresponding to groups 0 and 1 are 0008 81EFh and 0008 81EDh, respectively.

Note 3. When pulse output groups 6 and 7 have the same output trigger by PPG1.PCR settings, the PPG1.NDRH address is 0008 81FCh. When they have different output triggers, the PPG1.NDRH2 addresses corresponding to groups 6 and 7 are 0008 81FEh and 0008 81FCh, respectively.

Note 4. When pulse output groups 4 and 5 have the same output trigger by PPG1.PCR settings, the PPG1.NDRL address is 0008 81FDh. When they have different output triggers, the PPG1.NDRL2 addresses corresponding to groups 4 and 5 are 0008 81FFh and 0008 81FDh, respectively.

20.2.1 PPG Trigger Select Register (PTRSLR)

Address: 0008 81F0h



- PPG1.PTRSLR

Bit	Symbol	Bit Name	Description	R/W
b0	PTRSL	PPG Trigger Select	0: Selects the set of MTU0 to MTU3 as the trigger channels for PPG1. 1: Selects the set of MTU6 to MTU9 as the trigger channels for PPG1.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

PPG1.PTRSLR selects a set of trigger channels.

PTRSL Bit (PPG Trigger Select)

This bit selects either MTU0 to MTU3 or MTU6 to MTU9 as a set of trigger channels for PPG1.

When this bit is set to 0, MTU0 to MTU3 are selected as a set of trigger channels for PPG1. When it is set to 1, MTU6 to MTU9 are selected as a set of trigger channels for PPG1.

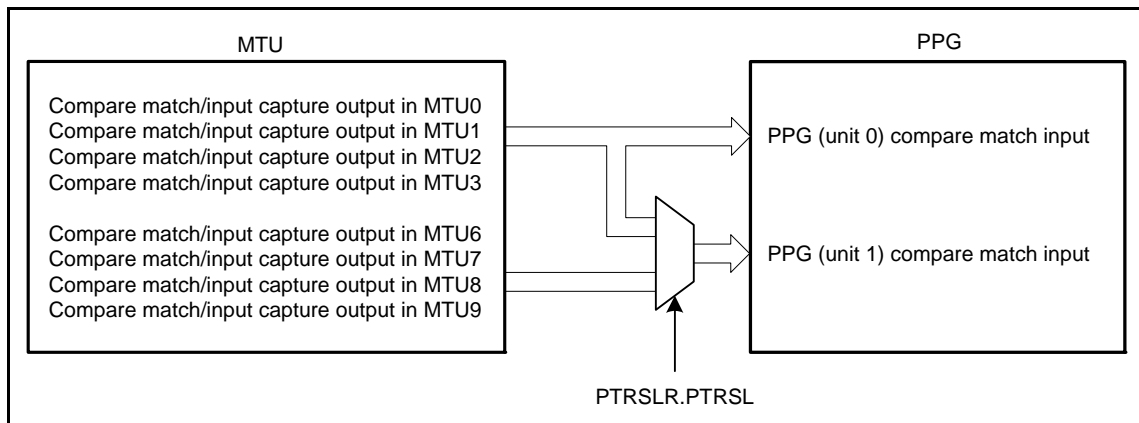


Figure 20.3 Block Diagram of PPG Trigger Selection

20.2.2 Next Data Transfer Enable Registers H and L (NDERH, NDERL)

Address: 0008 81E8h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
Value after reset:	0	0	0	0	0	0	0	0

Address: 0008 81E9h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Value after reset:	0	0	0	0	0	0	0	0

- PPG0.NDERH

Bit	Symbol	Bit Name	Description	R/W
b0	NDER8	Next Data Transfer Enable	0: Data transfer is disabled. 1: Data transfer is enabled.	R/W
b1	NDER9	Next Data Transfer Enable		R/W
b2	NDER10	Next Data Transfer Enable		R/W
b3	NDER11	Next Data Transfer Enable		R/W
b4	NDER12	Next Data Transfer Enable		R/W
b5	NDER13	Next Data Transfer Enable		R/W
b6	NDER14	Next Data Transfer Enable		R/W
b7	NDER15	Next Data Transfer Enable		R/W

PPG0.NDERH selects the pins (PO8 to PO15) for outputs of pulse from the PPG on a bit-by-bit basis.

NDER_i Bits (Next Data Transfer Enable) (i = 15 to 8)

When these bits are set to 1, the PTRSLR specified trigger transfers data from the corresponding bit in PPG0.NDRH to the bit in PPG0.PODRH. When these bits are set to 0, the data are not transferred from PPG0.NDRH to PPG0.PODRH.

- PPG0.NDERL

Bit	Symbol	Bit Name	Description	R/W
b0	NDER0	Next Data Transfer Enable	0: Data transfer is disabled. 1: Data transfer is enabled.	R/W
b1	NDER1	Next Data Transfer Enable		R/W
b2	NDER2	Next Data Transfer Enable		R/W
b3	NDER3	Next Data Transfer Enable		R/W
b4	NDER4	Next Data Transfer Enable		R/W
b5	NDER5	Next Data Transfer Enable		R/W
b6	NDER6	Next Data Transfer Enable		R/W
b7	NDER7	Next Data Transfer Enable		R/W

PPG0.NDERL selects the pins (PO0 to PO7) for outputs of pulse from the PPG on a bit-by-bit basis.

NDERi Bits (Next Data Transfer Enable) (i = 7 to 0)

When these bits are set to 1, the PTRSLR specified trigger transfers data from the corresponding bit in PPG0.NDRH to the bit in PPG0.PODRL. When these bits are set to 0, the data are not transferred from PPG0.NDRH to PPG0.PODRL.

Address: 0008 81F8h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDERH	NDER31	NDER30	NDER29	NDER28	NDER27	NDER26	NDER25	NDER24
Value after reset:	0	0	0	0	0	0	0	0

Address: 0008 81F9h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDERL	NDER23	NDER22	NDER21	NDER20	NDER19	NDER18	NDER17	NDER16
Value after reset:	0	0	0	0	0	0	0	0

- PPG1.NDERH

Bit	Symbol	Bit Name	Description	R/W
b0	NDER24	Next Data Transfer Enable	0: Data transfer is disabled.	R/W
b1	NDER25	Next Data Transfer Enable	1: Data transfer is enabled.	R/W
b2	NDER26	Next Data Transfer Enable		R/W
b3	NDER27	Next Data Transfer Enable		R/W
b4	NDER28	Next Data Transfer Enable		R/W
b5	NDER29	Next Data Transfer Enable		R/W
b6	NDER30	Next Data Transfer Enable		R/W
b7	NDER31	Next Data Transfer Enable		R/W

PPG1.NDERH selects the pins (PO24 to PO31) for outputs of pulse from the PPG on a bit-by-bit basis.

NDERi Bits (Next Data Transfer Enable) (i = 31 to 24)

When these bits are set to 1, the PTRSLR specified trigger transfers data from the corresponding bit in PPG1.NDRH to the bit in PPG1.PODRH. When these bits are set to 0, the data are not transferred from PPG1.NDRH to PPG1.PODRH.

- PPG1.NDERL

Bit	Symbol	Bit Name	Description	R/W
b0	NDER16	Next Data Transfer Enable	0: Data transfer is disabled.	R/W
b1	NDER17	Next Data Transfer Enable	1: Data transfer is enabled.	R/W
b2	NDER18	Next Data Transfer Enable		R/W
b3	NDER19	Next Data Transfer Enable		R/W
b4	NDER20	Next Data Transfer Enable		R/W
b5	NDER21	Next Data Transfer Enable		R/W
b6	NDER22	Next Data Transfer Enable		R/W
b7	NDER23	Next Data Transfer Enable		R/W

PPG1.NDERL selects the pins (PO16 to PO23) for outputs of pulse from the PPG on a bit-by-bit basis.

NDER_i Bits (Next Data Transfer Enable) (i = 23 to 16)

When these bits are set to 1, the PTRSLR specified trigger transfers data from the corresponding bit in PPG1.NDRL to the bit in PPG1.PODRL. When these bits are set to 0, the data are not transferred from PPG1.NDRL to PPG1.PODRL.

20.2.3 Output Data Registers H and L (PODRH, PODRL)

Address: 0008 81EAh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
Value after reset:	0	0	0	0	0	0	0	0

Address: 0008 81EBh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
Value after reset:	0	0	0	0	0	0	0	0

- PPG0.PODRH

Bit	Symbol	Bit Name	Description	R/W
b0	POD8	Output Data Register	For bits corresponding to pins that have been set for pulse output by PPG0.NDERH, the output trigger transfers the values in PPG0.NDRH to this register during PPG operation. If any of the NDERi (i = 8 to 15) bits in PPG0.NDERH is set to 1, the CPU cannot write to this register. Initial output values for the pins can be set while PPG0.NDERH is clear (00h).	R/W
b1	POD9	Output Data Register		R/W
b2	POD10	Output Data Register		R/W
b3	POD11	Output Data Register		R/W
b4	POD12	Output Data Register		R/W
b5	POD13	Output Data Register		R/W
b6	POD14	Output Data Register		R/W
b7	POD15	Output Data Register		R/W

PPG0.PODRH stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG0.NDERH, the output trigger transfers the values in PPG0.NDRH to this register.

- PPG0.PODRL

Bit	Symbol	Bit Name	Description	R/W
b0	POD0	Output Data Register	For bits corresponding to pins that have been set for pulse output by PPG0.NDERL, the output trigger transfers the values in PPG0.NDRL to this register during PPG operation. If any of the NDERi (i = 0 to 7) bits in PPG0.NDERL is set to 1, the CPU cannot write to this register. Initial output values for the pins can be set while PPG0.NDERL is clear (00h).	R/W
b1	POD1	Output Data Register		R/W
b2	POD2	Output Data Register		R/W
b3	POD3	Output Data Register		R/W
b4	POD4	Output Data Register		R/W
b5	POD5	Output Data Register		R/W
b6	POD6	Output Data Register		R/W
b7	POD7	Output Data Register		R/W

PPG0.PODRL stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG0.NDERL, the output trigger transfers the values in PPG0.NDRL to this register.

Address: 0008 81FAh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.PODRH	POD31	POD30	POD29	POD28	POD27	POD26	POD25	POD24
Value after reset:	0	0	0	0	0	0	0	0

Address: 0008 81FBh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.PODRL	POD23	POD22	POD21	POD20	POD19	POD18	POD17	POD16
Value after reset:	0	0	0	0	0	0	0	0

- PPG1.PODRH

Bit	Symbol	Bit Name	Description	R/W
b0	POD24	Output Data Register	For bits corresponding to pins that have been set for pulse output by PPG1.NDERH, the output trigger transfers the values in PPG1.NDRH to this register during PPG operation. If any of the NDERi (i = 24 to 31) bits in PPG1.NDERH is set to 1, the CPU cannot write to this register. Initial output values for the pins can be set while PPG1.NDERH is clear (00h).	R/W
b1	POD25	Output Data Register		R/W
b2	POD26	Output Data Register		R/W
b3	POD27	Output Data Register		R/W
b4	POD28	Output Data Register		R/W
b5	POD29	Output Data Register		R/W
b6	POD30	Output Data Register		R/W
b7	POD31	Output Data Register		R/W

PPG1.PODRH stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG1.NDERH, the output trigger transfers the values in PPG1.NDRH to this register.

- PPG1.PODRL

Bit	Symbol	Bit Name	Description	R/W
b0	POD16	Output Data Register	For bits corresponding to pins that have been set for pulse output by PPG1.NDERL, the output trigger transfers the values in PPG1.NDRL to this register during PPG operation. If any of the NDERi (i = 16 to 23) bits in PPG1.NDERL is set to 1, the CPU cannot write to this register. Initial output values for the pins can be set while PPG1.NDERL is clear (00h).	R/W
b1	POD17	Output Data Register		R/W
b2	POD18	Output Data Register		R/W
b3	POD19	Output Data Register		R/W
b4	POD20	Output Data Register		R/W
b5	POD21	Output Data Register		R/W
b6	POD22	Output Data Register		R/W
b7	POD23	Output Data Register		R/W

PPG1.PODRL stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG1.NDERL, the output trigger transfers the values in PPG1.NDRL to this register.

20.2.4 Next Data Registers H and L (NDRH, NDRL)

Addresses: NDRH.0008 81ECh, NDRH2.0008 81EEh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDRH	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
Value after reset:	0	0	0	0	0	0	0	0

Addresses: NDRL.0008 81EDh, NDRL2.0008 81EFh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDRL	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Value after reset:	0	0	0	0	0	0	0	0

- PPG0.NDRH

PPG0.NDRH stores the next data for pulse output. The PPG0.NDRH address differs depending on whether pulse output groups have the same output trigger or different output triggers.

(1) When pulse output groups 2 and 3 have the same output trigger

If pulse output groups 2 and 3 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

(0008 81ECh)

Bit	Symbol	Bit Name	Description	R/W
b0	NDR8	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRH.	R/W
b1	NDR9	Next Data Register		R/W
b2	NDR10	Next Data Register		R/W
b3	NDR11	Next Data Register		R/W
b4	NDR12	Next Data Register		R/W
b5	NDR13	Next Data Register		R/W
b6	NDR14	Next Data Register		R/W
b7	NDR15	Next Data Register		R/W

(2) When pulse output groups 2 and 3 have different output triggers

If pulse output groups 2 and 3 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

(Pulse output group 3: 0008 81ECh)

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 1. The write value should be 1.	R/W
b4	NDR12	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRH.	R/W
b5	NDR13	Next Data Register		R/W
b6	NDR14	Next Data Register		R/W
b7	NDR15	Next Data Register		R/W

(Pulse output group 2: 0008 81EEh)

Bit	Symbol	Bit Name	Description	R/W
b0	NDR8	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRH.	R/W
b1	NDR9	Next Data Register		R/W
b2	NDR10	Next Data Register		R/W
b3	NDR11	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are always read as 1. The write value should be 1.	R/W

- PPG0.NDRL

PPG0.NDRL stores the next data for pulse output. The PPG0.NDRL address differs depending on whether pulse output groups have the same output trigger or different output triggers.

(1) When pulse output groups 0 and 1 have the same output trigger

If pulse output groups 0 and 1 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

(0008 81EDh)

Bit	Symbol	Bit Name	Description	R/W
b0	NDR0	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRL.	R/W
b1	NDR1	Next Data Register		R/W
b2	NDR2	Next Data Register		R/W
b3	NDR3	Next Data Register		R/W
b4	NDR4	Next Data Register		R/W
b5	NDR5	Next Data Register		R/W
b6	NDR6	Next Data Register		R/W
b7	NDR7	Next Data Register		R/W

(2) When pulse output groups 0 and 1 have different output triggers

If pulse output groups 0 and 1 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

(Pulse output group 1: 0008 81EDh)

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 1. The write value should be 1.	R/W
b4	NDR4	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRL.	R/W
b5	NDR5	Next Data Register		R/W
b6	NDR6	Next Data Register		R/W
b7	NDR7	Next Data Register		R/W

(Pulse output group 0: 0008 81EFh)

Bit	Symbol	Bit Name	Description	R/W
b0	NDR0	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRL.	R/W
b1	NDR1	Next Data Register		R/W
b2	NDR2	Next Data Register		R/W
b3	NDR3	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are always read as 1. The write value should be 1.	R/W

Addresses: NDRH.0008 81FCh, NDRH2.0008 81FEh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDRH	NDR31	NDR30	NDR29	NDR28	NDR27	NDR26	NDR25	NDR24
Value after reset:	0	0	0	0	0	0	0	0

Addresses: NDRL.0008 81FDh, NDRL2.0008 81FFh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDRL	NDR23	NDR22	NDR21	NDR20	NDR19	NDR18	NDR17	NDR16
Value after reset:	0	0	0	0	0	0	0	0

- PPG1.NDRH

PPG1.NDRH stores the next data for pulse output. The PPG1.NDRH address differs depending on whether pulse output groups have the same output trigger or different output triggers.

(1) When pulse output groups 6 and 7 have the same output trigger

If pulse output groups 6 and 7 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

(0008 81FCh)

Bit	Symbol	Bit Name	Description	R/W
b0	NDR24	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRH.	R/W
b1	NDR25	Next Data Register		R/W
b2	NDR26	Next Data Register		R/W
b3	NDR27	Next Data Register		R/W
b4	NDR28	Next Data Register		R/W
b5	NDR29	Next Data Register		R/W
b6	NDR30	Next Data Register		R/W
b7	NDR31	Next Data Register		R/W

(2) When pulse output groups 6 and 7 have different output triggers

If pulse output groups 6 and 7 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

(Pulse output group 7: 0008 81FCh)

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 1. The write value should be 1.	R/W
b4	NDR28	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRH.	R/W
b5	NDR29	Next Data Register		R/W
b6	NDR30	Next Data Register		R/W
b7	NDR31	Next Data Register		R/W

(Pulse output group 6: 0008 81FEh)

Bit	Symbol	Bit Name	Description	R/W
b0	NDR24	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRH.	R/W
b1	NDR25	Next Data Register		R/W
b2	NDR26	Next Data Register		R/W
b3	NDR27	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are always read as 1. The write value should be 1.	R/W

- PPG1.NDRL

PPG1.NDRL stores the next data for pulse output. The PPG1.NDRL address differs depending on whether pulse output groups have the same output trigger or different output triggers.

(1) When pulse output groups 4 and 5 have the same output trigger

If pulse output groups 4 and 5 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

(0008 81FDh)

Bit	Symbol	Bit Name	Description	R/W
b0	NDR16	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRL.	R/W
b1	NDR17	Next Data Register		R/W
b2	NDR18	Next Data Register		R/W
b3	NDR19	Next Data Register		R/W
b4	NDR20	Next Data Register		R/W
b5	NDR21	Next Data Register		R/W
b6	NDR22	Next Data Register		R/W
b7	NDR23	Next Data Register		R/W

(2) When pulse output groups 4 and 5 have different output triggers

If pulse output groups 4 and 5 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

(Pulse output group 5: 0008 81FDh)

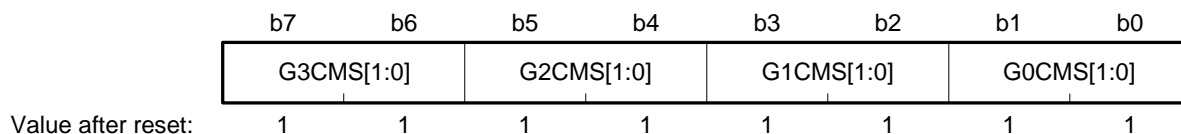
Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 1. The write value should be 1.	R/W
b4	NDR20	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRL.	R/W
b5	NDR21	Next Data Register		R/W
b6	NDR22	Next Data Register		R/W
b7	NDR23	Next Data Register		R/W

(Pulse output group 4: 0008 81FFh)

Bit	Symbol	Bit Name	Description	R/W
b0	NDR16	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRL.	R/W
b1	NDR17	Next Data Register		R/W
b2	NDR18	Next Data Register		R/W
b3	NDR19	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are always read as 1. The write value should be 1.	R/W

20.2.5 PPG Output Control Register (PCR)

Addresses: PPG0.PCR 0008 81E6h, PPG1.PCR 0008 81F6h



- PPG0.PCR

Bit	Symbol	Bit Name	Description	R/W
b1, b0	G0CMS[1:0]	Group 0 Compare Match Select	b1 b0 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W
b3, b2	G1CMS[1:0]	Group 1 Compare Match Select	b3 b2 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W
b5, b4	G2CMS[1:0]	Group 2 Compare Match Select	b5 b4 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W
b7, b6	G3CMS[1:0]	Group 3 Compare Match Select	b7 b6 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W

- PPG1.PCR

Bit	Symbol	Bit Name	Description	R/W
b1, b0	G0CMS[1:0]	Group 4 Compare Match Select	<ul style="list-style-type: none"> When the PTRSL bit in PPG1.PTRSLR is set to 0. <ul style="list-style-type: none"> b1 b0 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3 When the PTRSL bit in PPG1.PTRSLR is set to 1. <ul style="list-style-type: none"> b1 b0 0 0: Compare match in MTU6 0 1: Compare match in MTU7 1 0: Compare match in MTU8 1 1: Compare match in MTU9 	R/W
b3, b2	G1CMS[1:0]	Group 5 Compare Match Select	<ul style="list-style-type: none"> When the PTRSL bit in PPG1.PTRSLR is set to 0. <ul style="list-style-type: none"> b3 b2 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3 When the PTRSL bit in PPG1.PTRSLR is set to 1. <ul style="list-style-type: none"> b3 b2 0 0: Compare match in MTU6 0 1: Compare match in MTU7 1 0: Compare match in MTU8 1 1: Compare match in MTU9 	R/W
b5, b4	G2CMS[1:0]	Group 6 Compare Match Select	<ul style="list-style-type: none"> When the PTRSL bit in PPG1.PTRSLR is set to 0. <ul style="list-style-type: none"> b5 b4 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3 When the PTRSL bit in PPG1.PTRSLR is set to 1. <ul style="list-style-type: none"> b5 b4 0 0: Compare match in MTU6 0 1: Compare match in MTU7 1 0: Compare match in MTU8 1 1: Compare match in MTU9 	R/W
b7, b6	G3CMS[1:0]	Group 7 Compare Match Select	<ul style="list-style-type: none"> When the PTRSL bit in PPG1.PTRSLR is set to 0. <ul style="list-style-type: none"> b7 b6 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3 When the PTRSL bit in PPG1.PTRSLR is set to 1. <ul style="list-style-type: none"> b7 b6 0 0: Compare match in MTU6 0 1: Compare match in MTU7 1 0: Compare match in MTU8 1 1: Compare match in MTU9 	R/W

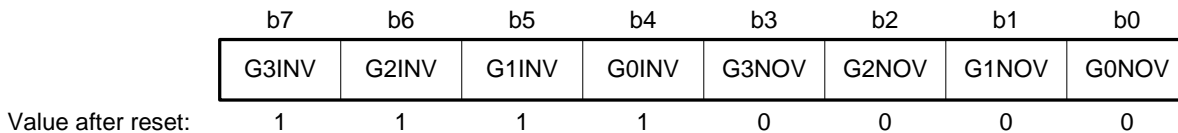
PPGn.PCR (n = 0, 1) selects pulse output trigger signals on a group-by-group basis. For details on output trigger selection, see section 20.2.6, PPG Output Mode Register (PMR).

GiCMS[1:0] Bits (Group j Compare Match Select) (i = 0 to 3)

Each bit selects an output trigger of pulse output group j (j = 0 to 7).

20.2.6 PPG Output Mode Register (PMR)

Addresses: PPG0.PMR 0008 81E7h, PPG1.PMR 0008 81F7h



- PPG0.PMR

Bit	Symbol	Bit Name	Description	R/W
b0	G0NOV	Group 0 Non-Overlap	0:Normal operation (Output values updated on compare match A of TGRA in the selected MTUn) 1:Non-overlapping operation (Output values updated on compare match A or B of TGRA or TGRB in the selected MTUn) (n = 0 to 3)	R/W
b1	G1NOV	Group 1 Non-Overlap	0:Normal operation (Output values updated on compare match A of TGRA in the selected MTUn) 1:Non-overlapping operation (Output values updated on compare match A or B of TGRA or TGRB in the selected MTUn) (n = 0 to 3)	R/W
b2	G2NOV	Group 2 Non-Overlap	0:Normal operation (Output values updated on compare match A of TGRA in the selected MTUn) 1:Non-overlapping operation (Output values updated on compare match A or B of TGRA or TGRB in the selected MTUn) (n = 0 to 3)	R/W
b3	G3NOV	Group 3 Non-Overlap	0:Normal operation (Output values updated on compare match A of TGRA in the selected MTUn) 1:Non-overlapping operation (Output values updated on compare match A or B of TGRA or TGRB in the selected MTUn) (n = 0 to 3)	R/W
b4	G0INV	Group 0 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b5	G1INV	Group 1 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b6	G2INV	Group 2 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b7	G3INV	Group 3 Output Polarity Change	0: Inverted output 1: Direct output	R/W

- PPG1.PMR

Bit	Symbol	Bit Name	Description	R/W
b0	G0NOV	Group 4 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3, 6 to 9)	R/W
b1	G1NOV	Group 5 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3, 6 to 9)	R/W
b2	G2NOV	Group 6 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3, 6 to 9)	R/W
b3	G3NOV	Group 7 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3, 6 to 9)	R/W
b4	G0INV	Group 4 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b5	G1INV	Group 5 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b6	G2INV	Group 6 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b7	G3INV	Group 7 Output Polarity Change	0: Inverted output 1: Direct output	R/W

PPGn.PMR (n = 0, 1) selects the pulse output mode of the PPG on a group-by-group basis.

While inverted output is selected, a low-level pulse is output when the values in PPGn.PODRH and PPGn.PODRL are 1, and a high-level pulse is output when the values in PPGn.PODRH and PPGn.PODRL are 0.

In addition, when non-overlapping operation is selected, the PPG updates its output values on compare match A or B in a MPU channel that functions as an output trigger.

For details, see section 20.3.4, Non-Overlapping Pulse Output.

GiNOV Bits (Group j Non-Overlap) (i = 0 to 3)

Each bit selects normal operation or non-overlapping operation for pulse output group j (j = 0 to 7).

GiINV Bits (Group j Output Polarity Change) (i = 0 to 3)

Each bit selects direct output or inverted output for pulse output group j (j = 0 to 7).

20.3 Operation

Figure 20.4 shows a schematic diagram of the PPG.

PPG pulse output is enabled when the corresponding bits in PPGn.NDERH and PPGn.NDERL (n = 0, 1) are set to 1 (data transfer is enabled).

An initial output value is determined by the initial settings in the corresponding PPGn.PODRH and PPGn.PODRL. When the compare match event selected in PPGn.PCR occurs, the output values are updated by transfer of the values in the corresponding PPGn.NDRH and PPGn.NDRL to PPGn.PODRH and PPGn.PODRL, respectively.

Consecutive output of up to 16 bits of data is possible by writing new output data to PPGn.NDRH and PPGn.NDRL before the next compare match.

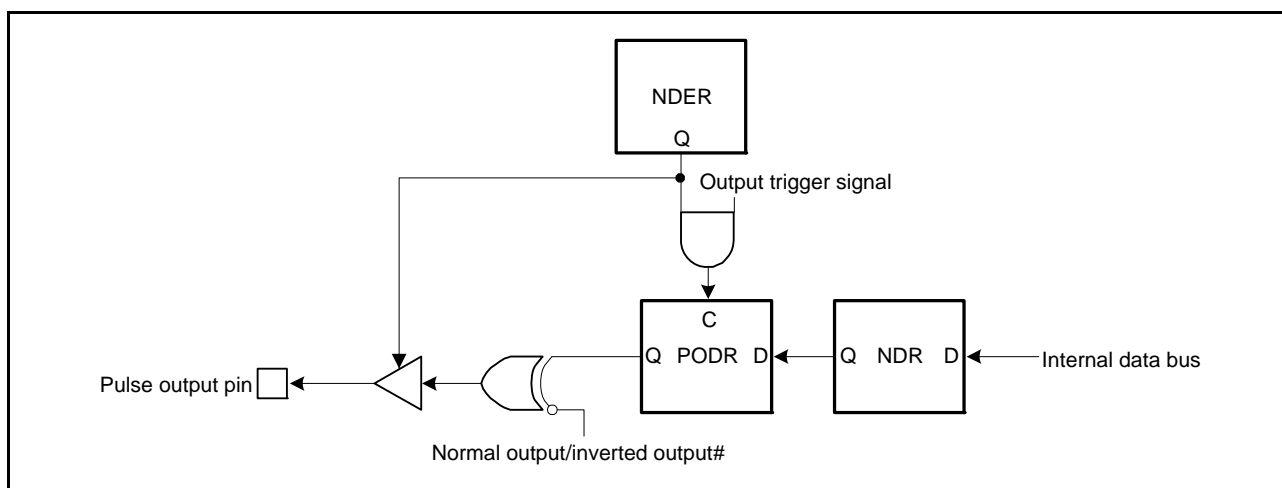


Figure 20.4 Schematic Diagram of PPG

20.3.1 Output Timing

When the selected compare match event occurs while pulse output is enabled, the values in PPGn.NDRH and PPGn.NDRL ($n = 0, 1$) are transferred to PPGn.PODRH and PPGn.PODRL, respectively, and then output on the corresponding pins.

Figure 20.5 shows the timing of the above operation. In this case, the timing when compare match A triggers normal output from groups 2 and 3 is shown.

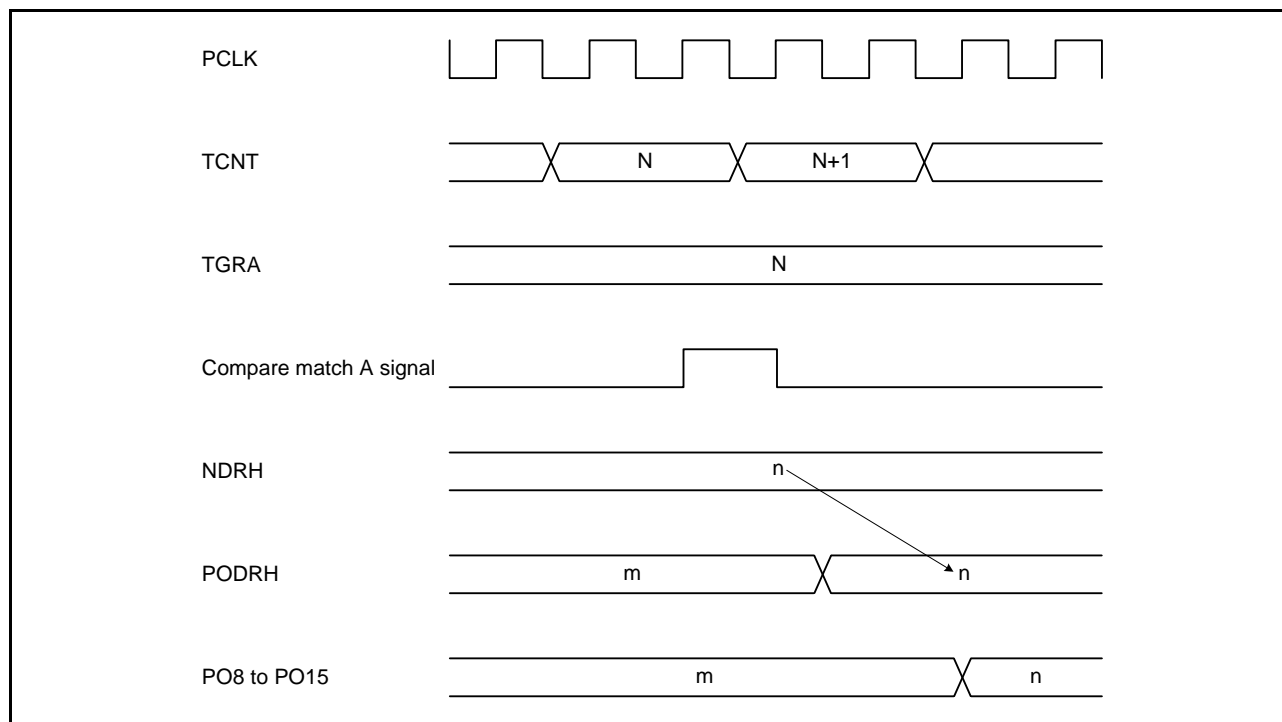


Figure 20.5 Timing of Transfer and Output of the Values in NDR (Example)

20.3.2 Sample Setup Procedure for Normal Pulse Output

Figure 20.6 and Figure 20.7 show sample procedures for setting normal pulse output.

(1) PPG0 Setting

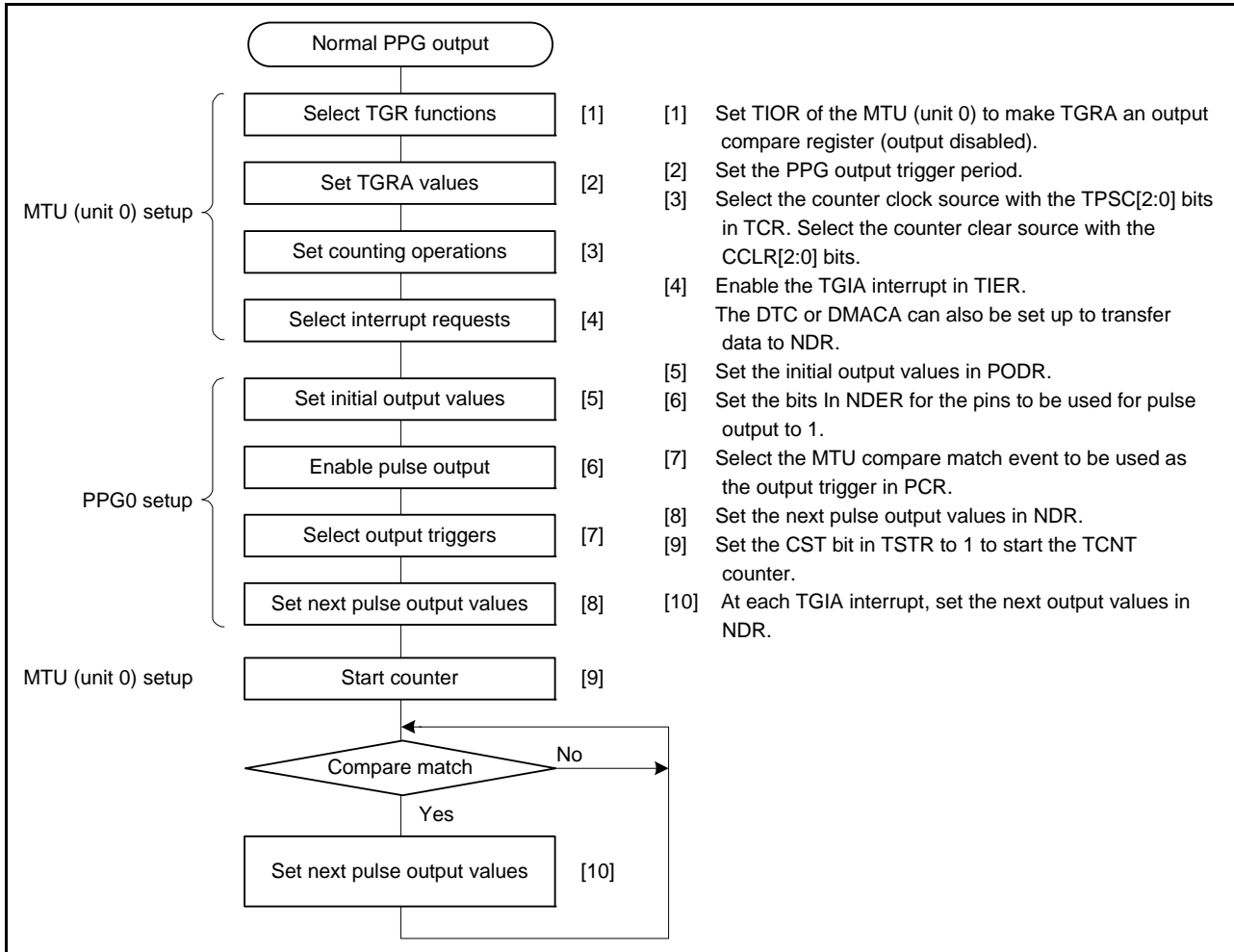


Figure 20.6 Sample Setup Procedure for Normal Pulse Output (PPG0 Setting)

(2) PPG1 Setting

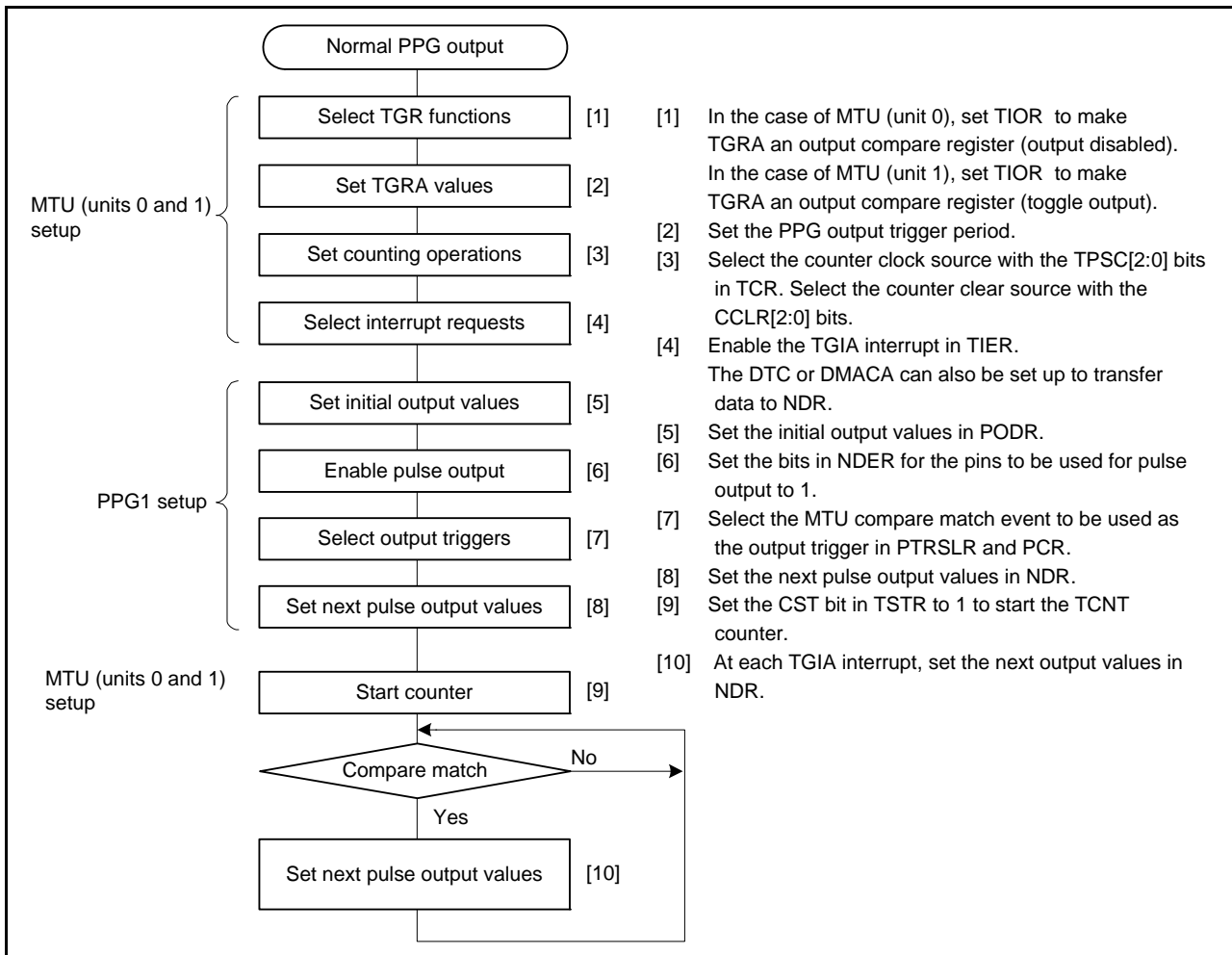


Figure 20.7 Sample Setup Procedure for Normal Pulse Output (PPG1 Setting)

20.3.3 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

Figure 20.8 shows an example in which pulse output from the PPG0 is used for cyclic five-phase pulse output.

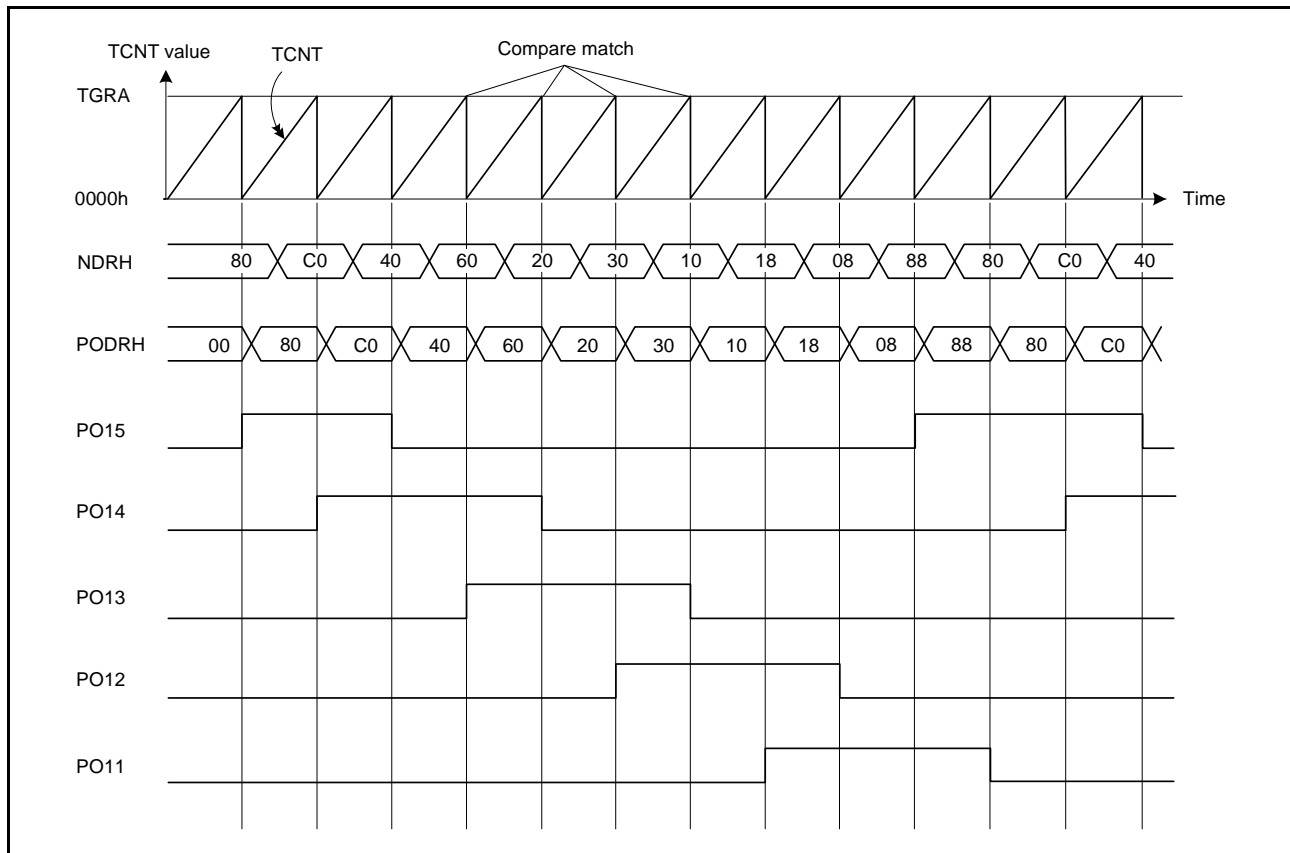


Figure 20.8 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

- Note 1. Set an output compare register of the MTU, i.e. MTUn.TGRA (n = 0 to 3) so that the corresponding compare match signal is the output trigger. Set a cycle in TGRA so that the counter will be cleared by compare match A. Set the TGIEA bit in MTUn.TIER to 1 to enable the compare match/input capture A (TGInA) interrupt.
- Note 2. Write F8h to PPG0.NDRH, and set the G3CMS[1:0] and G2CMS[1:0] bits in PPG0.PCR to select the respective compare matches in the MTUn selected in the previous step to be the output triggers. Write output data 80h to PPG0.NDRH.
- Note 3. The timer counter in the MTUn starts. When compare match A occurs, the values in PPG0.NDRH are transferred to PPG0.PODRH and output. The TGInA interrupt handling routine writes the next output data C0h to PPG0.NDRH.
- Note 4. Five-phase pulse output (one or two phases active at a time) can be obtained subsequently by writing 40h, 60h, 20h, 30h, 10h, 18h, 08h, 88h... at successive TGInA interrupts.
If the DTC or DMACA is set for activation by the TGInA interrupt, pulse output can be obtained without imposing a load on the CPU.

20.3.4 Non-Overlapping Pulse Output

During non-overlapping operation, data transfer from PPGn.NDRH and PPGn.NDRL (n = 0, 1) to PPGn.PODRH and PPGn. PODRL is performed as follows.

- On compare match A, the values in PPGn.NDRH and PPGn.NDRL are always transferred to PPGn.PODRH and PPGn. PODRL.
- On compare match B, data transfer proceeds for bits in PPGn.NDRH and PPGn.NDRL that have the value 0. It does not proceed for bits having the value 1.

Figure 20.9 shows the non-overlapping pulse output operation.

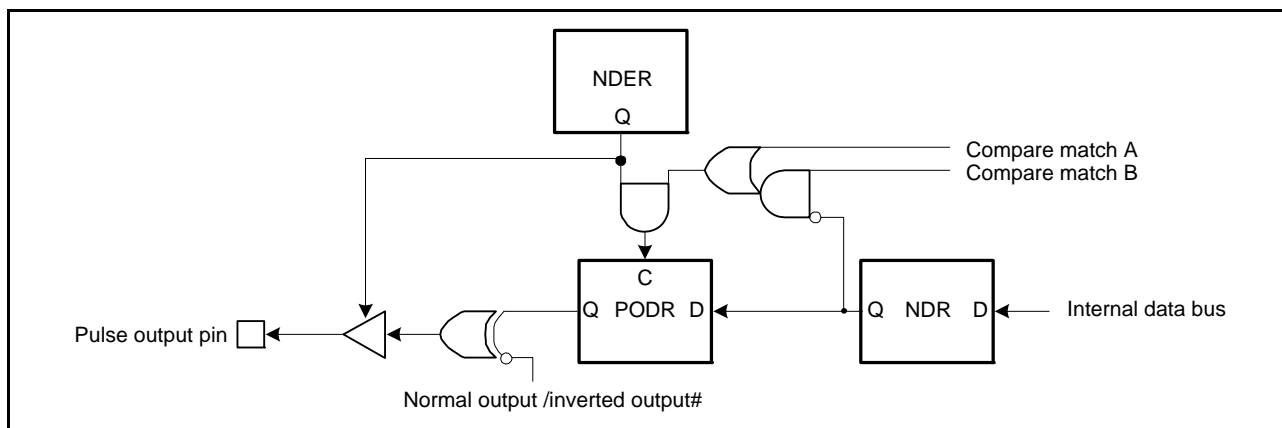


Figure 20.9 Non-Overlapping Pulse Output

Therefore, compare match B before compare match A allows 0-valued data to be transferred in advance of 1-valued data. Do not change the values in PPGn.NDRH and PPGn.NDRL during the interval from compare match B to compare match A (the non-overlap margin).

To transfer 0-valued data in advance of 1-valued data, write the next data to PPGn.NDRH and PPGn.NDRL from within the TGIA interrupt handling routine or by using a TGIA interrupt to activate transfer by the DTC or DMACA. In any case, the next data must be written before the next compare match B occurs.

Figure 20.10 shows the timing of the above procedure.

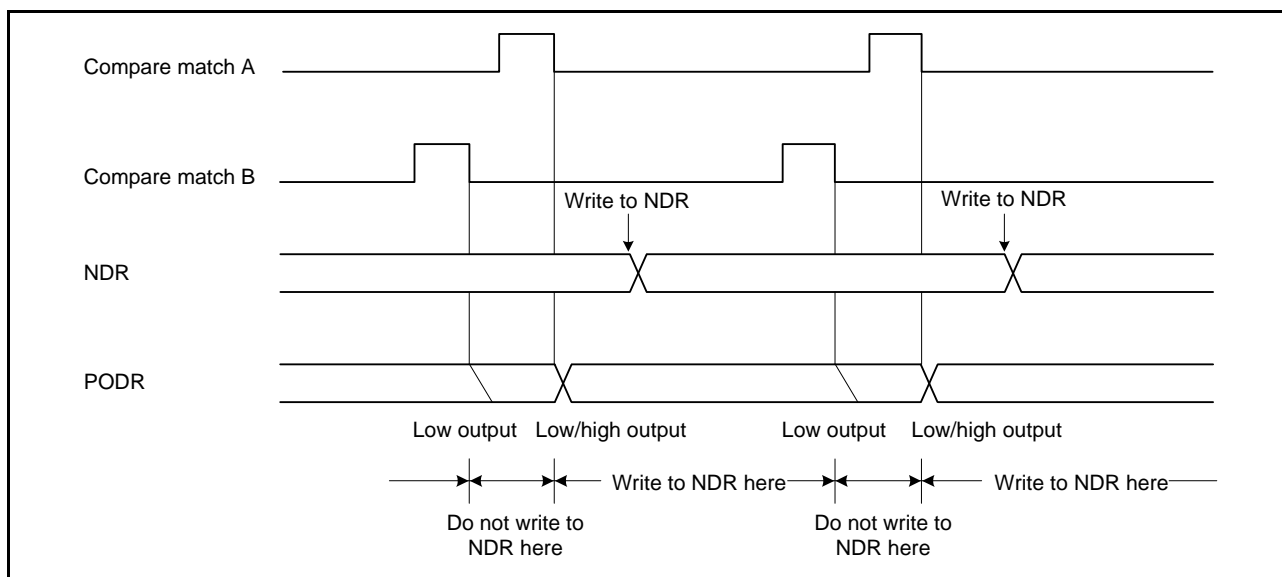


Figure 20.10 Non-Overlapping Operation and Write Timing to PPGn.NDRH and PPGn.NDRL

20.3.5 Sample Setup Procedure for Non-Overlapping Pulse Output

Figure 20.11 and Figure 20.12 show sample procedures for setting up non-overlapping pulse outputs.

(1) PPG0 Setting

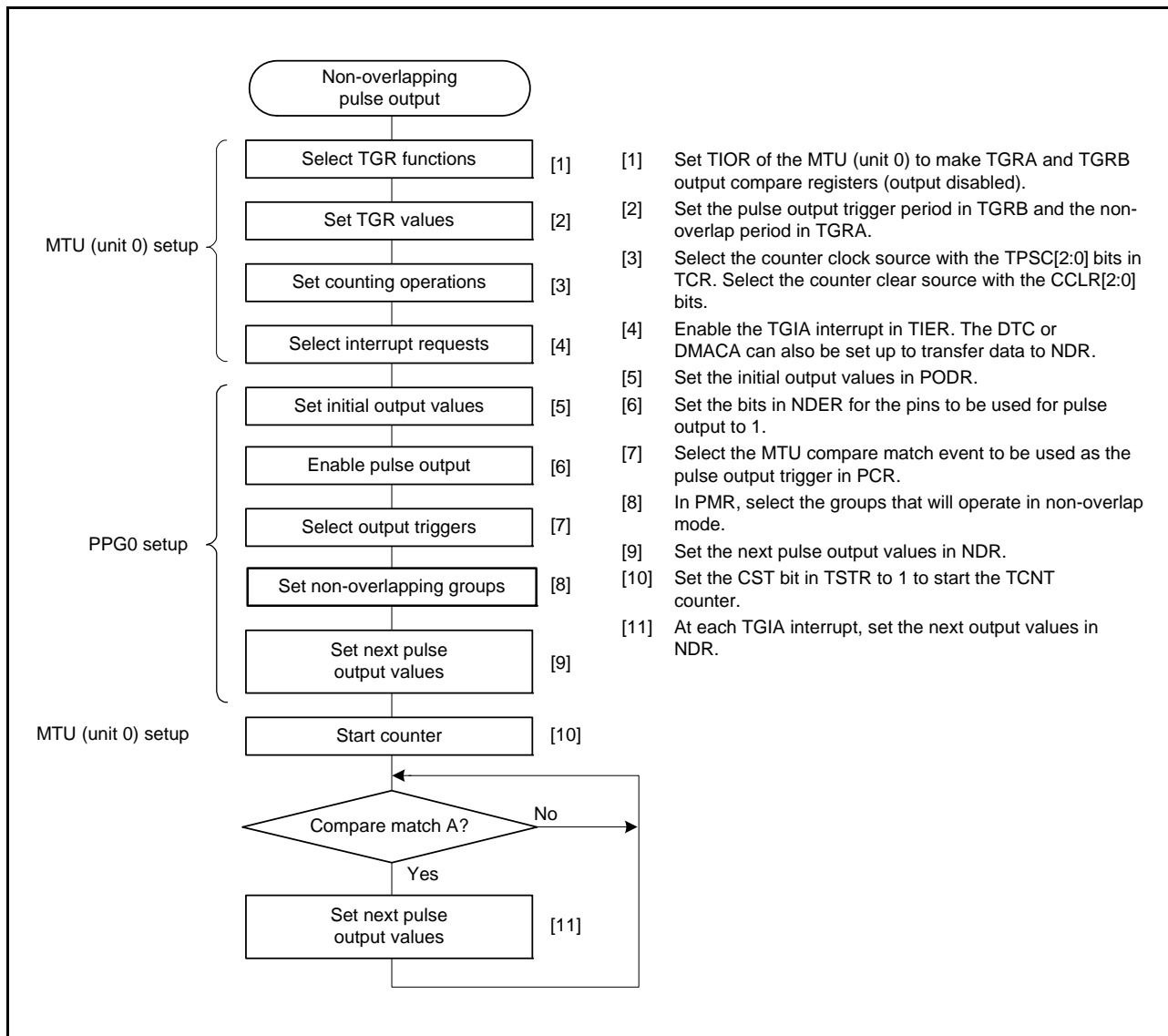


Figure 20.11 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG0 Setting)

(2) PPG1 Setting

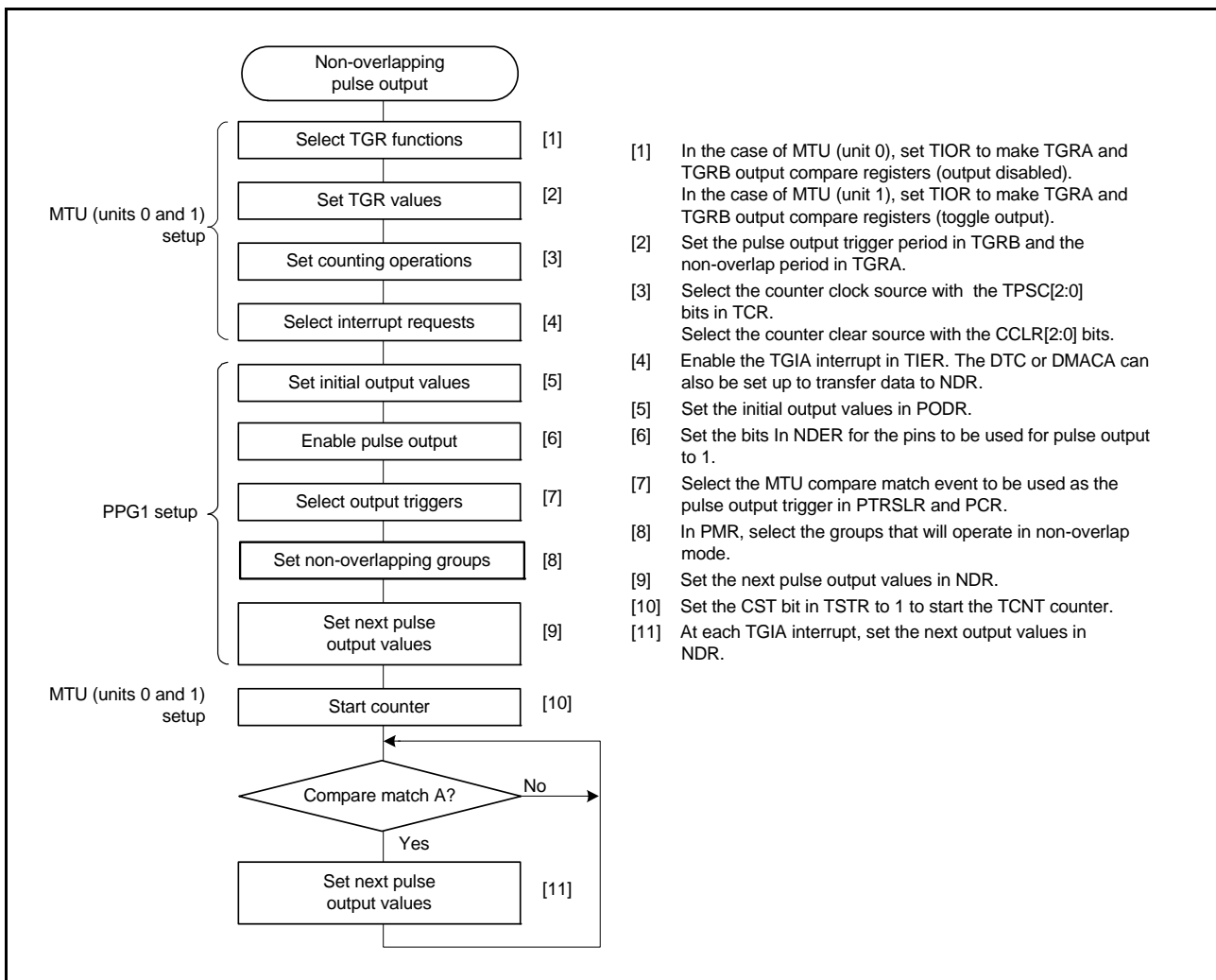


Figure 20.12 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG1 Setting)

20.3.6 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

Figure 20.13 shows an example in which pulse output from the PPG0 is used for four-phase complementary non-overlapping pulse output.

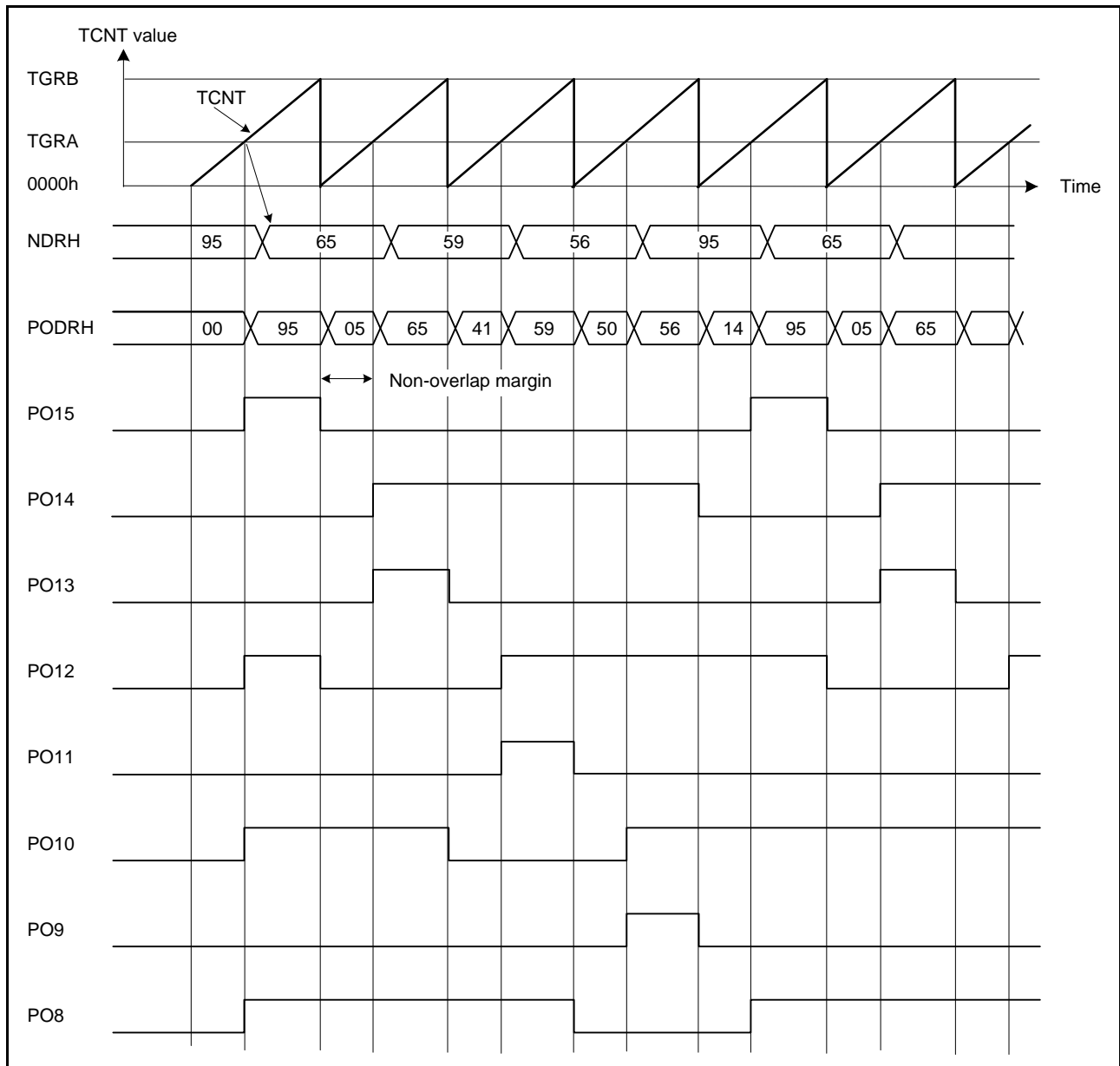


Figure 20.13 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

1. Set output compare registers of the MTU, i.e. MTUn.TGRA and MTUn.TGRB (n = 0 to 3) so that the corresponding compare match signals are the output triggers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared by compare match B. Set the TGIEA bit in MTUn.TIER to 1 to enable the compare match/input capture A (TGInA) interrupt.
2. Write FFh in PPG0.NDERH, and set the G3CMS[1:0] and G2CMS[1:0] bits in PPG0.PCR to select the respective compare matches in the MTUn selected in the previous step to be the output triggers.
Set the G3NOV and G2NOV bits in PPG0.PMR to 1 to select non-overlapping outputs. Write output data 95h in PPG0.NDRH.
3. The timer counter in the MTUn starts. When a compare match with TGRB occurs, outputs change from "high" to "low". When a compare match with TGRA occurs, outputs change from "low" to "high" (the change from "low" to "high" is delayed by the value set in TGRA).
The TGInA interrupt handling routine writes the next output data 65h in PPG0.NDRH.
4. Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing 59h, 56h, 95h... at successive TGInA interrupts.
If the DTC or DMACA is set for activation by the TGInA interrupt, pulse output can be obtained without imposing a load on the CPU.

20.3.7 Inverted Pulse Output

When the G3INV, G2INV, G1INV, and G0INV bits in PPG0.PMR are cleared to 0, the values that are the inverse of the respective values in PPG0.PODRH and PPG0.PODRL can be output.

Figure 20.14 shows the outputs when the G3INV and G2INV bits are cleared to 0 in addition to the settings in Figure 20.13.

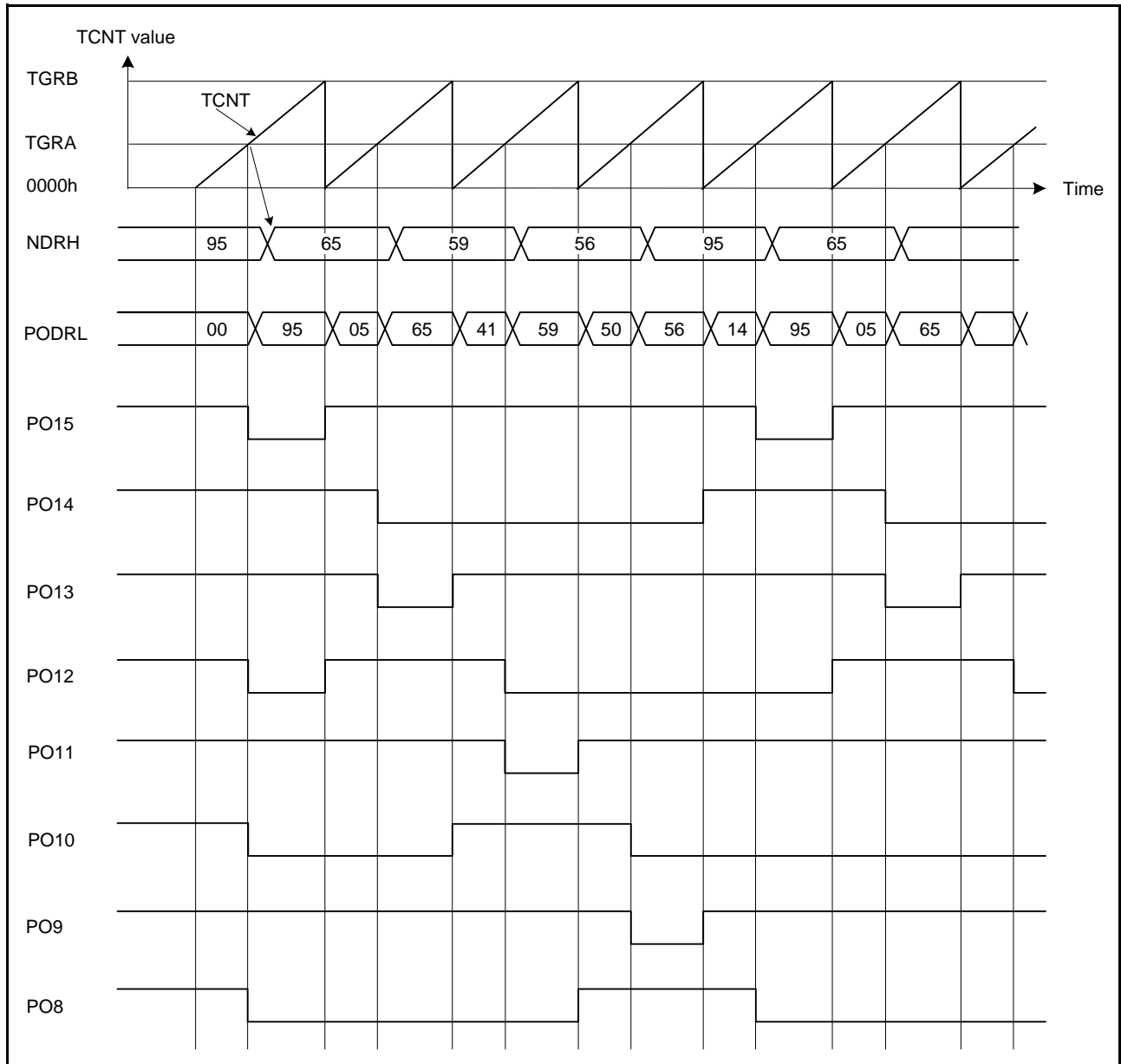


Figure 20.14 Inverted Pulse Output (Example)

20.3.8 Pulse Output Triggered by Input Capture

Pulse output from the PPG0 can be triggered by the MTU (unit 0) input capture as well as by compare match. When MTUn.TGRA (n = 0 to 3) functions as an input capture register in the MTU (unit 0) channel selected by PPG0.PCR, pulse output is triggered by the input capture signal.

Figure 20.15 shows the timing of pulse output triggered by input capture.

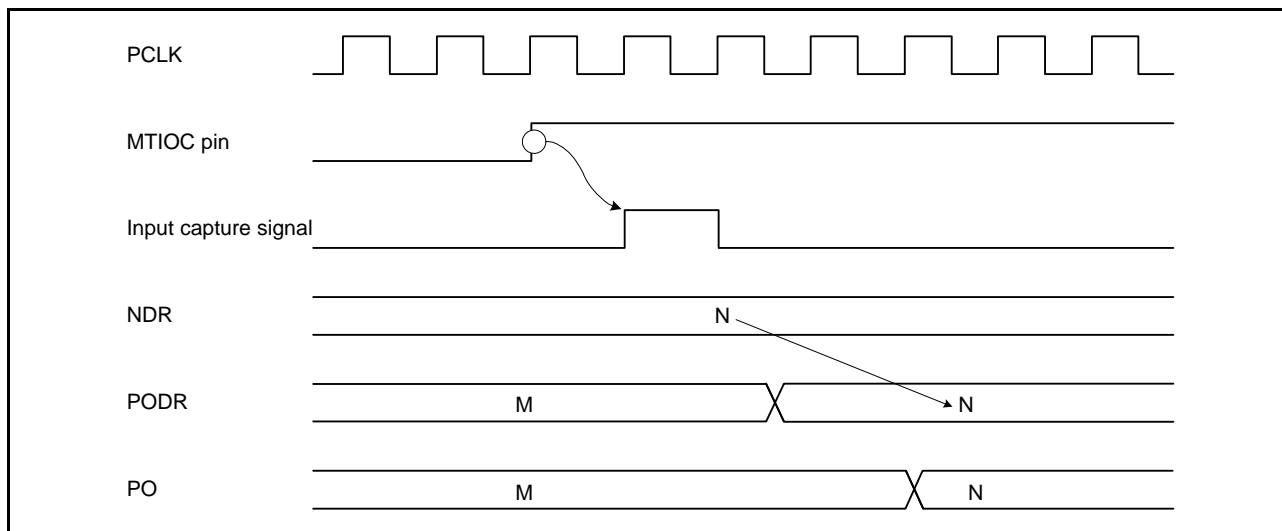


Figure 20.15 Timing of Pulse Output Triggered by Input Capture (Example)

20.4 Usage Note

20.4.1 Module Stop Function Setting

Operation of the PPG can be disabled or enabled by the module stop control register. The initial setting is for operation of the PPG to be halted. Register access is enabled by clearing module stop state. For details, see [section 9, Low Power Consumption](#).

21. 8-Bit Timer (TMR)

The RX62N/RX621 Group has two units (unit 0, unit 1) of an on-chip 8-bit timer (TMR) module that comprise two 8-bit counter channels, totaling four channels. The 8-bit timer module can be used to count external events and also be used as a multifunction timer in a variety of applications, such as generation of counter reset, interrupt requests, and pulse output with a desired duty cycle using a compare-match signal with two registers.

Unit 0 and unit 1 can generate a baud rate clock signal for the SCI and have the same functions.

21.1 Overview

Table 21.1 shows the specifications of the TMR.

Figure 21.1 shows a block diagram of the 8-bit timer module (unit 0), and Figure 21.2 shows that of the 8-bit timer module (unit 1).

Table 21.1 Specifications of TMR

Item	Description
Count clock	<ul style="list-style-type: none"> Internal clock: PCLK, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192 External clock
Number of channels	(8 bits x 2 channels) x 2 units
Compare match	<ul style="list-style-type: none"> 8-bit mode (compare match A, compare match B) 16-bit mode (compare match A, compare match B)
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"> 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) Compare match count mode TMR1 can be used to count TMR0 compare matches (TMR3 can be used to count TMR2 compare matches).
Interrupt sources	Compare match A, compare match B, and overflow
DTC activation	DTC can be activated by compare match A interrupts or compare match B interrupts.
Generation of trigger to start A/D converter conversion	Compare match A of TMR0 and TMR2*1
Capable of generating baud rate clock for SCI	Generates baud rate clock for SCI5 and SCI6.*2
Low power consumption facilities	Each unit can be placed in a module stop state.

Note 1. For details, see section 34, 12-Bit A/D Converter (S12AD), and section 35, 10-Bit A/D Converter (ADa).

Note 2. For details, see section 29, Serial Communications Interface (SCIa).

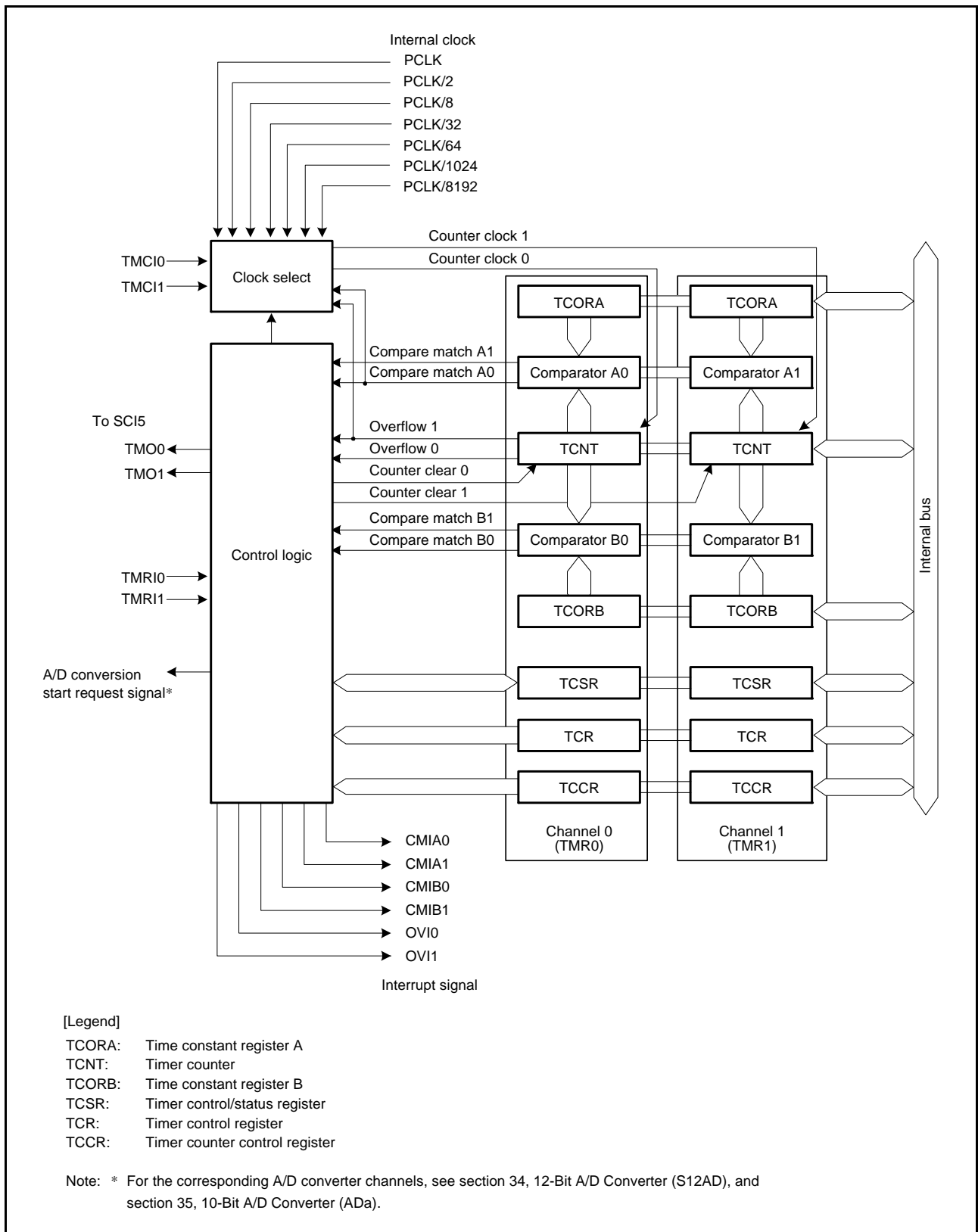


Figure 21.1 Block Diagram of TMR (Unit 0)

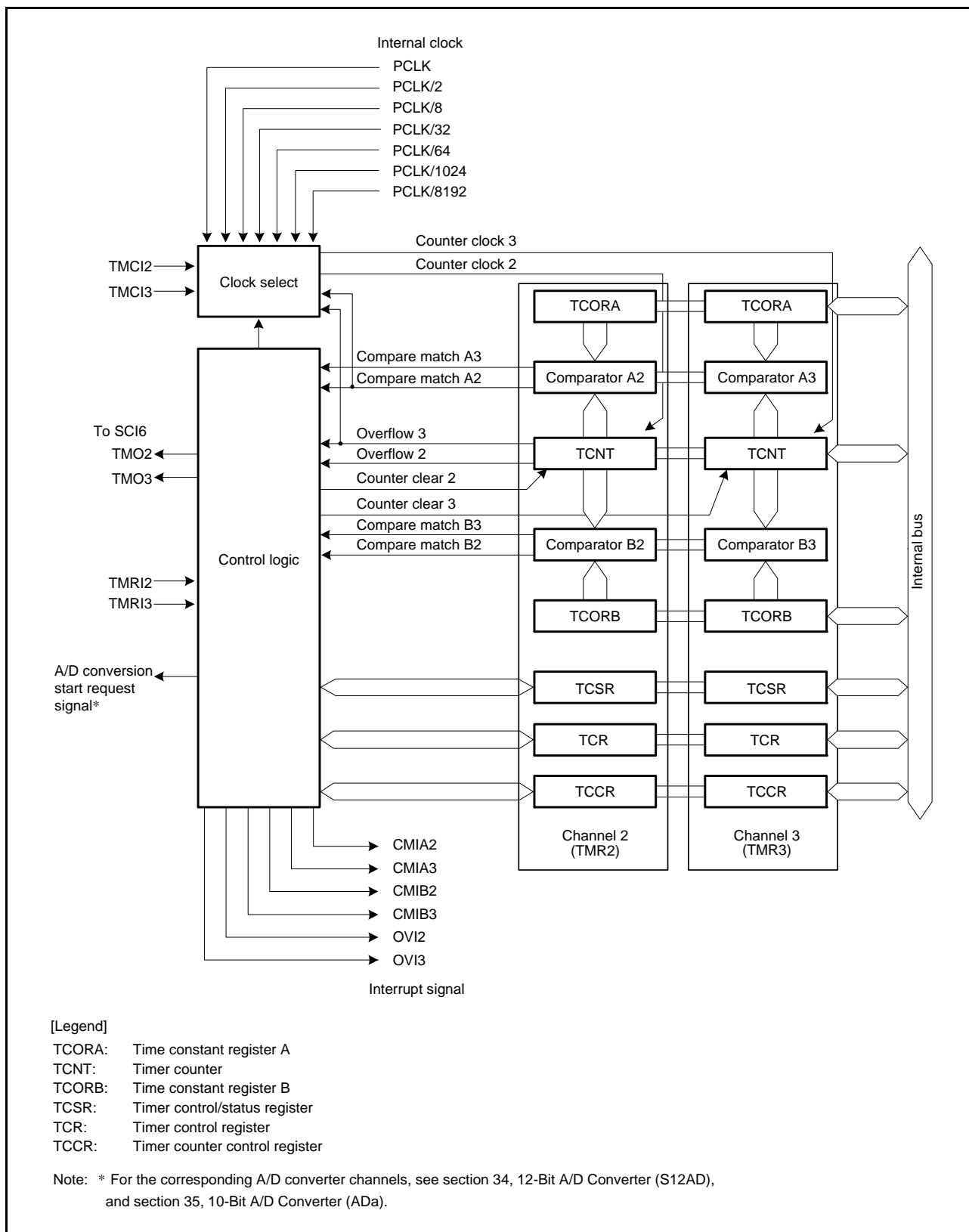


Figure 21.2 Block Diagram of TMR (Unit 1)

Table 21.2 lists the input/output pins of the TMR.

Table 21.2 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
		TMRI0	Input	Inputs external reset to counter
	TMR1	TMO1	Output	Outputs compare match
		TMC11	Input	Inputs external clock for counter
		TMRI1	Input	Inputs external reset to counter
1	TMR2	TMO2	Output	Outputs compare match
		TMC12	Input	Inputs external clock for counter
		TMRI2	Input	Inputs external reset to counter
	TMR3	TMO3	Output	Outputs compare match
		TMC13	Input	Inputs external clock for counter
		TMRI3	Input	Inputs external reset to counter

21.2 Register Descriptions

Table 21.3 shows the registers of the TMR.

Table 21.3 Registers of TMR

Unit	Channel	Register Name	Symbol	Value after Reset	Address*	Access Size
0	TMR0	Timer counter	TCNT	00h	0008 8208h	8 or 16
		Time constant register A	TCORA	FFh	0008 8204h	8 or 16
		Time constant register B	TCORB	FFh	0008 8206h	8 or 16
		Timer control register	TCR	00h	0008 8200h	8
		Timer counter control register	TCCR	00h	0008 820Ah	8 or 16
		Timer control/status register	TCSR	xxx0 0000b	0008 8202h	8
	TMR1	Timer counter	TCNT	00h	0008 8209h	8 or 16*
		Time constant register A	TCORA	FFh	0008 8205h	8 or 16*
		Time constant register B	TCORB	FFh	0008 8207h	8 or 16*
		Timer control register	TCR	00h	0008 8201h	8
		Timer counter control register	TCCR	00h	0008 820Bh	8 or 16*
		Timer control/status register	TCSR	xxx0 0000b	0008 8203h	8
1	TMR2	Timer counter	TCNT	00h	0008 8218h	8 or 16
		Time constant register A	TCORA	FFh	0008 8214h	8 or 16
		Time constant register B	TCORB	FFh	0008 8216h	8 or 16
		Timer control register	TCR	00h	0008 8210h	8
		Timer counter control register	TCCR	00h	0008 821Ah	8 or 16
		Timer control/status register	TCSR	xxx0 0000b	0008 8212h	8
	TMR3	Timer counter	TCNT	00h	0008 8219h	8 or 16*
		Time constant register A	TCORA	FFh	0008 8215h	8 or 16*
		Time constant register B	TCORB	FFh	0008 8217h	8 or 16*
		Timer control register	TCR	00h	0008 8211h	8
		Timer counter control register	TCCR	00h	0008 821Bh	8 or 16*
		Timer control/status register	TCSR	xxx0 0000b	0008 8213h	8

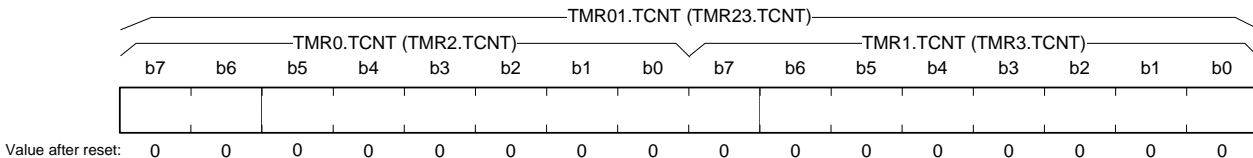
Note : * Odd addresses should not be accessed in 16-bit units. When accessing a register in 16-bit units, access the address of the TMR0 or TMR2 register. Table 21.4 lists register allocation for 16-bit access.

Table 21.4 Register Allocation and Symbol for 16-Bit Access

Address	Upper 8 Bits	Lower 8 Bits	Symbol for 16-Bit Access
0008 8208h	TMR0.TCNT	TMR1.TCNT	TMR01.TCNT
0008 8204h	TMR0.TCORA	TMR1.TCORA	TMR01.TCORA
0008 8206h	TMR0.TCORB	TMR1.TCORB	TMR01.TCORB
0008 820Ah	TMR0.TCCR	TMR1.TCCR	TMR01.TCCR
0008 8218h	TMR2.TCNT	TMR3.TCNT	TMR23.TCNT
0008 8214h	TMR2.TCORA	TMR3.TCORA	TMR23.TCORA
0008 8216h	TMR2.TCORB	TMR3.TCORB	TMR23.TCORB
0008 821Ah	TMR2.TCCR	TMR3.TCCR	TMR23.TCCR

21.2.1 Timer Counter (TCNT)

Addresses: TMR0.TCNT 0008 8208h, TMR1.TCNT 0008 8209h
 TMR2.TCNT 0008 8218h, TMR3.TCNT 0008 8219h
 TMR01.TCNT 0008 8208h, TMR23.TCNT 0008 8218h



TCNT is an 8-bit readable/writable up-counter.

Counters TMR0.TCNT and TMR1.TCNT (or counters TMR2.TCNT and TMR3.TCNT) are also accessible in word units as a 16-bit counter (TMR01.TCNT or TMR23.TCNT).

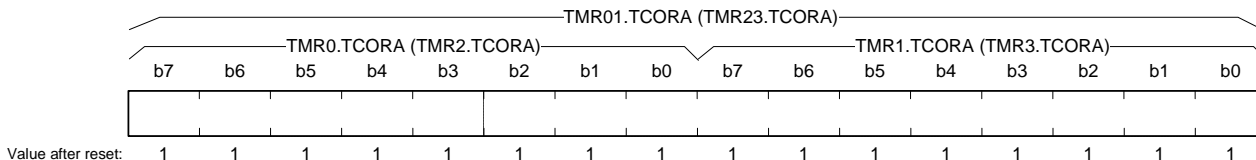
The TCCR.CSS[1:0] and CKS[2:0] bits are used to select a clock.

TCNT can be cleared by an external reset input signal, compare match A signal, or compare match B signal. Which signal to be used for clearing is selected by the TCR.CCLR[1:0] bits.

When TCNT overflows from FFh to 00h, the interrupt flag is set to 1. For details on the corresponding interrupt vector number, see section 11, Interrupt Control Unit (ICUa), and Table 21.6, TMR Interrupt Sources.

21.2.2 Time Constant Register A (TCORA)

Addresses: TMR0.TCORA: 0008 8204h, TMR1.TCORA: 0008 8205h
 TMR2.TCORA: 0008 8214h, TMR3.TCORA: 0008 8215h
 TMR01.TCORA: 0008 8204h, TMR23.TCORA: 0008 8214h



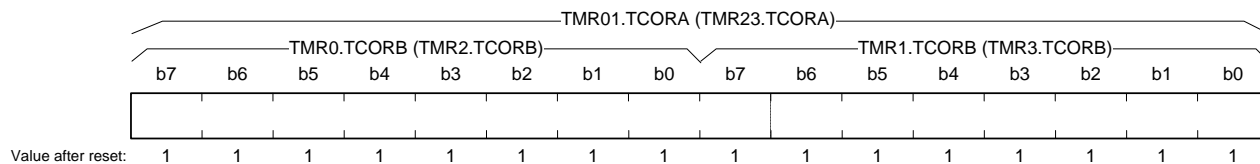
TCORA is an 8-bit readable/writable register.

Registers TMR0.TCORA and TMR1.TCORA (or registers TMR2.TCORA and TMR3.TCORA) are also accessible in word units as a 16-bit register (TMR01.TCORA or TMR23.TCNT).

The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare match A signal is set to “high”. Note however that comparison is not performed during writing to TCORA. The timer output from the TMO_n pin can be freely controlled by this compare match A signal and the settings of the TCSR.OSA[1:0] bits.

21.2.3 Time Constant Register B (TCORB)

Addresses: TMR0.TCORB 0008 8206h, TMR1.TCORB 0008 8207h
 TMR2.TCORB 0008 8216h, TMR3.TCORB 0008 8217h
 TMR01.TCORA 0008 8204h, TMR23.TCORA 0008 8214h



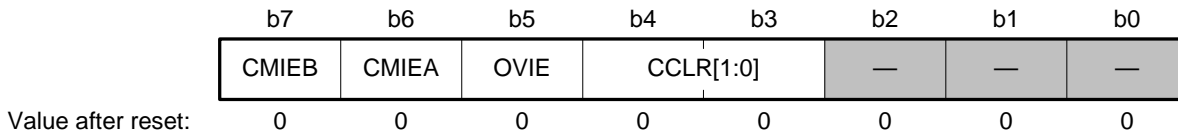
TCORB is an 8-bit readable/writable register.

Registers TMR0.TCORB and TMR1.TCORB (or registers TMR2.TCORB and TMR3.TCORB) are also accessible in word units as a 16-bit register (TMR01.TCORB or TMR23.TCORB).

The value in TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare match B signal is set to “high”. Note however that comparison is not performed during writing to TCORB. The timer output from the TMO pin can be freely controlled by this compare match B signal and the settings of the TCSR.OSB[1:0] bits.

21.2.4 Timer Control Register (TCR)

Addresses: 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 always read as 0. The write value should always be 0.	R/W
b4, b3	CCLR[1:0]	Counter Clear*	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 _n) are disabled 1: Compare match A interrupt requests (CMIA _n) are enabled	R/W
b7	CMIEB	Compare Match Interrupt Enable B	0: Compare match B interrupt requests (CMIB _n) are disabled 1: Compare match B interrupt requests (CMIB _n) are enabled	R/W

Note: * To use an external reset, set the Pn.DDR.Bi bit for the corresponding pin to "0" and the Pn.ICR.Bi bit to "1". For details, see section 17, I/O Ports

TCR specifies the condition for clearing TCNT.

CCLR[1:0] Bits (Counter Clear)

Select the method 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)

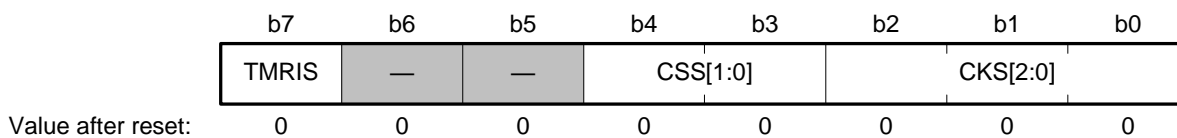
Selects whether compare match A interrupt requests (CMIA_n) that are issued when the value of TCORA corresponds to that of TCNT are enabled or disabled.

CMIEB Bit (Compare Match Interrupt Enable B)

Selects whether compare match B interrupt requests (CMIB_n) that are issued when the value of TCORB corresponds to that of TCNT are enabled or disabled.

21.2.5 Timer Counter Control Register (TCCR)

Addresses: TMR0.TCCR 0008 820Ah, TMR1.TCCR 0008 820Bh
 TMR2.TCCR 0008 821Ah, TMR3.TCCR 0008 821Bh
 TMR01.TCCR 0008 820Ah, TMR23.TCCR 0008 821A



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKS[2:0]	Clock Select*	See Table 21.5.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 21.5.	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always 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 : * To use an external clock, set the Pn.DDR.Bi bit for the corresponding pin to "0" and the PnICR.Bi bit to "1". For details, see section 17, I/O Ports.

TCCR selects an internal clock source for TCNT and the condition for detecting external reset.

Registers TMR0.TCCR and TMR1.TCCR (or registers TMR2.TCCR and TMR3.TCCR) are also accessible in word units as a 16-bit register (TMR01.TCCR or TMR23.TCCR).

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 21.5.

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 21.5 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.
					0	Uses external clock. Counts at falling edge*1.
	0	1	0	0	1	Uses external clock. Counts at both rising and falling edges*1.
					0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
					0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
					0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
					0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
					1	Setting prohibited
					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.
					0	Uses external clock. Counts at falling edge*1.
	0	1	0	0	1	Uses external clock. Counts at both rising and falling edges*1.
					0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
					0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
					0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
					0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
					1	Setting prohibited
					1	Counts at TMR0.TCNT (TMR2.TCNT) compare match A*2.

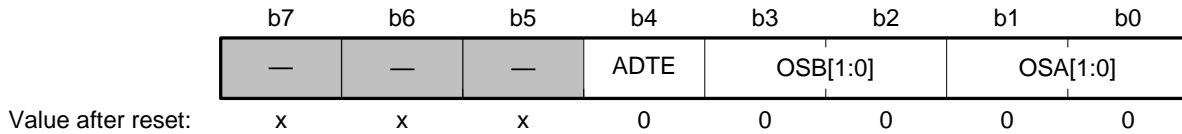
Note 1. To use an external clock, set the Pn.DDR.Bi bit for the corresponding pin to "0" and the Pn.ICR.Bi bit to "1". For details, see section 17, I/O Ports.

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.

21.2.6 Timer Control/Status Register (TCSR)

- TMR0.TCSR, TMR2.TCSR

Addresses: TMR0.TCSR 0008 8202h, TMR2.TCSR 0008 8212h



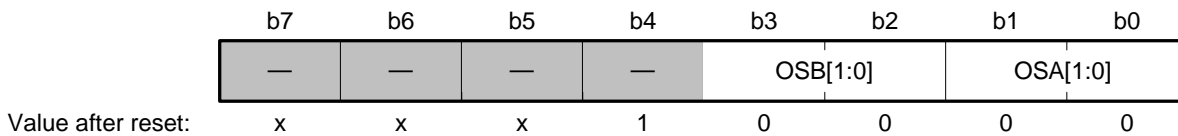
Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change when compare match A occurs 0 1: "Low" is output when compare match A occurs 1 0: "High" is output when compare match A occurs 1 1: Output is inverted when compare match A occurs (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change when compare match B occurs 0 1: "Low" is output when compare match B occurs 1 0: "High" is output when compare match B occurs 1 1: Output is inverted when compare match B occurs (toggle output)	R/W
b4	ADTE	A/D Trigger Enable*2	0: A/D converter start requests by compare match A are disabled 1: A/D converter start requests by compare match A are enabled	R/W
b7 to b5	—	Reserved	These bits are always read as an indefinite value. The write value should always be 1.	R/W

Note 1. Timer output is disabled when the OSB[1:0] and OSA[1:0] bits are all 0. Timer output is 0 until the first compare match occurs after a reset.

Note 2. For the corresponding A/D converter channels, see section 34, 12-Bit A/D Converter (S12AD), and section 35, 10-Bit A/D Converter (ADa).

- TMR1.TCSR, TMR3.TCSR

Addresses: TMR1.TCSR 0008 8203h, TMR3.TCSR 0008 8213h



[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*	b1 b0 0 0: No change when compare match A occurs 0 1: "Low" is output when compare match A occurs 1 0: "High" is output when compare match A occurs 1 1: Output is inverted when compare match A occurs (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*	b3 b2 0 0: No change when compare match B occurs 0 1: "Low" is output when compare match B occurs 1 0: "High" is output when compare match B occurs 1 1: Output is inverted when compare match B occurs (toggle output)	R/W
b4	—	Reserved	These bits are always read as 1. The write value should always be 1.	R/W
b7 to b5	—	Reserved	These bits are always read as an indefinite value. The write value should always be 1.	R/W

Note : * Timer output is disabled when the OSB[1:0] and OSA[1:0] bits are all 0. Timer output is 0 until the first compare match occurs after a reset.

TCSR controls compare match output.

OSA[1:0] Bits (Output Select A)

These bits select a method of TMO pin output when compare match A of TCORA and TCNT occurs.

OSB[1:0] Bits (Output Select B)

These bits select a method of TMO pin output when compare match B of TCORB and TCNT occurs.

ADTE Bit (A/D Trigger Enable)

Selects enabling or disabling of A/D converter start requests by compare match A.

This bit is reserved for TMR1.TCSR and TMR3.TCSR.

21.3 Operation

21.3.1 Pulse Output

Figure 21.3 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” until the first compare match occurs after a reset.

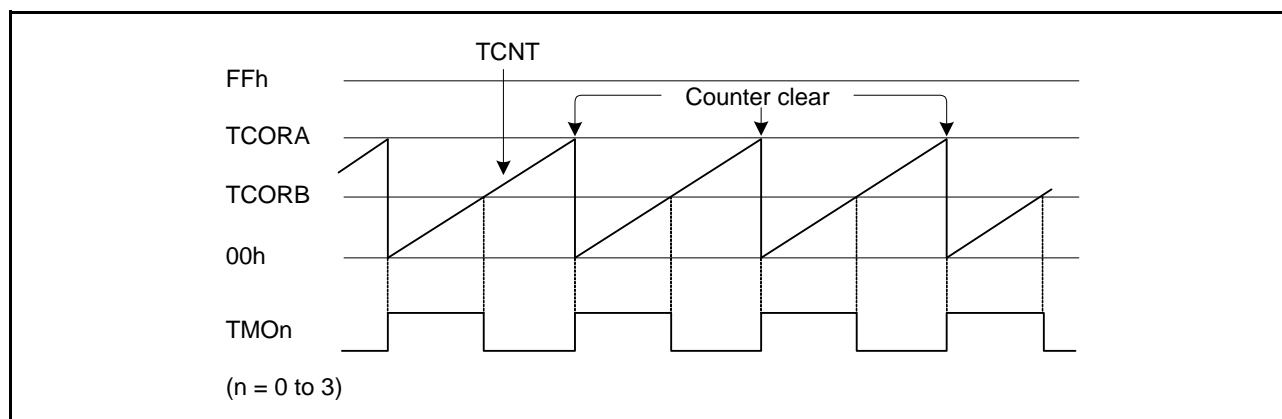


Figure 21.3 Example of Pulse Output

21.3.2 Reset Input

Figure 21.4 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.

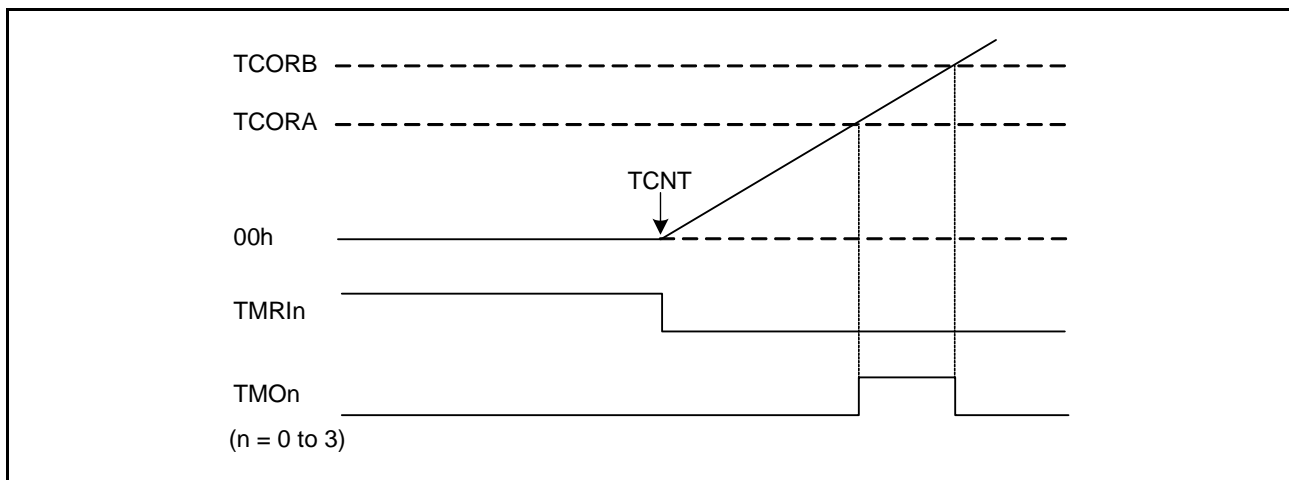


Figure 21.4 Example of Reset Input

21.4 Operation Timing

21.4.1 TCNT Count Timing

Figure 21.5 shows the count timing of TCNT for internal clock input. Figure 21.6 shows the count timing of TCNT for external clock input.

Note that the external clock pulse width must be at least 1.5 states for increment at a single edge, and at least 2.5 states for increment at both edges. The counter will not increment correctly if the pulse width is less than these values.

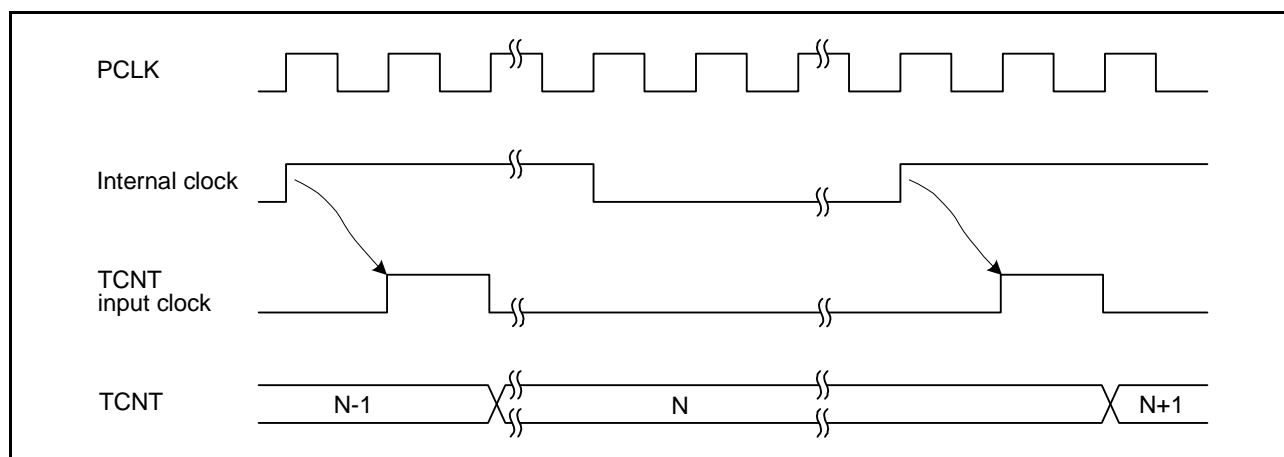


Figure 21.5 Count Timing for Internal Clock Input

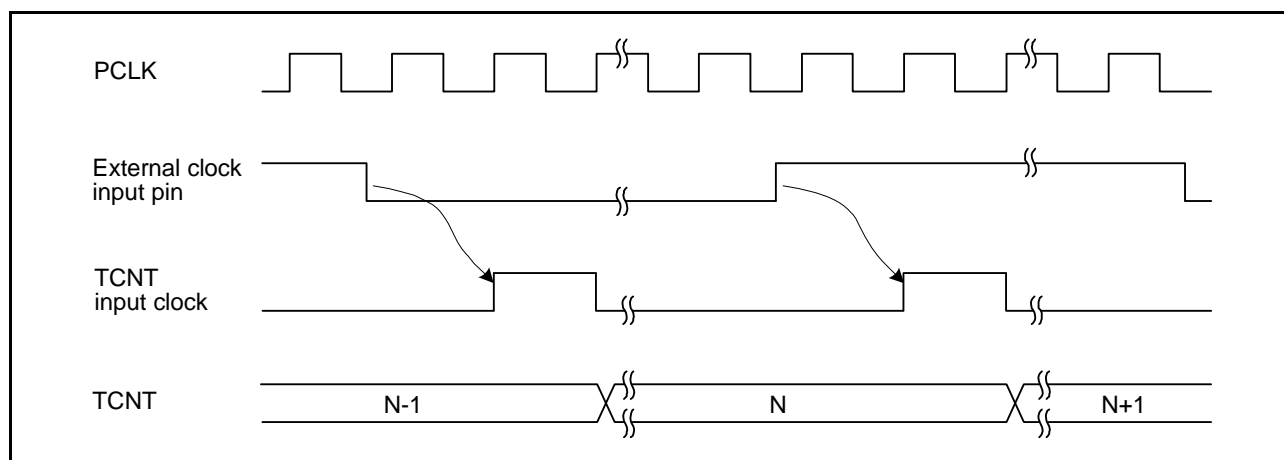


Figure 21.6 Count Timing for External Clock Input

21.4.2 Timing of the Interrupt for a Compare-Match

When a compare-match signal is output in response to a match between register TCORA or TCORB and the counter (TCNT), an interrupt signal is also produced.

The compare match signal is generated, just before the timer counter is updated. Therefore, when values of TCORA and TCORB and that of TCNT match, the compare match signal is not generated until the next TCNT clock input.

Figure 21.7 shows the timing of the interrupt for a compare-match. For details on the corresponding interrupt vector number, see section 11, Interrupt Control Unit (ICUa) and Table 21.6, TMR Interrupt Sources.

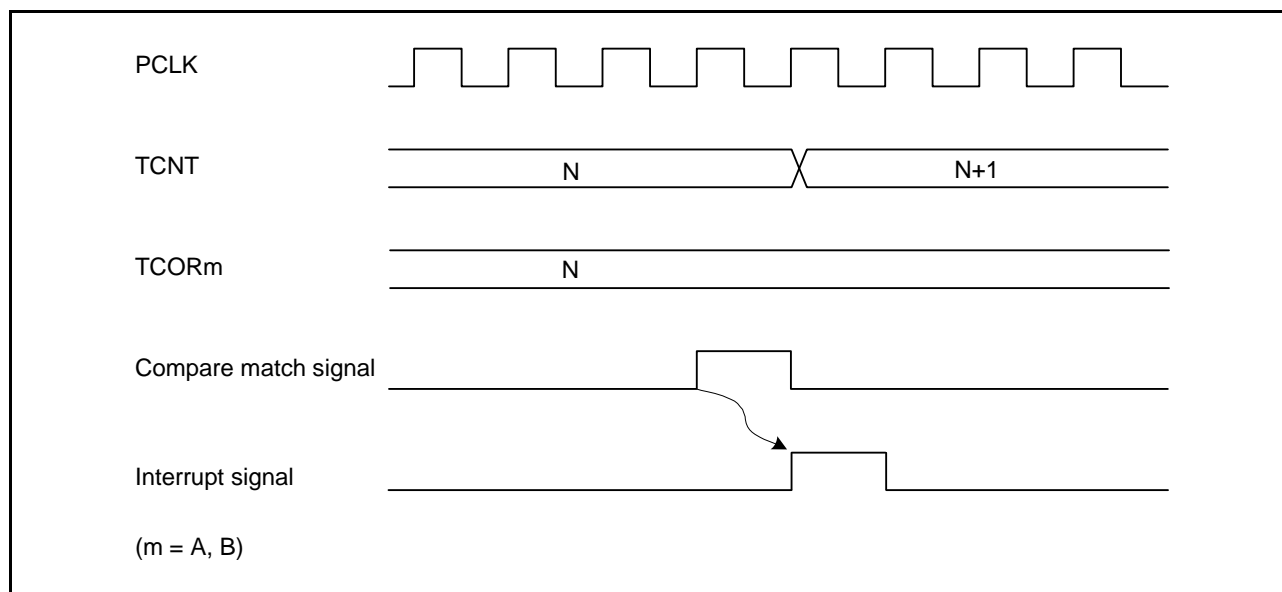


Figure 21.7 Timing of the Interrupt for a Compare-Match

21.4.3 Timing of Timer Output at Compare Match

When a compare match signal is generated, the timer output changes as specified by the TCSR.OSA[1:0] and OSB[1:0] bits.

Figure 21.8 shows the timing when the timer output is toggled by the compare match A signal.

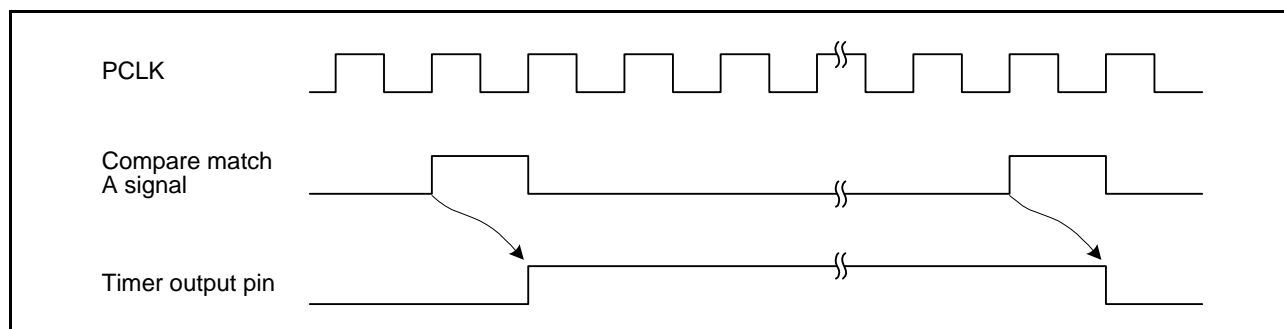


Figure 21.8 Timing of Timer Output at Compare Match A

21.4.4 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 21.9 shows the timing of this operation.

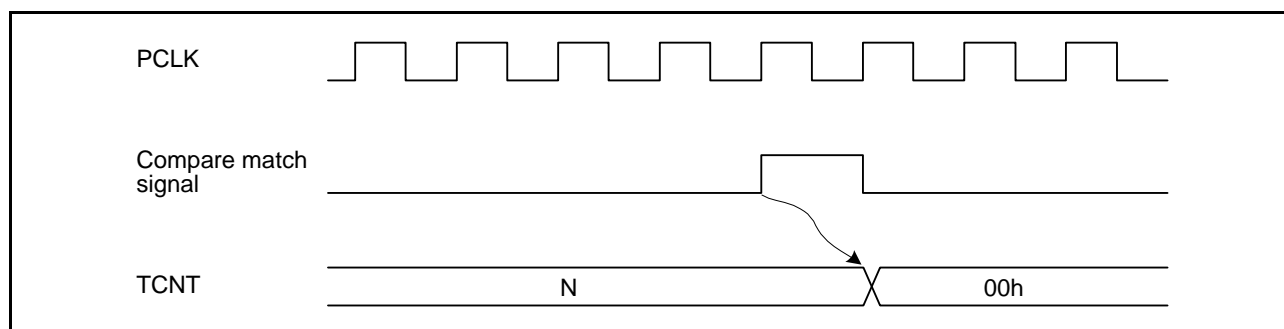


Figure 21.9 Timing of Counter Clear by Compare Match

21.4.5 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 cycles are required from an external reset input to clearing of TCNT.

Figure 21.10 and Figure 21.11 show the timing of this operation.

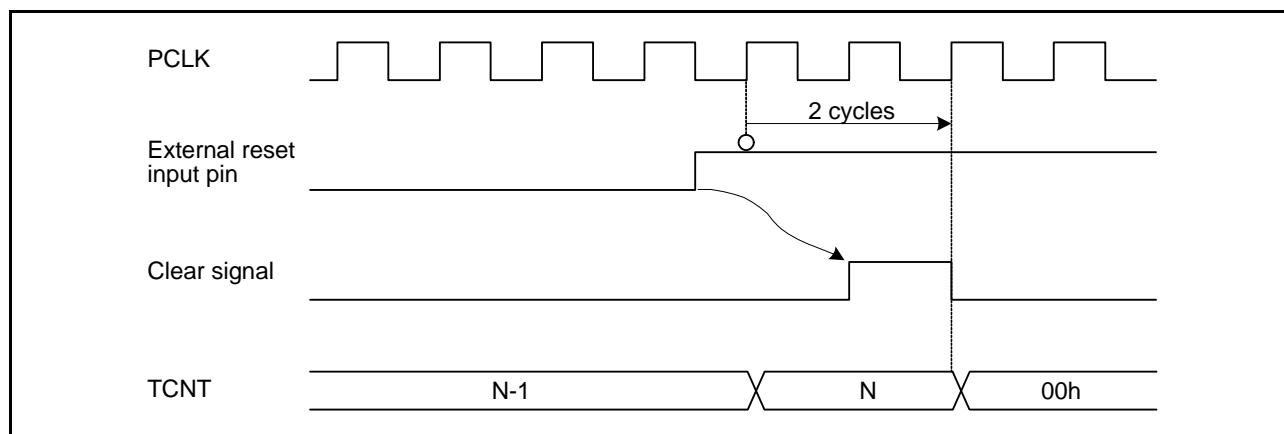


Figure 21.10 Timing of Clearance by External Reset (Rising Edge)

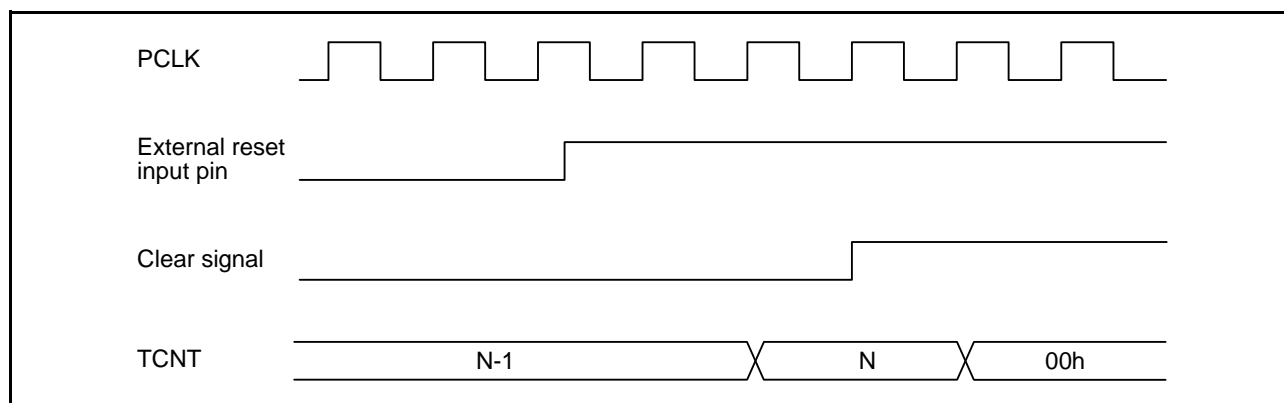


Figure 21.11 Timing of Clearance by External Reset (High Level)

21.4.6 Timing of the Interrupt for an Overflow

The interrupt signal is output by an overflow signal outputted when TCNT overflows (changes from FFh to 00h).

Figure 21.12 shows the interrupt timing by an overflows. For details on the corresponding interrupt vector number, see section 11, Interrupt Control Unit (ICUa) and Table 21.6, TMR Interrupt Sources.

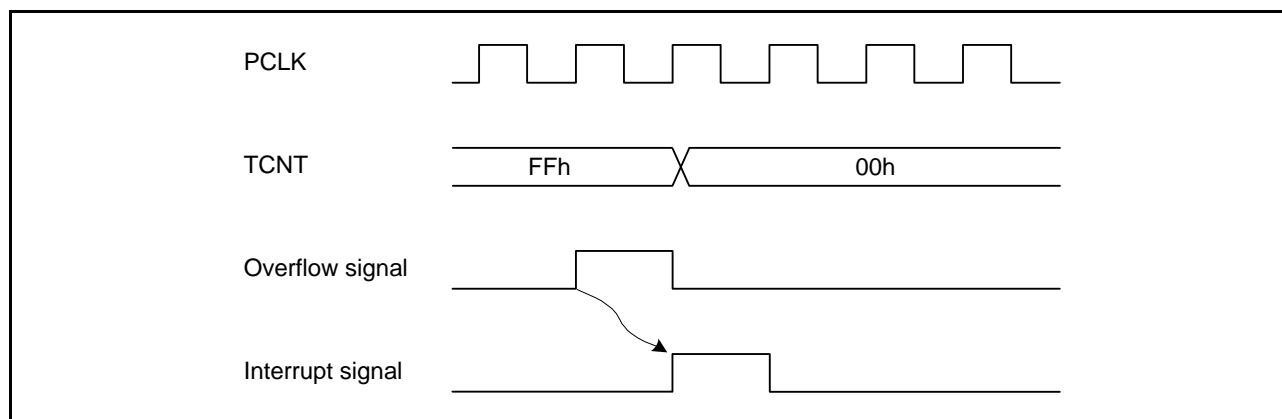


Figure 21.12 Timing of the Interrupt for an Overflow

21.5 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.

21.5.1 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. Only in this mode, registers whose Access Size column in Table 21.3 is 16 can be accessed in 16-bit units.

(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.

21.5.2 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_n (n = 0, 1) pin, and counter clear are in accordance with the settings for each channel.

21.6 Interrupt Sources

21.6.1 Interrupt Sources and DTC Activation

There are three interrupt sources for TMRn: CMIA_n, CMIB_n, and OVI_n. Their interrupt sources and priorities are listed in Table 21.6.

It is also possible to activate the DTC by means of CMIA_n and CMIB_n interrupts. The DMACA cannot be activated by the interrupt sources for TMRn.

Table 21.6 TMR Interrupt Sources

Name	Interrupt Sources	Interrupt Status Flag*	DTC Activation	Priority
CMIA0	TMR0.TCORA compare match	IR174.IR	Possible	High
CMIB0	TMR0.TCORB compare match	IR175.IR	Possible	↑ Low
OVI0	TMR0.TCNT overflow	IR176.IR	Not possible	
CMIA1	TMR1.TCORA compare match	IR177.IR	Possible	
CMIB1	TMR1.TCORB compare match	IR178.IR	Possible	
OVI1	TMR1.TCNT overflow	IR179.IR	Not possible	
CMIA2	TMR2.TCORA compare match	IR180.IR	Possible	
CMIB2	TMR2.TCORB compare match	IR181.IR	Possible	
OVI2	TMR2.TCNT overflow	IR182.IR	Not possible	
CMIA3	TMR3.TCORA compare match	IR183.IR	Possible	
CMIB3	TMR3.TCORB compare match	IR184.IR	Possible	
OVI3	TMR3.TCNT overflow	IR185.IR	Not possible	

Note : * For details, see section 11, Interrupt Control Unit (ICUa).

21.6.2 A/D Converter Activation

The A/D converter can be activated by a compare match A for TMR0 or TMR2.

If the TMRn.TCSR.ADTE bit (n = 0, 2) is set to 1 (A/D converter start requests by compare match A are enabled), a request to start A/D conversion is sent to the A/D converter by the occurrence of a compare match A. If the 8-bit timer conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started. For the corresponding A/D converter units, see section 34, 12-Bit A/D Converter (S12AD), and section 35, 10-Bit A/D Converter (ADa).

21.7 Usage Notes

21.7.1 Module Stop State Setting

Operation of the TMR can be disabled or enabled by using the module-stop control registers. The initial setting is for halting of TMR operation. Register access becomes possible after release from the module-stop state. For details, see section 9, Low Power Consumption.

21.7.2 Notes on Setting Cycle

If the compare match is selected for counter clear, TCNT is cleared at the last state 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).

$$f = PCLK / (N + 1)$$

21.7.3 Conflict between TCNT Write and Counter Clear

If a counter clear signal is generated concurrently with CPU write to TCNT, the clear takes priority and the write is not performed as shown in Figure 21.13.

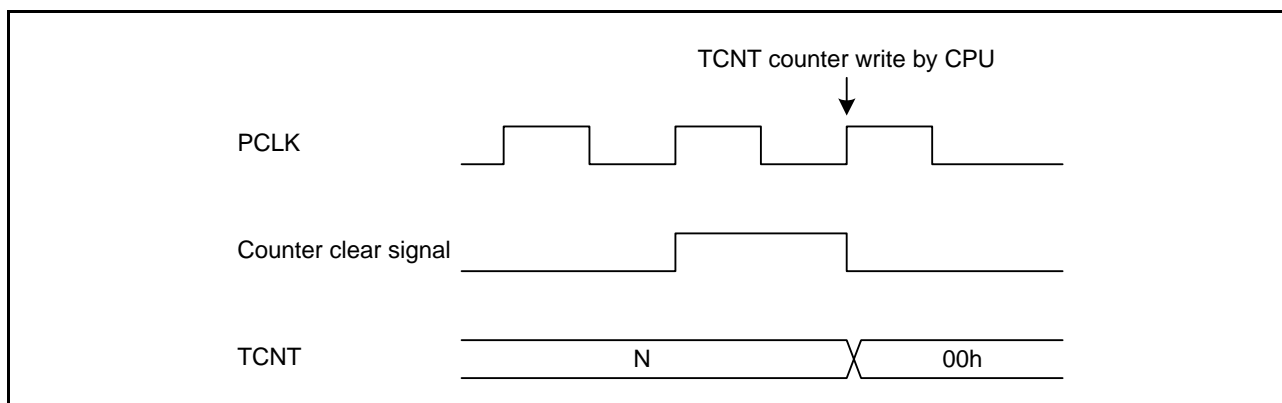


Figure 21.13 Conflict between TCNT Write and Counter Clear

21.7.4 Conflict between TCNT Write and Increment

Even if a counting-up signal is generated concurrently with CPU write to TCNT, the counting-up is not performed and the write takes priority as shown in Figure 21.14.

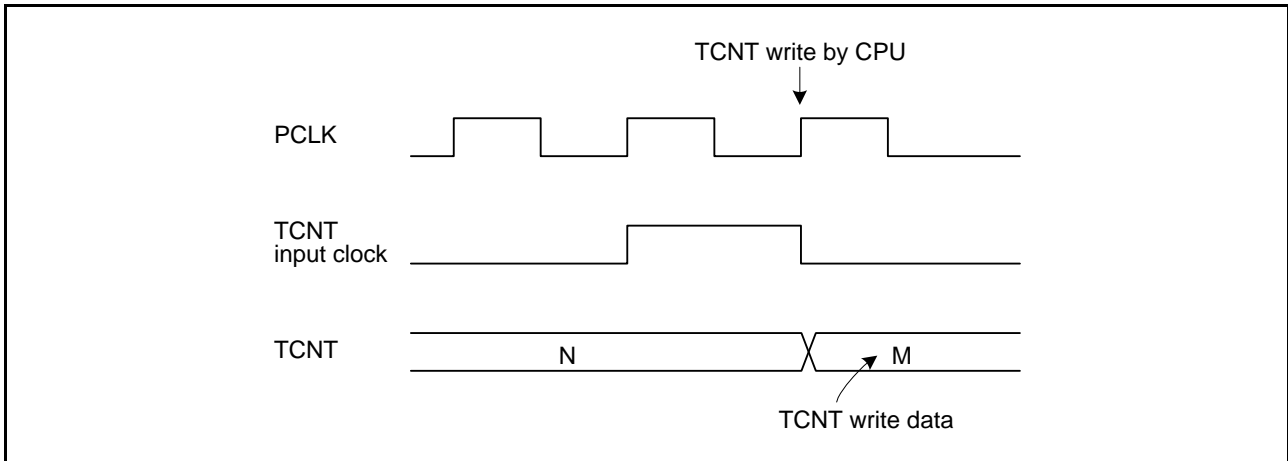


Figure 21.14 Conflict between TCNT Write and Increment

21.7.5 Conflict between TCORA or TCORB Write and Compare Match

Even if a compare match signal is generated simultaneously with CPU write to TCORA or TCORB, the write takes priority and the compare match signal is not set as shown in Figure 21.15.

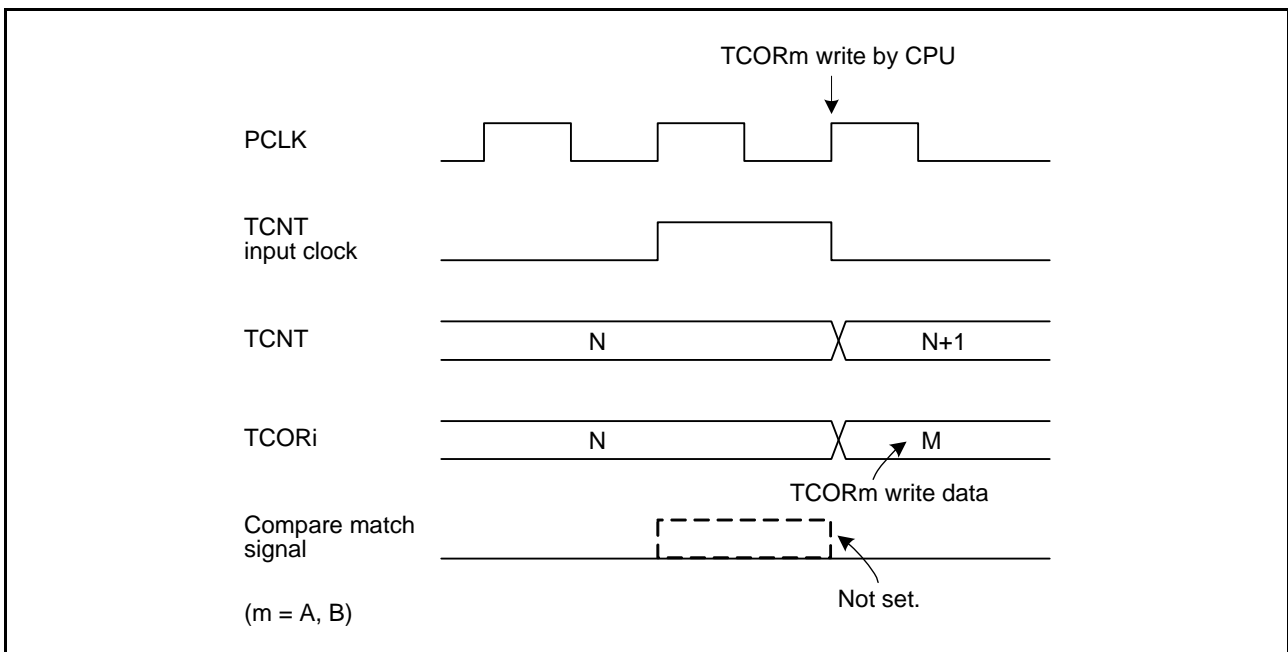


Figure 21.15 Conflict between TCORA or TCORB Write and Compare Match

21.7.6 Conflict between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output statuses set for compare match A and compare match B, as shown in Table 21.7.

Table 21.7 Timer Output Priorities

Output Setting	Priority
Toggle output	High
High output	
Low output	
No change	Low

21.7.7 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switched. Table 21.8 shows the relationship between the timing at which the internal clock is switched (by writing to the TCCR.CKS[2:0] bits) and the operation of TCNT.

When TCNT clock is generated from an internal clock, the rising edge of the internal 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 21.8, the change is considered as an edge. Therefore, a TCNT clock pulse is generated and TCNT is incremented.

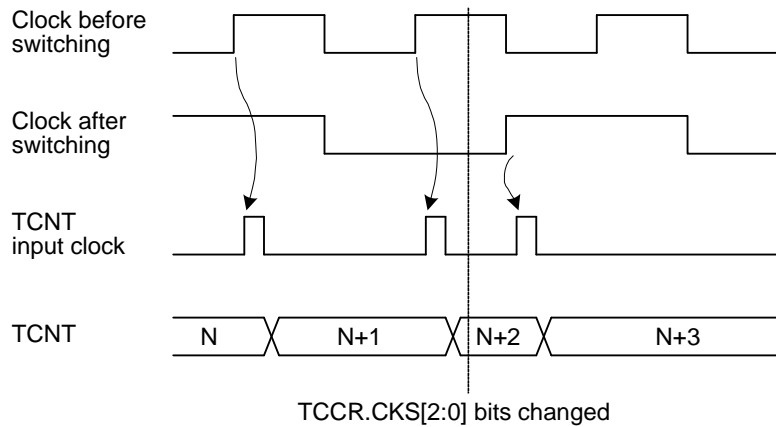
The erroneous increment of TCNT can also happen when switching between internal and external clocks.

Table 21.8 Switching of Internal Clocks and TCNT Operation

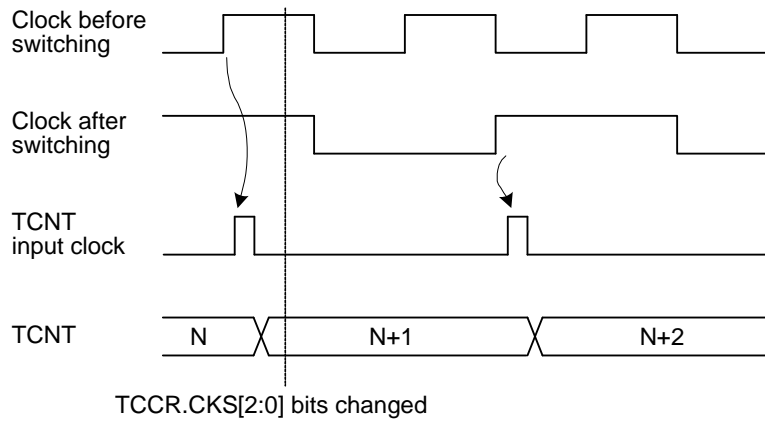
No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Clock Operation
1	Switching from low to low*1	
2	Switching from low to high*2	

Table 21.8 Switching of Internal Clocks and TCNT Operation

3 Switching from high to low*4



4 Switching from high to high



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.

21.7.8 Clock Source Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, input clocks for TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) are not generated, and the counter stops. Do not specify 16-bit counter mode and compare match count mode simultaneously.

22. Compare Match Timer (CMT)

The RX62N/RX621 Group has two on-chip compare match timer (CMT) units (unit 0 and unit 1) each consisting of a two-channel 16-bit timer (i.e., a total of four channels). The CMT has a 16-bit counter, and can generate interrupts at set intervals.

22.1 Overview

Table 22.1 lists the specifications for the CMT.

Figure 22.1 shows a block diagram of the CMT (unit 0). A two-channel CMT constitutes a unit. Unit 0 and unit 1 are the same in terms of specifications.

Table 22.1 Specifications of CMT

Item	Description
Count clock	<ul style="list-style-type: none"> Four internal clocks One clock from PCLK/8, PCLK/32, PCLK/128, and PCLK/512 can be selected individually for each channel.
Interrupt	A compare match interrupt can be requested individually for each channel.
Low power consumption facilities	Each unit can be placed in a module stop state.

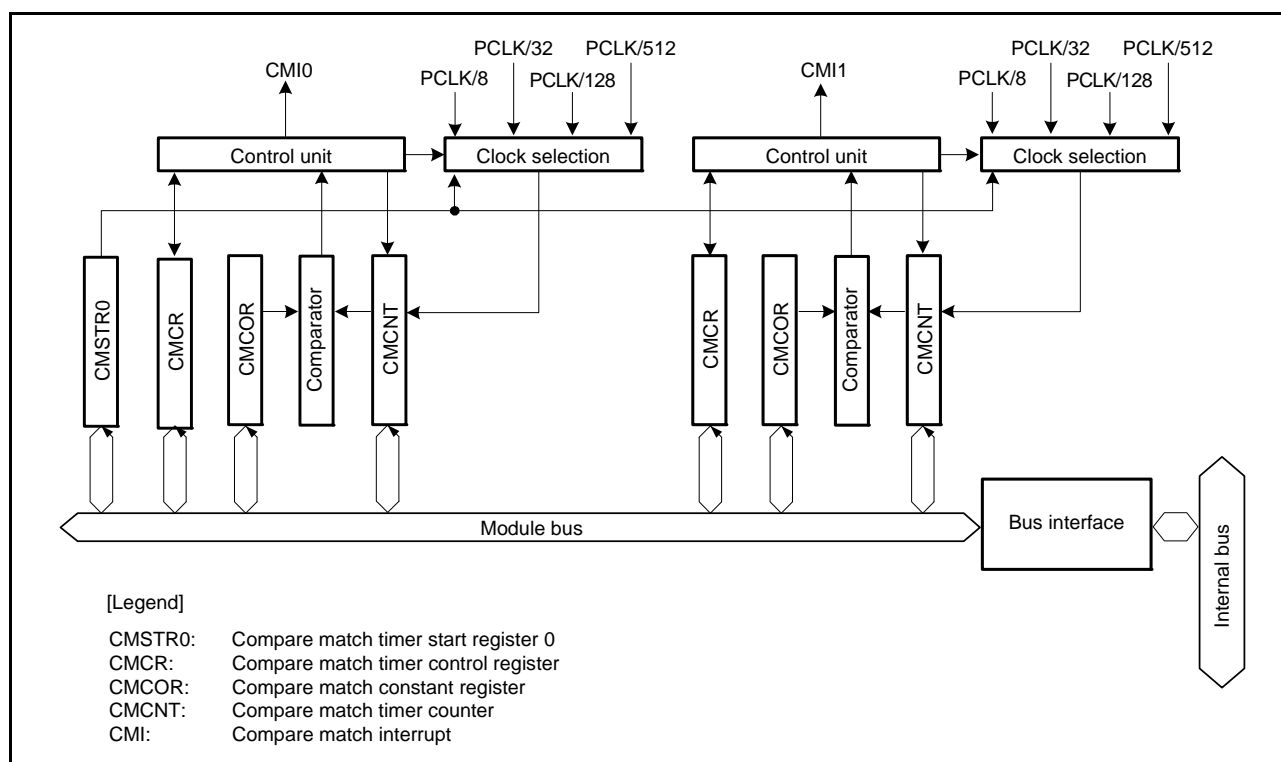


Figure 22.1 Block Diagram of CMT (Unit 0)

22.2 Register Descriptions

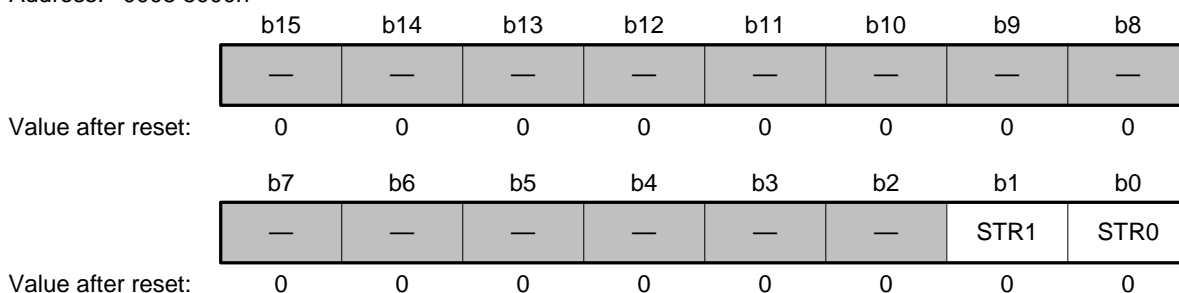
Table 22.2 lists the registers of the CMT.

Table 22.2 List of CMT Registers

Unit	Channel	Register Name	Symbol	Value after Reset	Address	Access Size		
Unit 0	CMT	Compare match timer start register 0	CMSTR0	0000h	0008 8000h	16		
		Compare match timer control register	CMCR	00x0h	0008 8002h	16		
		Compare match timer counter	CMCNT	0000h	0008 8004h	16		
	CMT0	Compare match timer constant register	CMCOR	FFFFh	0008 8006h	16		
		CMT1	Compare match timer control register	CMCR	00x0h	0008 8008h	16	
			Compare match timer counter	CMCNT	0000h	0008 800Ah	16	
			Compare match timer constant register	CMCOR	FFFFh	0008 800Ch	16	
		Unit 1	CMT	Compare match timer start register 1	CMSTR1	0000h	0008 8010h	16
				CMT2	Compare match timer control register	CMCR	00x0h	0008 8012h
Compare match timer counter	CMCNT				0000h	0008 8014h	16	
Compare match timer constant register	CMCOR		FFFFh		0008 8016h	16		
CMT3	Compare match timer control register		CMCR	00x0h	0008 8018h	16		
	Compare match timer counter		CMCNT	0000h	0008 801Ah	16		
	Compare match timer constant register		CMCOR	FFFFh	0008 801Ch	16		

22.2.1 Compare Match Timer Start Register 0 (CMSTR0)

Address: 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 always read as 0. The write value should always be 0.	R/W

CMSTR0 selects whether the CMT0.CMCNT or CMT1.CMCNT counter operates or is stopped.

STR0 Bit (Count Start 0)

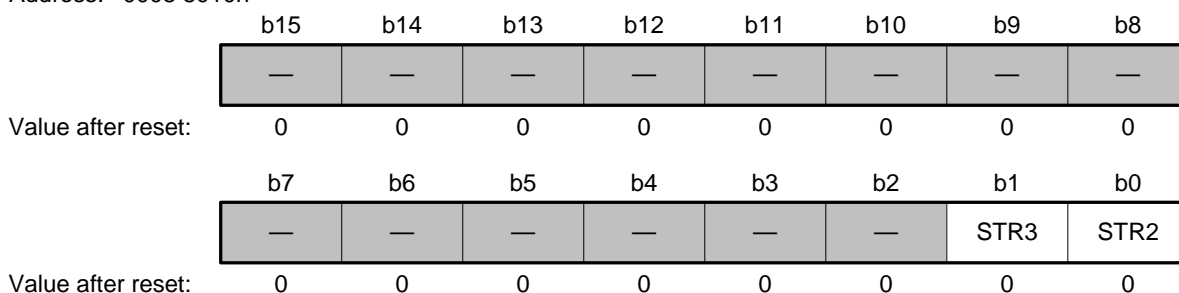
The STR0 bit specifies whether CMT0.CMCNT operates or is stopped.

STR1 Bit (Count Start 1)

The STR1 bit specifies whether CMT1.CMCNT operates or is stopped.

22.2.2 Compare Match Timer Start Register 1 (CMSTR1)

Address: 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 always read as 0. The write value should always be 0.	R/W

CMSTR1 selects whether the CMT2.CMCNT or CMT3.CMCNT counter operates or is stopped.

STR2 Bit (Count Start 2)

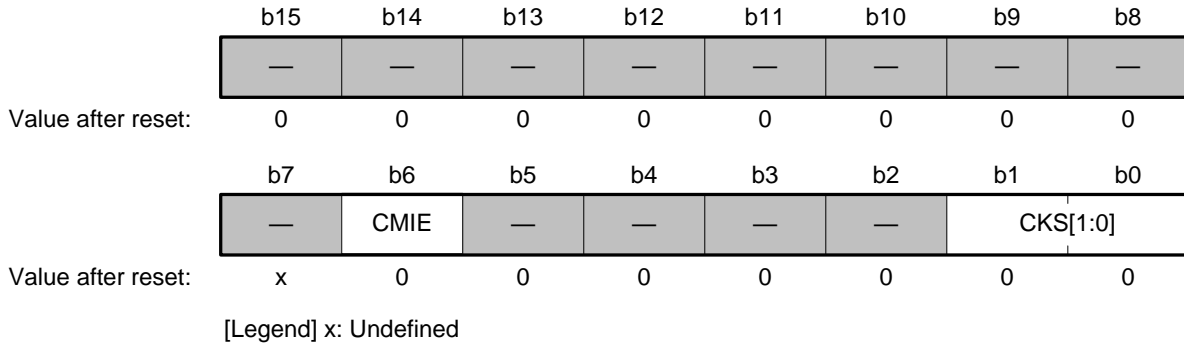
The STR2 bit specifies whether CMT2.CMCNT operates or is stopped.

STR3 Bit (Count Start 3)

The STR3 bit specifies whether CMT3.CMCNT operates or is stopped.

22.2.3 Compare Match Timer Control Register (CMCR)

Addresses: 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 always read as 0. The write value should always 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 always read as undefined. The write value should always be 1.	R/W
b15 to b8	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

CMCR sets the clock used for counting up.

If data write to the CMCR register conflicts with the generation of a compare-match, data write to the CMCR register is ignored. For details, see section 22.5.4, Notes on Data Write to the Compare-Match Timer Control Register (CMCR).

CKS[1:0] Bits (Clock Select)

These bits select the count clock to be input to CMCNT from four internal clocks obtained by dividing the peripheral clock (PCLK).

When the STR_n (n = 0 to 3) bit in CMSTR_m (m = 0 or 1) is set to 1, CMCNT starts counting up on the count clock selected with bits CKS[1:0].

CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0 to 3) generation when CMCNT and CMCOR values match.

22.2.4 Compare Match Timer Counter (CMCNT)

Addresses: CMT0.CMCNT 0008 8004h, CMT1.CMCNT 0008 800Ah,
CMT2.CMCNT 0008 8014h, CMT3.CMCNT 0008 801Ah



CMCNT is a readable/writable up-counter to generate interrupt requests.

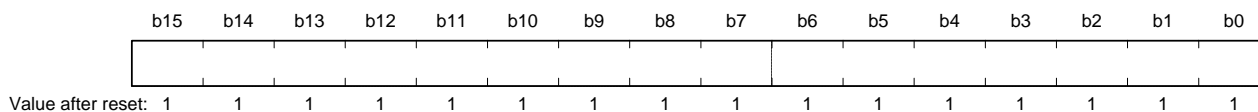
When a count clock is selected by bits CKS[1:0] in CMCR and the STR_n (n = 0 to 3) bit in CMSTR_m (m = 0 or 1) is set to 1, CMCNT starts counting up using the selected count clock.

When the value in CMCNT and the value in CMCOR match, CMCNT is cleared to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated.

Do not set the CMCNT counter and the CMCOR register to the same value while the CMCNT counter counting operation is halted. For details, see section 22.5.5, Notes on the Compare-Match Timer Counter (CMCNT) and the Compare-Match Constant Register (CMCOR).

22.2.5 Compare Match Timer Constant Register (CMCOR)

Addresses: CMT0.CMCOR 0008 8006h, CMT1.CMCOR 0008 800Ch,
CMT2.CMCOR 0008 8016h, CMT3.CMCOR 0008 801Ch



CMCOR sets the interval up to a compare match with CMCNT.

Do not set the CMCNT counter and the CMCOR register to the same value while the CMCNT counter counting operation is halted. For details, see section 22.5.5, Notes on the Compare-Match Timer Counter (CMCNT) and the Compare-Match Constant Register (CMCOR).

22.3 Operation

22.3.1 Periodic Count Operation

When a count clock is selected by bits CKS[1:0] in CMCR and the STRn (n = 0 to 3) bit in CMSTRm (m = 0 or 1) is set to 1, CMCNT starts counting up using the selected count clock.

When the value in CMCNT and the value in CMCOR match, CMCNT is cleared to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated. CMCNT then starts counting up again from 0000h. Figure 22.2 shows the operation of the CMCNT counter.

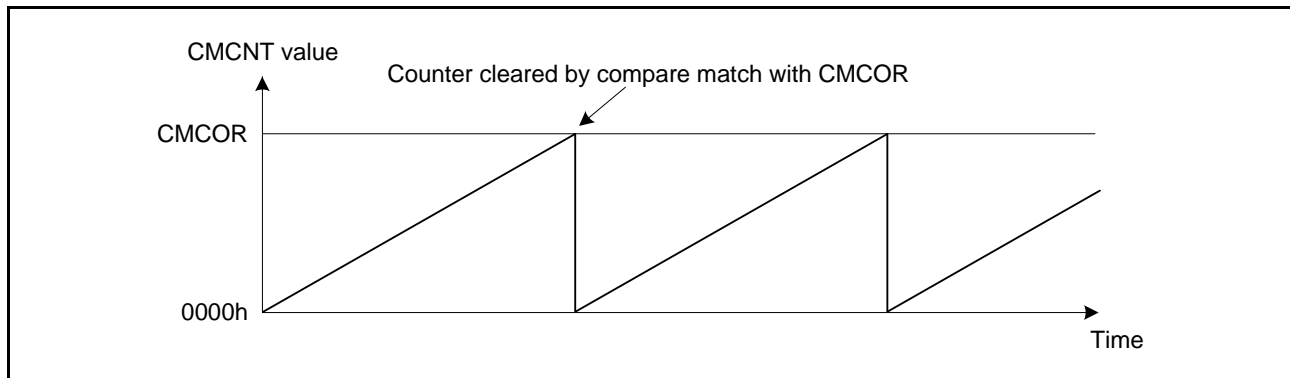


Figure 22.2 CMCNT Counter Operation

22.3.2 CMCNT Count Timing

One of four internal clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral clock (PCLK) can be selected as a count clock with the CKS[1:0] bits in CMCSR. Figure 22.3 shows the timing of CMCNT.

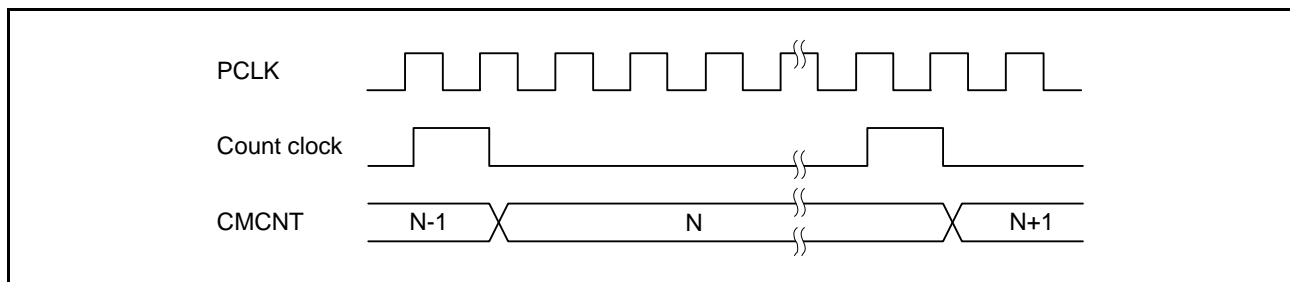


Figure 22.3 CMCNT Count Timing

22.4 Interrupts

22.4.1 Interrupt Sources

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt (CMI_n) (n = 0 to 3). When a compare match interrupt occurs, the corresponding interrupt request is output.

When the interrupt is used to activate a CPU interrupt, the priority of channels can be changed by the interrupt control unit settings. For details, see section 11, Interrupt Control Unit (ICUa).

Table 22.3 CMT Interrupt Sources

Name	Interrupt Sources	Interrupt Status Flag	DTC Activation	DMACA Activation
CMI0	Compare match between CMT0.CMCNT and CMT0.CMCOR	IR028.IR	Possible	Possible
CMI1	Compare match between CMT1.CMCNT and CMT1.CMCOR	IR029.IR	Possible	Possible
CMI2	Compare match between CMT2.CMCNT and CMT2.CMCOR	IR030.IR	Possible	Possible
CMI3	Compare match between CMT3.CMCNT and CMT3.CMCOR	IR031.IR	Possible	Possible

22.4.2 Timing of Compare Match Interrupt Generation

When CMCNT and CMCOR match, a compare match interrupt (CMI_n) (n = 0 to 3) is generated.

A compare match signal is generated the timing when the CMCNT counter updates the matched count value. That is, after a match between CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT counter clock input.

Figure 22.4 illustrates the timing of the generation of the compare match signal.

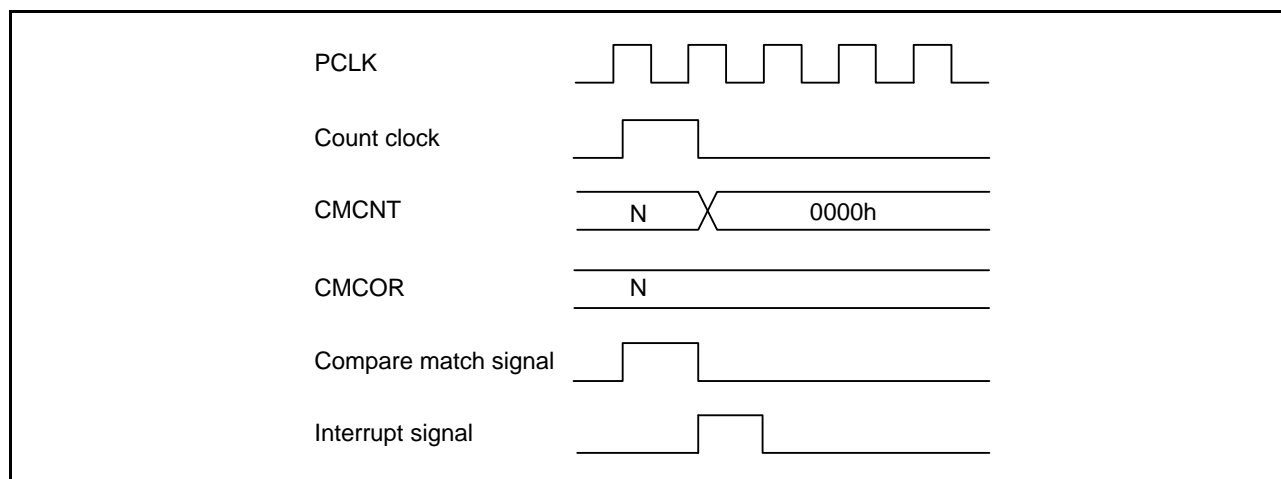


Figure 22.4 Timing of Generation of Compare Match Signal

22.5 Usage Notes

22.5.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the module stop control register. The CMT is disabled by default. The registers can be accessed by canceling the module stop state. For details, see section 9, Low Power Consumption.

22.5.2 Conflict between Write and Compare-Match Processes of Compare Match Timer Counter (CMCNT)

When the compare match signal is generated while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 22.5 shows the timing to clear the CMCNT counter.

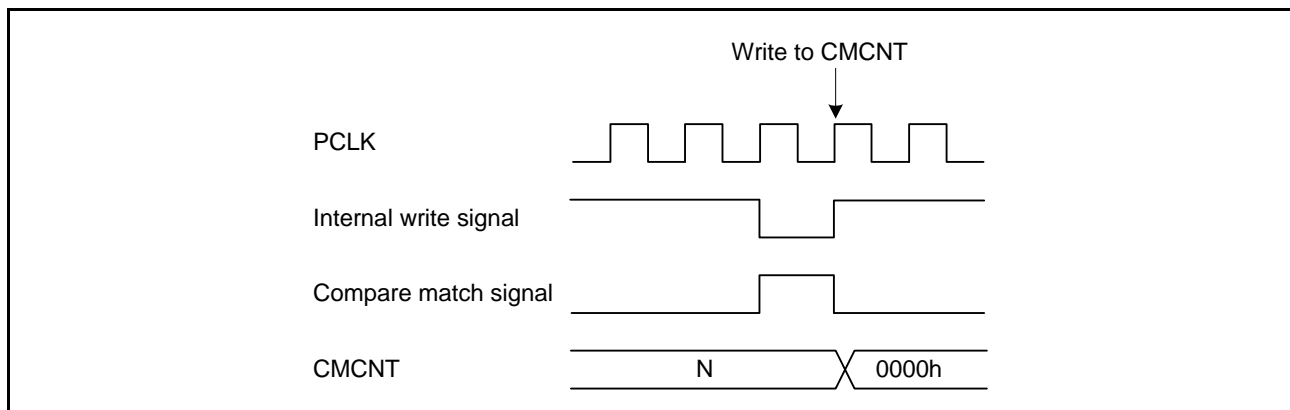


Figure 22.5 Conflict between Write and Compare Match Processes of CMCNT

22.5.3 Conflict between Write and Count-Up Processes of Compare Match Timer Counter (CMCNT)

Even when the count-up occurs while writing to CMCNT, the writing has priority over the count-up. In this case, the count-up is not performed. Figure 22.6 shows the timing to write the CMCNT counter.

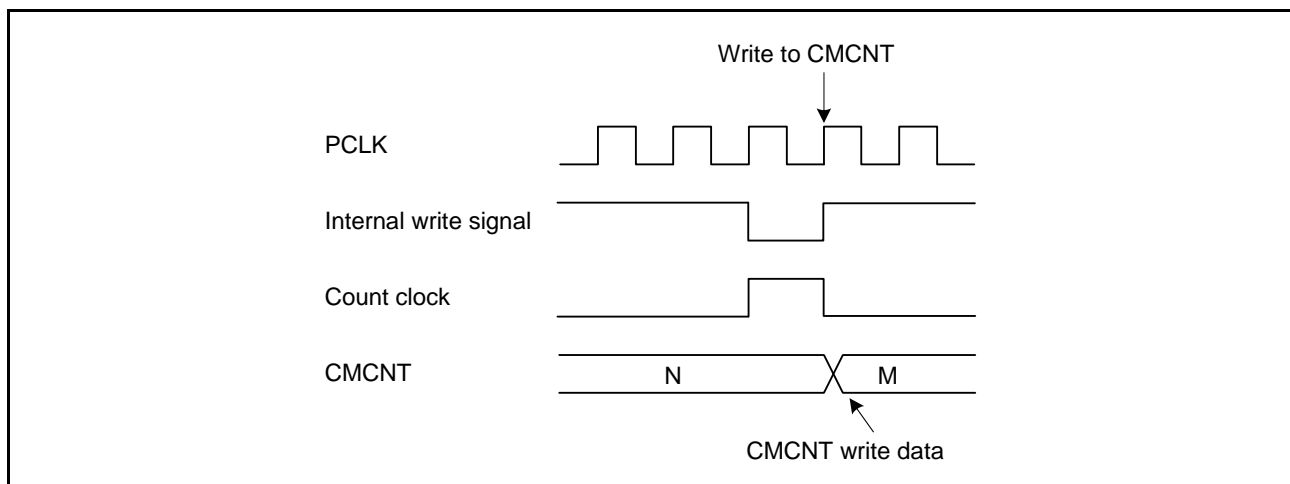


Figure 22.6 Conflict between Write and Count-Up Processes of CMCNT

22.5.4 Notes on Data Write to the Compare-Match Timer Control Register (CMCR)

If data write to the CMCR register conflicts with the generation of a compare-match, data write to the CMCR register is ignored. Therefore, write data to the CMCR register, then read the data from the CMCR register to confirm that the data is written correctly. If the data is not written correctly, write data to the CMCR register again.

Since undefined data is read from bit 7 in the CMCR register, comparison with the written data requires special care.

22.5.5 Notes on the Compare-Match Timer Counter (CMCNT) and the Compare-Match Constant Register (CMCOR)

Do not set the CMCNT counter and the CMCOR register to the same value while the CMCNT counter is halted.

Otherwise, a compare-match occurs even while the counter is halted. At this time, if the compare-match interrupt enable bit (CMCR.CMIE) is set to 1 (enabled), a compare-match interrupt is generated.

Note that the CMCNT counter is automatically cleared to 0000h when a compare-match with the CMCOR register value occurs regardless of whether a compare-match interrupt is disabled or enabled.

23. Realtime Clock (RTC)

23.1 Overview

The RX62N/RX621 Group has a realtime clock (RTC) and 32.768-kHz crystal resonator.

Table 23.1 shows the specifications of the RTC.

Table 23.1 Specifications of RTC

Item	Description
Count source	32.768-kHz clock dedicated for the RTC
Clock and calendar functions	<ul style="list-style-type: none"> • Year, month, the date, day of the week, hours, minutes, and seconds are counted and represented in BCD • Indicates the state of 1Hz, 2Hz, 4Hz, 8Hz, 16Hz, 32Hz, and 64Hz in binary • Start/stop function • 30-second adjustment (30 seconds or less are rounded down to 00 second, and 30 seconds or more are rounded up to one minute) • Automatic leap year adjustment • Output a 1-Hz clock
Interrupts	<ul style="list-style-type: none"> • Alarm interrupt (ALM) Year, month, the date, day of the week, hours, minutes, and seconds can be selected as conditions for the alarm interrupt • Periodic interrupt (PRD) 2 seconds, 1 second, 1/2 second, 1/4 second, 1/16 second, 1/64 second or 1/256 second can be selected as an interrupt period. • Carry interrupt (CUP) Indicates occurrence of a carry to the seconds counter or a carry to the 64-Hz counter from the prescaler during reading of the 64-Hz counter • Recovery from software standby mode or deep software standby mode can be performed by an alarm interrupt

The RTC adopts the frequency of the count source (32.768 kHz) as its reference clock. The respective counter adopts the 128-Hz clock as its reference clock; that is acquired by dividing the count source with the prescaler. The time period for a second, that is the base for the time counter function, is generated by the 64-Hz counter.

Figure 23.1 shows a block diagram of the RTC and Table 23.2 shows the pin configuration of the RTC.

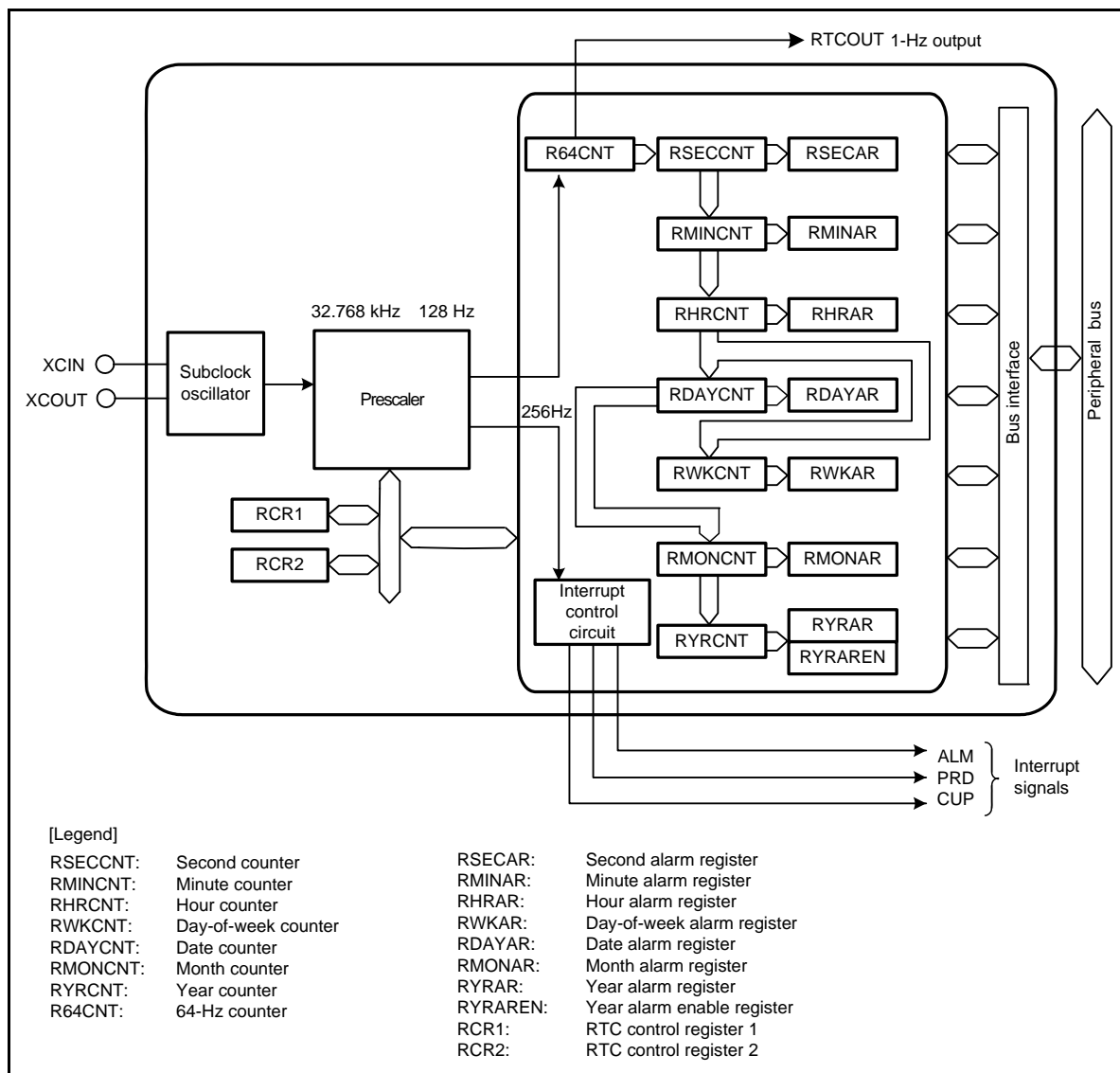


Figure 23.1 Block Diagram of RTC

Table 23.2 Pin Configuration

Pin Name	I/O	Function
XCOUT	Output	Connect a 32.768-kHz crystal resonator for the RTC to these pins. The external clock can also be input to the XCIN pin.
XCIN	Input	
RTCOUT	Output	Output a 1-Hz clock.

23.2 Register Descriptions

Table 23.3 lists the registers of the RTC.

Table 23.3 Registers of RTC

Register Name	Symbol	Value after Reset	Address	Access Size
64-Hz counter	R64CNT	xxh	0008 C400h	8
Second counter	RSECCNT	xxh	0008 C402h	8
Minute counter	RMINCNT	xxh	0008 C404h	8
Hour counter	RHRCNT	xxh	0008 C406h	8
Day-of-week counter	RWKCNT	0xh	0008 C408h	8
Date counter	RDAYCNT	xxh	0008 C40Ah	8
Month counter	RMONCNT	xxh	0008 C40Ch	8
Year counter	RYRCNT	xxxxh	0008 C40Eh	16
Second alarm register	RSECAR	xxh	0008 C410h	8
Minute alarm register	RMINAR	xxh	0008 C412h	8
Hour alarm register	RHRAR	xxh	0008 C414h	8
Day-of-week alarm register	RWKAR	xxh	0008 C416h	8
Date alarm register	RDAYAR	xxh	0008 C418h	8
Month alarm register	RMONAR	xxh	0008 C41Ah	8
Year alarm register	RYRAR	xxxxh	0008 C41Ch	16
Year alarm enable register	RYRAREN	x0h	0008 C41Eh	8
RTC control register 1	RCR1	00h	0008 C422h	8
RTC control register 2	RCR2	01h	0008 C424h	8

[Legend] x: Undefined

Note 1. Executing a reset or entering deep software standby mode does not initialize the values of bits in RTC registers for which the state after a reset is indicated by an "X" (undefined). Furthermore, if counting operations are in progress (the RCR2.START bit is 1), operation of the year, month, day, date, hours, minutes, seconds, and 64-Hz counters continues after a transition to the reset or low-power-consumption state.

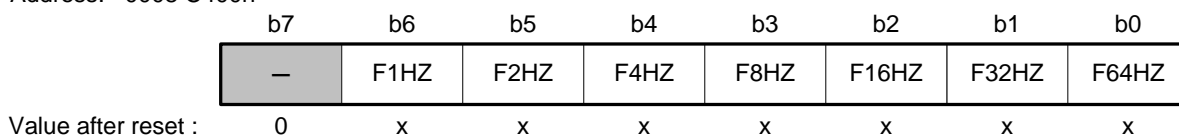
However, care is required on this point because there is a chance of a register's value being lost if processing to overwrite or otherwise change its value was in progress at the time. Furthermore, do not cause a transition to software standby mode or deep software standby mode immediately after setting an RTC register. For details, refer to section 23.5.3, Transition to Low Power Consumption Modes after Setting Registers

For products of the RX62N and RX621 Groups, applying a reset through the #RES pin in deep software standby mode will destroy the values of the registers. For this reason, re-initialize the values of the registers after a pin reset.

Writing and reading of RTC registers must be in accord with section 23.5.4, Points for Caution When Writing to and Reading from Registers.

23.2.1 64-Hz Counter (R64CNT)

Address: 0008 C400h



[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	F64HZ	64 Hz	Indicate the state between 64 Hz and 1 Hz.	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 always read as 0 and cannot be modified.

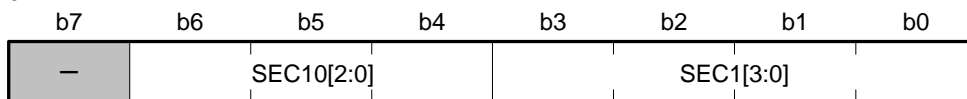
The 64-Hz counter (R64CNT) generates the period for a second by counting up periods of the 128-Hz clock. The state in the sub-seconds range can be confirmed by reading this counter.

This counter is cleared to 0 when the RESET or ADJ bit in the RTC control register 2 (RCR2) is set to 1.

To read this counter, follow the procedure in Figure 23.4.

23.2.2 Second Counter (RSECCNT)

Address: 0008 C402h



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

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	Ones Place of Seconds	Counts from 0 to 9 once per second. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	SEC10[2:0]	Tens Place of Seconds	Counts from 0 to 5 for 60-second counting.	R/W
b7	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

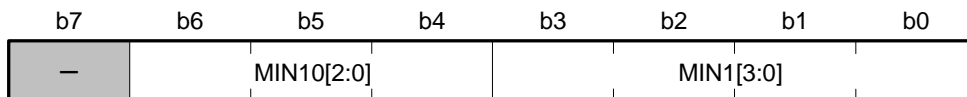
RSECCNT is used for setting and counting the BCD-coded second value. It counts carries generated once per second in the 64-Hz counter.

The setting range is decimal 00 to 59. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2.

To read this counter, follow the procedure in Figure 23.4.

23.2.3 Minute Counter (RMINCNT)

Address: 0008 C404h



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

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	Tens Place of Minutes	Counts from 0 to 9 once per minute. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	MIN10[2:0]	Ones Place of Minutes	Counts from 0 to 5 for 60-minute counting.	R/W
b7	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

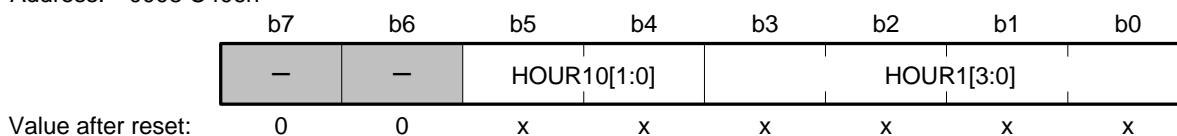
RMINCNT is used for setting and counting the BCD-coded minute value. It counts carries generated once per minute in the second counter.

A value from 00 through 59 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 23.3.4, Reading 64-Hz Counter and Time.

23.2.4 Hour Counter (RHRCNT)

Address: 0008 C406h



Value after reset:

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HOUR1[3:0]	Ones Place of Hours	Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	HOUR10[1:0]	Tens Place of Hours	Counts from 0 to 2 once per carry from the ones place.	R/W
b7, b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

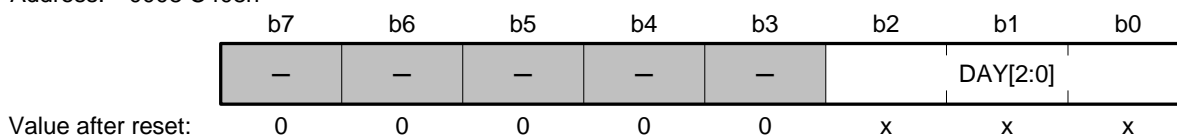
RHRCNT is used for setting and counting the BCD-coded hour value. It counts carries generated once per hour in the minute counter.

A value from 00 through 23 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 23.3.4, Reading 64-Hz Counter and Time.

23.2.5 Day-of-Week Counter (RWKCNT)

Address: 0008 C408h



Value after reset:

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAY[2:0]	Day-of-Week Counting	000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: (Setting Prohibited)	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

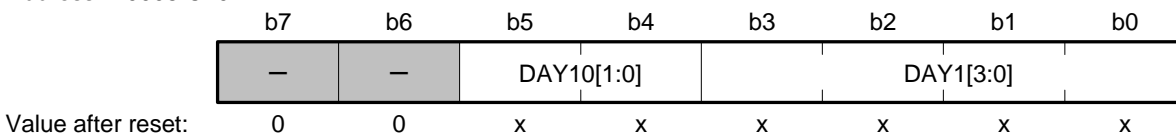
RWKCNT is used for setting and counting in the BCD-coded day-of-week value. It counts carries generated once per day in the hour counter.

A value from 0 through 6 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 23.3.4, Reading 64-Hz Counter and Time.

23.2.6 Date Counter (RDAYCNT)

Address: 0008 C40Ah



Value after reset:

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DAY1[3:0]	Ones Place of Days	Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	DAY10[1:0]	Tens Place of Days	Counts from 0 to 3 once per carry from the ones place.	R/W
b7, b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RDAYCNT is used for setting and counting the BCD-coded date value. It counts carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year.

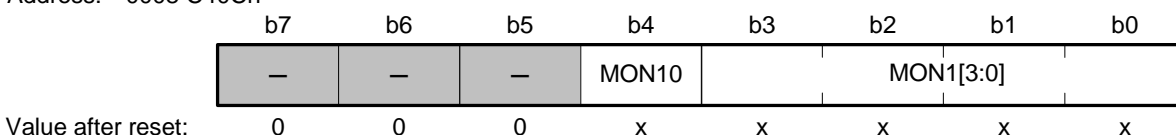
Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.

A value from 01 through 31 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. (When specifying a value, note that the range of specifiable days depends on the month and whether the year is a leap year.) Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 23.3.4, Reading 64-Hz Counter and Time.

23.2.7 Month Counter (RMONCNT)

Address: 0008 C40Ch



Value after reset:

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	Ones Place of Months	Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
b4	MON10	Tens Place of Months	Counts from 0 to 1 once per carry from the ones place.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

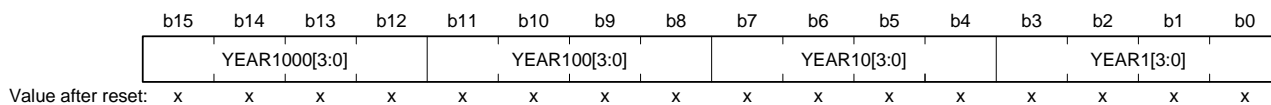
RMONCNT is used for setting and counting the BCD-coded month value. It counts carries generated once per month in the date counter.

A value from 01 through 12 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 23.3.4, Reading 64-Hz Counter and Time.

23.2.8 Year Counter (RYRCNT)

Address: 0008 C40Eh



[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YEAR1[3:0]	Ones Place of Years	Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
b7 to b4	YEAR10[3:0]	Tens Place of Years	Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place.	R/W
b11 to b8	YEAR100[3:0]	Hundreds Place of Years	Counts from 0 to 9 once per carry from tens place. When a carry is generated in the hundreds place, 1 is added to the thousands place.	R/W
b15 to b12	YEAR1000[3:0]	Thousands Place of Years	Counts from 0 to 9 once per carry from hundreds place.	R/W

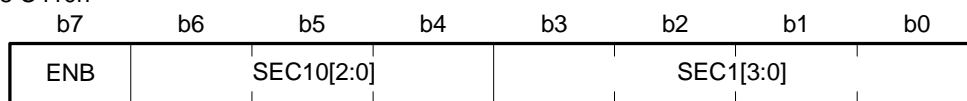
RYRCNT is used for setting and counting the BCD-coded year value. It counts carries generated once per year in the month counter.

A value from 0000 through 9999 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 23.3.4, Reading 64-Hz Counter and Time.

23.2.9 Second Alarm Register (RSECAR)

Address: 0008 C410h



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

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1 Second	Value for the ones place of seconds	R/W
b6 to b4	SEC10[2:0]	10 Seconds	Value for the tens place of seconds	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RSECCNT value.	R/W

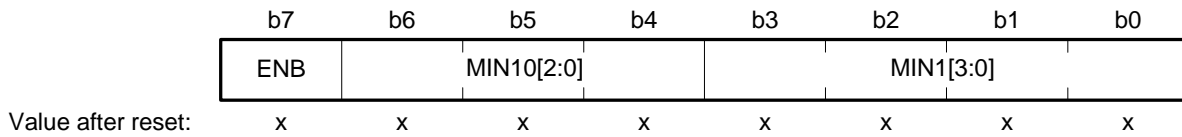
RSECAR is an alarm register corresponding to the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RSECAR values from 00 through 59 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RSECAR is cleared to 00h when the RESET bit in the RTC control register 2 (RCR2) is set to 1.

23.2.10 Minute Alarm Register (RMINAR)

Address: 0008 C412h



Value after reset:

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1 Minute	Value for the ones place of minutes	R/W
b6 to b4	MIN10[2:0]	10 Minutes	Value for the tens place of minutes	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RMINCNT value.	R/W

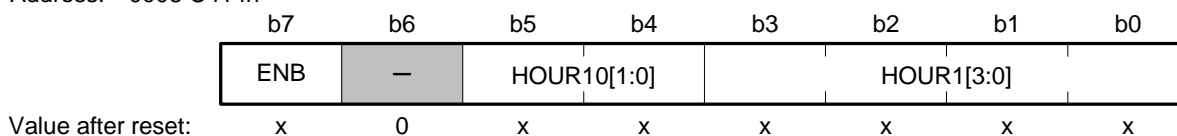
RMINAR is an alarm register corresponding to the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RMINAR values from 00 through 59 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RMINAR is cleared to 00h when the RESET bit in the RTC control register 2 (RCR2) is set to 1.

23.2.11 Hour Alarm Register (RHRAR)

Address: 0008 C414h



Value after reset:

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HOUR1[3:0]	1 Hour	Value for the ones place of hours	R/W
b5, b4	HOUR10[2:0]	10 Hours	Value for the tens place of hours	R/W
b6	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RHRCNT value.	R/W

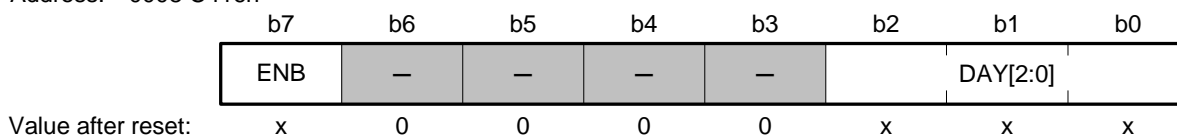
RHRAR is an alarm register corresponding to the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RHRAR values from 00 through 23 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RHRAR is cleared to 00h when the RESET bit in the RTC control register 2 (RCR2) is set to 1.

23.2.12 Day-of-Week Alarm Register (RWKAR)

Address: 0008 C416h



Value after reset:

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAY[2:0]	Day-of-Week Setting	000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: (Setting Prohibited)	R/W
b6 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RWKCNT value.	R/W

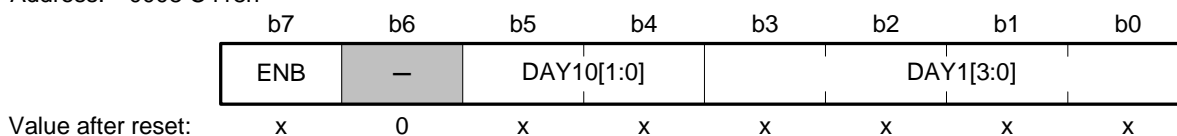
RWKAR is an alarm register corresponding to the BCD-coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RWKAR values from 0 through 6 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RWKAR is cleared to 00h when the RESET bit in the RTC control register 2 (RCR2) is set to 1.

23.2.13 Date Alarm Register (RDAYAR)

Address: 0008 C418h



Value after reset:

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DAY1[3:0]	1 Day	Value for the ones place of days	R/W
b5, b4	DAY10[1:0]	10 Days	Value for the tens place of days	R/W
b6	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RDAYCNT value.	R/W

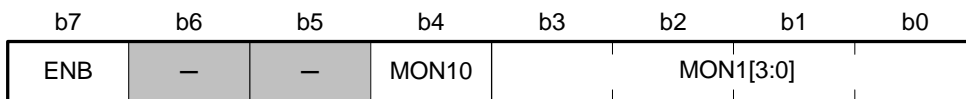
RDAYAR is an alarm register corresponding to the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RDAYAR values from 01 through 31 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RDAYAR is cleared to 00h when the RESET bit in the RTC control register 2 (RCR2) is set to 1.

23.2.14 Month Alarm Register (RMONAR)

Address: 0008 C41Ah



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

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	MON1[3:0]	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	These bits are always read as 0. The write value should always be 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RMONCNT value.	R/W

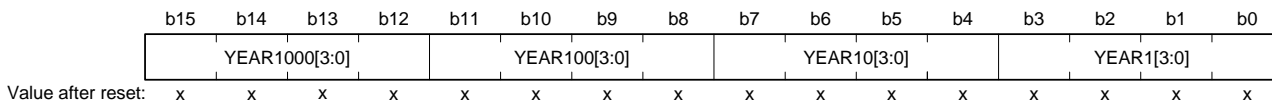
RMONAR is an alarm register corresponding to the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RMONAR values from 01 through 12 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RMONAR is cleared to 00h when the RESET bit in the RTC control register 2 (RCR2) is set to 1.

23.2.15 Year Alarm Register (RYRAR)

Address: 0008 C41Ch



[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YEAR1[3:0]	1 Year	Value for the ones place of years	R/W
b7 to b4	YEAR10[3:0]	10 Years	Value for the tens place of years	R/W
b11 to b8	YEAR100[3:0]	100 Years	Value for the hundreds place of years	R/W
b15 to b12	YEAR1000[3:0]	1000 Years	Value for the thousands place of years	R/W

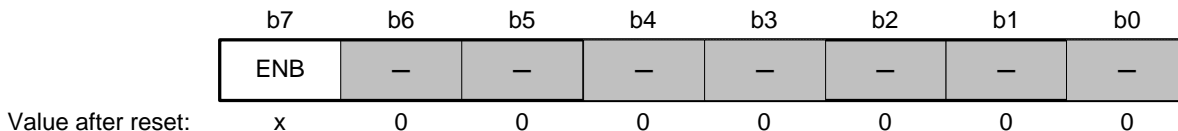
RYRAR is an alarm register corresponding to the BCD-coded year counter RYRCNT.

RYRAR values from 0000 through 9999 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RYRAR is cleared to 0000h when the RESET bit in the RTC control register 2 (RCR2) is set to 1.

23.2.16 Year Alarm Enable Register (RYRAREN)

Address: 0008 C41Eh



[Legend] x: Undefined

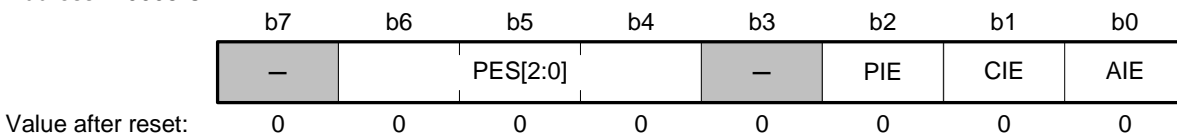
Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	ENB	ENB	When this bit is set to 1, the RYRAR value is compared with the RYRCNT value.	R/W

When the ENB bit in RYRAREN is set to 1, the RYRCAR value is compared with the RYRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RYRAREN is cleared to 00h when the RESET bit in the RTC control register 2 (RCR2) is set to 1.

23.2.17 RTC Control Register 1 (RCR1)

Address: 0008 C422h



Bit	Symbol	Bit Name	Description	R/W
b0	AIE	Alarm Interrupt Enable	0: An alarm interrupt is not requested 1: An alarm interrupt is requested	R/W
b1	CIE	Carry Interrupt Enable	0: A carry interrupt is not requested when a carry to the second counter occurred or a carry to the 64-Hz counter occurred during read access to the 64-Hz counter. 1: A carry interrupt is requested when a carry to the second counter occurred or a carry to the 64-Hz counter occurred during read access to the 64-Hz counter.	R/W
b2	PIE	Periodic Interrupt Enable	0: A periodic interrupt is not requested. 1: A periodic interrupt is requested.	R/W
b3	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b6 to b4	PES[2:0]	Periodic Interrupt Select	000: No periodic interrupts generated 001: A periodic interrupt generated every 1/256 second 010: A periodic interrupt generated every 1/64 second 011: A periodic interrupt generated every 1/16 second 100: A periodic interrupt generated every 1/4 second 101: A periodic interrupt generated every 1/2 second 110: A periodic interrupt generated every 1 second 111: A periodic interrupt generated every 2 seconds	R/W
b7	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

RCR1 controls interrupts.

AIE Bit (Alarm Interrupt Enable)

This bit enables or disables alarm interrupt requests.

With regard to the source of the alarm interrupt on return from deep software standby, an interrupt request is generated if the times indicated by the counters and alarm settings match, regardless of the value of the AIE bit.

The AIE bit is updated in synchronization with the 128-Hz clock. When the AIE bit is modified, the value before the modification is continuously read until the modification is completed. In cases where the AIE bit is modified, check that the bit has been updated without fail before continuing with further processing.

CIE Bit (Carry Interrupt Enable)

This bit enables or disabled interrupt requests when a carry to the second counter occurred or a carry to the 64-Hz counter occurred during read access to the 64-Hz counter.

PIE Bit (Periodic Interrupt Enable)

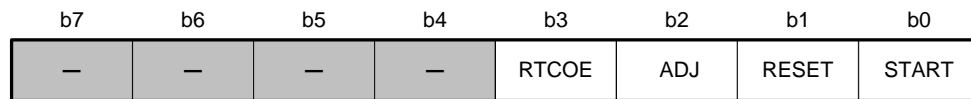
This bit enables or disabled periodic interrupt requests.

PES[2:0] Bits (Periodic Interrupt Select)

These bits specify the period for the periodic interrupt. A periodic interrupt (PRD) is generated with the period specified by these bits.

23.2.18 RTC Control Register 2 (RCR2)

Address: 0008 C424h



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

Bit	Symbol	Bit Name	Description	R/W
b0	START	Start	0: Year, month, day-of-week, date, hour, minute, second, and 64-Hz counters, and prescaler are stopped. 1: Year, month, day-of-week, date, hour, minute, second, and 64-Hz counters, and prescaler operate normally.	R/W
b1	RESET	Reset	0: Normal count operation or the initialization is completed. 1: Prescaler, 64-Hz counter, and alarm registers are initialized.	R/W
b2	ADJ	30-Second Adjustment	0: Normal count operation or completion of 30-second adjustment 1: During 30-second adjustment	R/W
b3	RTCOE	RTCOU Output Control	0: Clock signals are not output from the RTCOUT pin. 1: Clock signals are output from the RTCOUT pin.	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RCR2 is a register for 30-second adjustment, prescaler and R64CNT reset, and count control.

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, the value before the modification is continuously read until the modification is completed. In case when the START bit is modified, check that the bit is updated without fail, and then make next settings.

RESET Bit (Reset)

This bit initializes the prescaler, R64CNT, alarm registers, and RYRAREN.

During the normal count operation (the RESET bit = 1), the above registers are initialized in synchronization with the 128-Hz clock. After 1 is written, the RESET bit is automatically cleared to 0 when the reset of the above registers is completed.

While the normal count is stopped (the RESET bit = 0), the above registers are initialized immediately after 1 is written to the RESET bit. After the initialization is completed, the RESET bit is automatically cleared to 0.

In case when 1 is written to the RESET bit, check that the bit is cleared to 0, and then make next settings.

ADJ Bit (30-Second Adjustment)

This bit is for the 30-second adjustment (30 seconds or less are rounded down to 00 second, and 30 seconds or more are rounded up to one minute).

The 30-second adjustment is performed in synchronization with the 128-Hz clock. When 1 is written to this bit, the ADJ bit is automatically cleared to 0 after the 30-second adjustment is completed. In case when 1 is written to the ADJ bit, check that the bit is cleared to 0, and then make next settings.

RTCOE Bit (RTCOU Output Control)

This bit controls outputs of 1-Hz clock signals from the RTCOUT pin.

23.3 Operation

23.3.1 Initial Settings of Registers after Power-On

All the registers should be set after the power is turned on.

23.3.2 Setting Time

Figure 23.2 shows how to set the time.

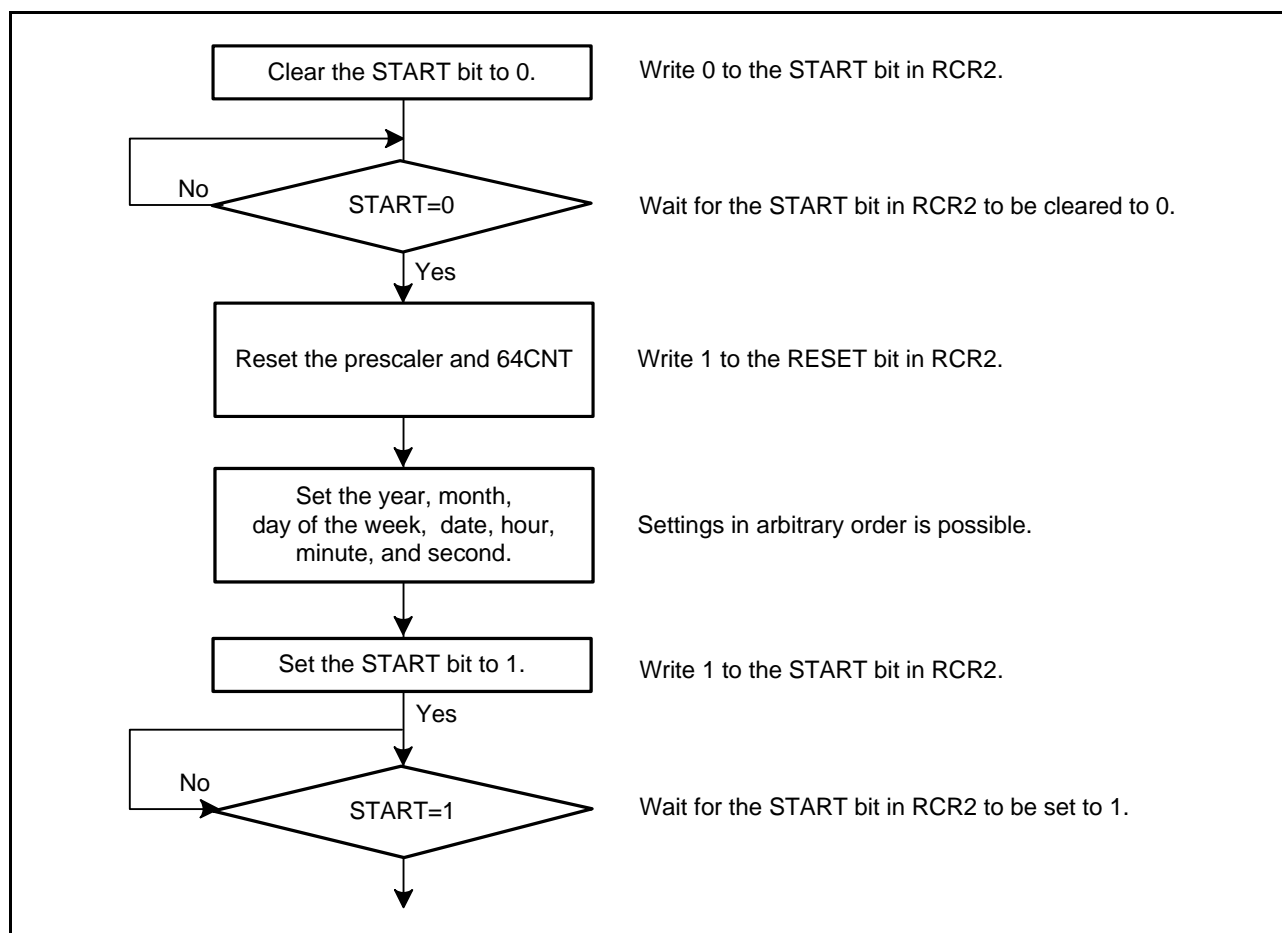


Figure 23.2 Setting Time

23.3.3 30-Second Adjustment

Figure 23.3 shows how to execute 30-second adjustment.

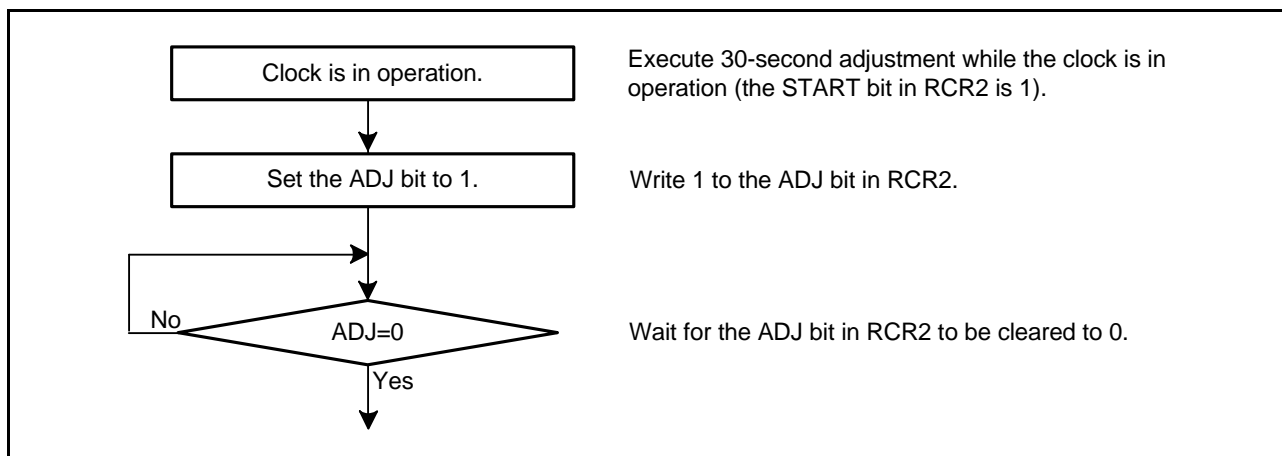


Figure 23.3 30-Second Adjustment

23.3.4 Reading 64-Hz Counter and Time

Figure 23.4 shows how to read the 64-Hz counter and time.

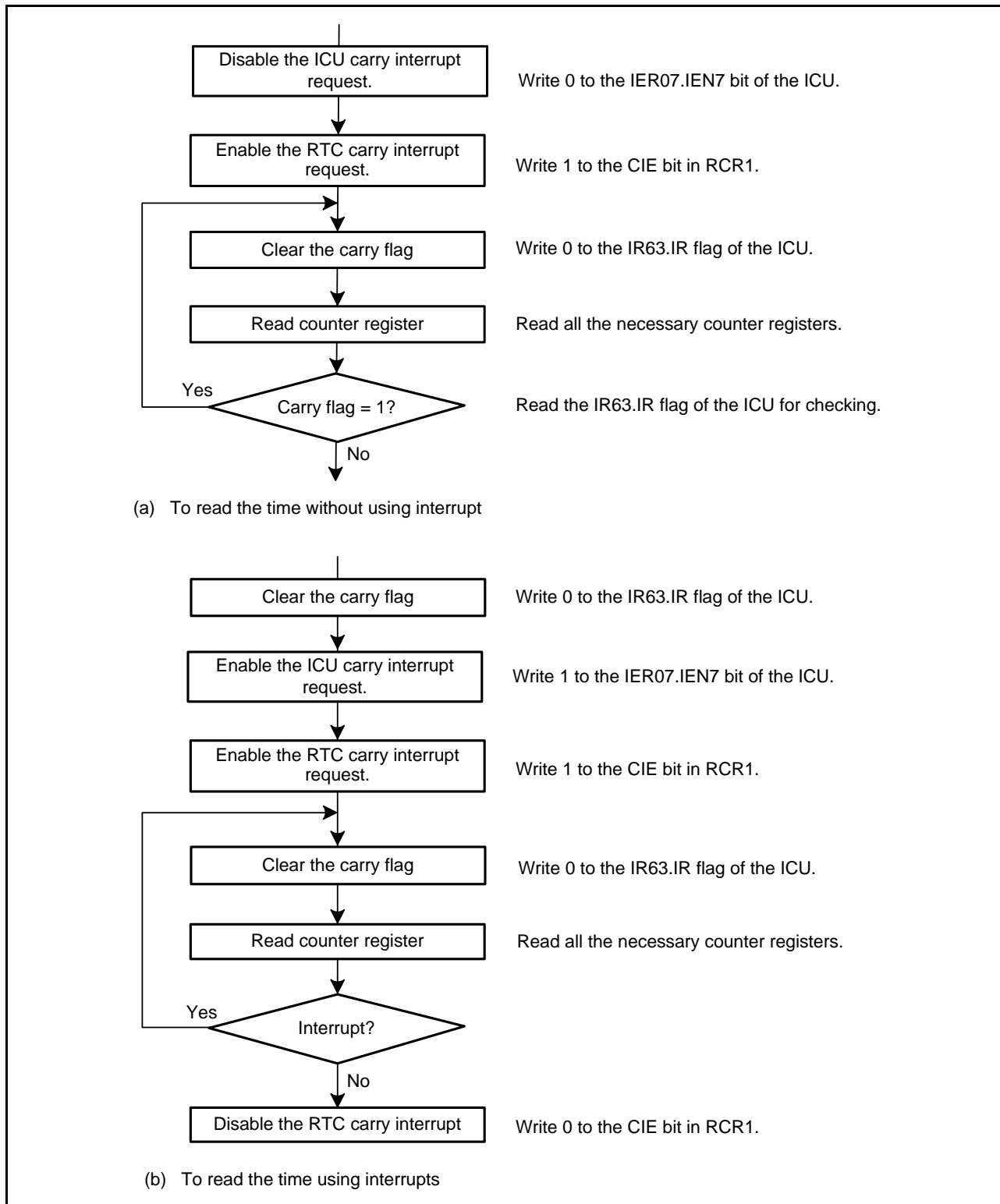


Figure 23.4 Reading Time

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 23.4, and the procedure using carry interrupts in (b). To keep the program simple, method (a) should be used in most cases.

23.3.5 Alarm Function

Figure 23.5 shows how to use the alarm function.

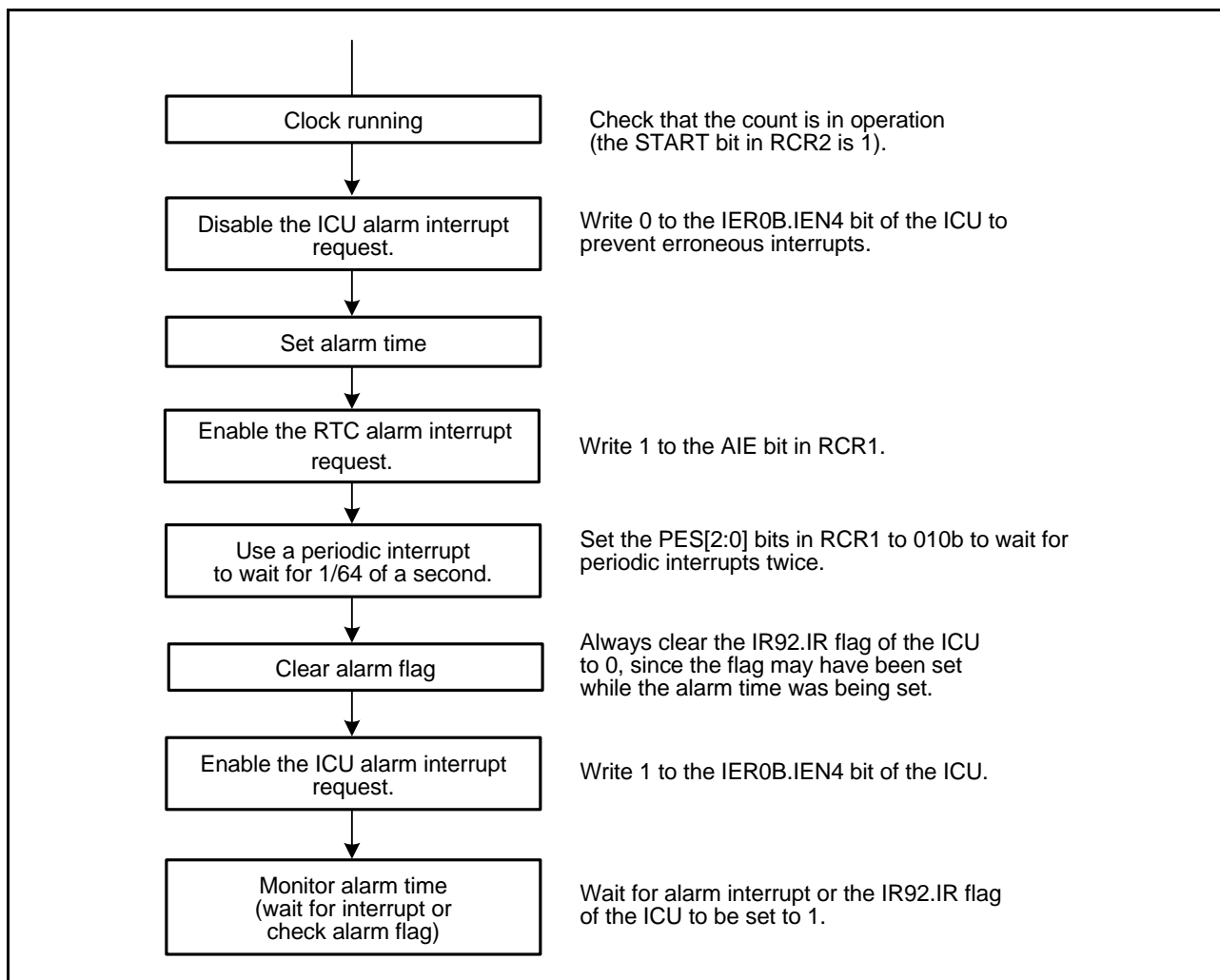


Figure 23.5 Using Alarm Function

An alarm can be generated by the year, month, date, day-of-week, hour, minute, or second value, or a combination of these. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

When the counter and the alarm time match, the IR92.IR flag of the ICU is set to 1. Alarm detection can be confirmed by reading this bit, but an interrupt should be used in most cases. If 1 has been set in the IER0B.IEN4 bit of the ICU, an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

Writing 0 clears the IR92.IR flag of the ICU.

When the counter and the alarm time match in a low power consumption state, this LSI returns from the low power consumption state.

Note: For the corresponding interrupt source number, see Table 23.4.

23.3.6 Procedure for Disabling Alarm Interrupt

Figure 23.6 shows the procedure for disabling the enabled alarm interrupt request.

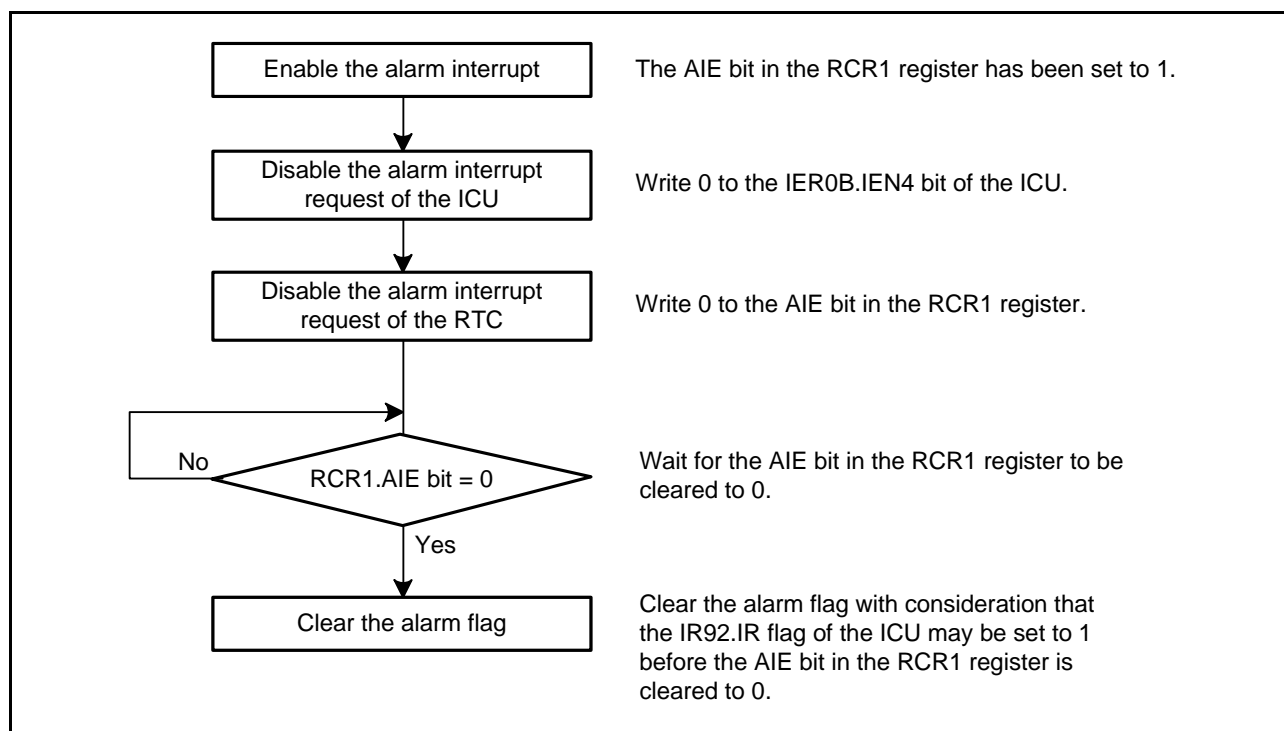


Figure 23.6 Procedure for Disabling Alarm Interrupt Request

23.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 23.4 lists interrupt sources for the RTC.

Table 23.4 RTC Interrupt Sources

Name	Interrupt Sources	Interrupt Status Flag
ALM	Alarm interrupt	IR92.IR
PRD	Periodic interrupt	IR62.IR
CUP	Carry interrupt	IR63.IR

(1) Alarm interrupt (ALM)

This interrupt is generated according to the result of comparison between the alarm registers and realtime clock counters (for details, refer to the description of each alarm register).

Since there is a possibility of the interrupt flag being set when the settings of the alarm registers match the clock counters, wait for 1/64th of a second after modifying values of the alarm registers, then clear the IR flag in IRi for the interrupt if an interrupt has occurred. Once the interrupt flag for the alarm interrupt has been cleared and the state has returned to non-matching of the alarm registers and clock counters, the flag will not be set again until there is a further match or the values of the alarm registers are modified again.

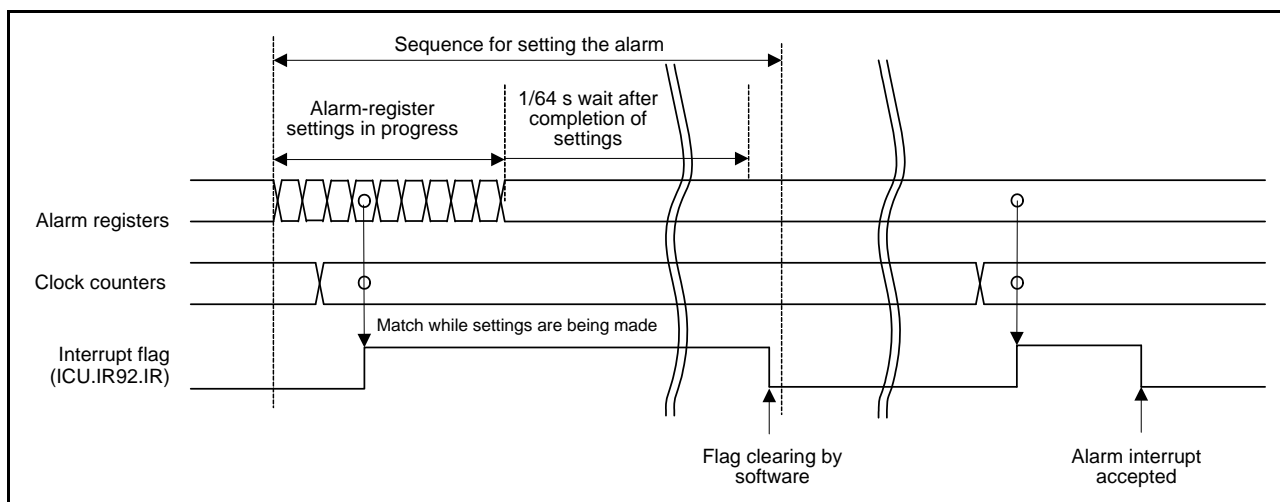


Figure 23.7 Timing Chart for the Alarm Interrupt (ALM)

(2) Periodic interrupt (PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/16 second, 1/64 second, or 1/256 second. The interrupt interval can be selected through the PES bit in RCR1.

(3) Carry interrupt (CUP)

This interrupt is generated when a carry to the second counter occurred or a carry to the 64-Hz counter occurred during read access to the 64-Hz counter.

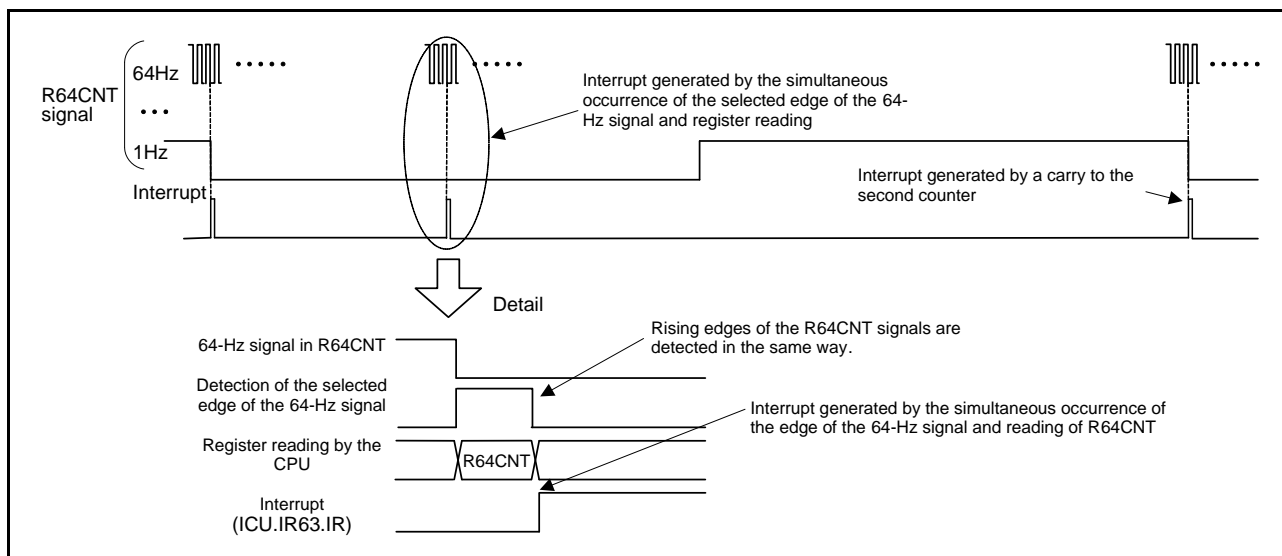


Figure 23.8 Carry Interrupt (CUP) Timing Chart

23.5 Usage Notes

23.5.1 Register Writing during Counting

The following registers should not be written to during counting (while the RCR2.START bit = 1).

RSECCNT, RMINCNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, RYRCNT

The counter must be stopped before writing to any of the above registers.

23.5.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 23.9.

The generation and period of the periodic interrupt can be changed by the setting of the PES[2:0] bits in RCR1. However, since the prescaler, R64CNT, and RSECCNT are used to generate interrupts, note that the interrupt period is not guaranteed immediately after setting of the PES[2:0] bits in RCR1. Furthermore, stopping/restarting or resetting counter operation and the 30-second adjustment by changing the RCR2 value affects the interrupt period.

For the corresponding interrupt source number, see Table 23.4.

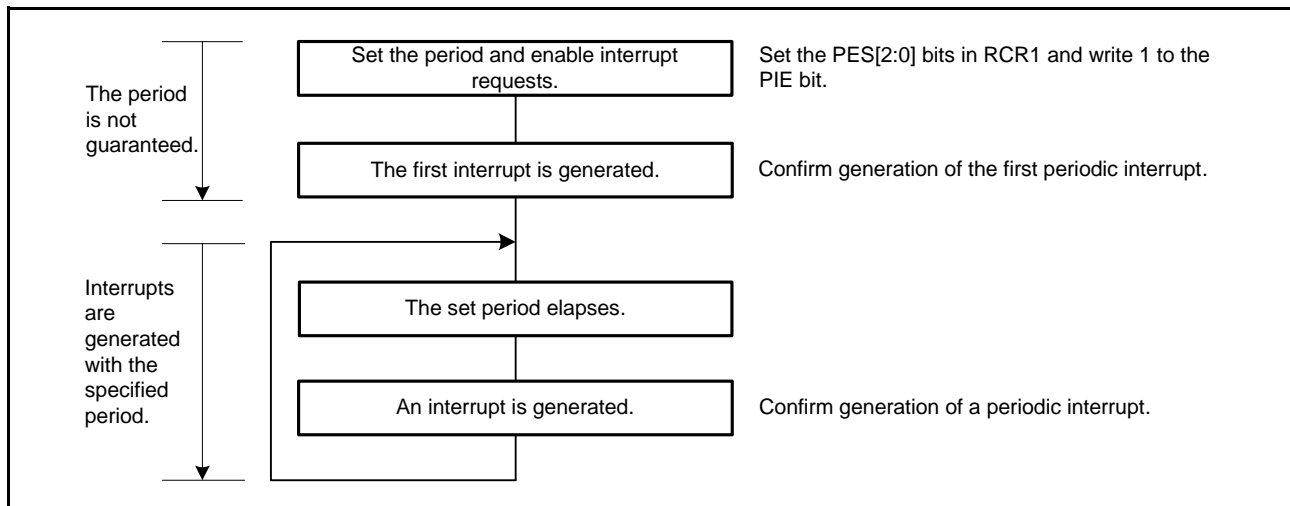


Figure 23.9 Using Periodic Interrupt Function

23.5.3 Transition to Low Power Consumption Modes after Setting Registers

When a transition to software standby mode or deep software standby mode is made after registers in this module have been set, correct counting will sometimes not be possible. After setting the registers, be sure to confirm that the setting has been successfully made before initiating a transition to software standby mode or deep software standby mode.

23.5.4 Points for Caution When Writing to and Reading from Registers

- When reading a counter register such as the second counter after having written to the counter register, follow the procedure in section 23.3.4, Reading 64-Hz Counter and Time.
- When reading the counters, alarm registers, or year alarm enable register after having written to the registers, perform three dummy read operations to ensure that the written value has been reflected in the register.
- When reading RCR1 (except for the AIE bit) and the RCR2.RTCOE bit after having written to the register, the written value can be read.
- When the operation shifts to the reset, software standby mode, or deep software standby mode while the count operation is stopped (the RCR2.START bit = 0), the correct value cannot be read from the time counter in return from the reset state, software standby mode, or deep software standby mode. To read the correct value from the timer counter in return from the mode, restart the counter (by setting the RCR2.START bit to 1) and wait for 1/128 second.
- In case when the value before shifting to the software standby mode or deep software standby mode is required in return from the mode, save the value before the operation shifts.
- When the operation shifts to the reset, software standby mode, or deep software standby mode while the count operation is in operation (the RCR2.START bit = 1), the correct value cannot be read from the time counter in return from the reset state, software standby mode, or deep software standby mode for one cycle of the 128-Hz clock. To read the correct value from the timer counter in such a case, wait for 1/128 second.

24. Watchdog Timer (WDT)

The watchdog timer (WDT) is an 8-bit timer that outputs an overflow signal (WDTOVF#) if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow. At the same time, the WDT can internally reset the RX62N/RX621.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

24.1 Overview

Table 24.1 lists the specifications of the WDT.

Figure 24.1 shows a block diagram of the WDT.

Table 24.1 Specifications of WDT

Item	Specifications
Count clocks	PCLK/4, PCLK/64, PCLK/128, PCLK/512, PCLK/2048, PCLK/8192, PCLK/32768, and PCLK/131072
Number of channels	8 bits x 1 channel
Counter clear	Write to TCNT
Operating modes	Switchable between watchdog timer mode and interval timer mode
Watchdog timer mode	Outputs a WDTOVF# signal when the counter overflows. Selectable whether or not to internally reset the LSI at the same time.
Interval timer mode	Generates an interval timer interrupt (WOVI) when the counter overflows.

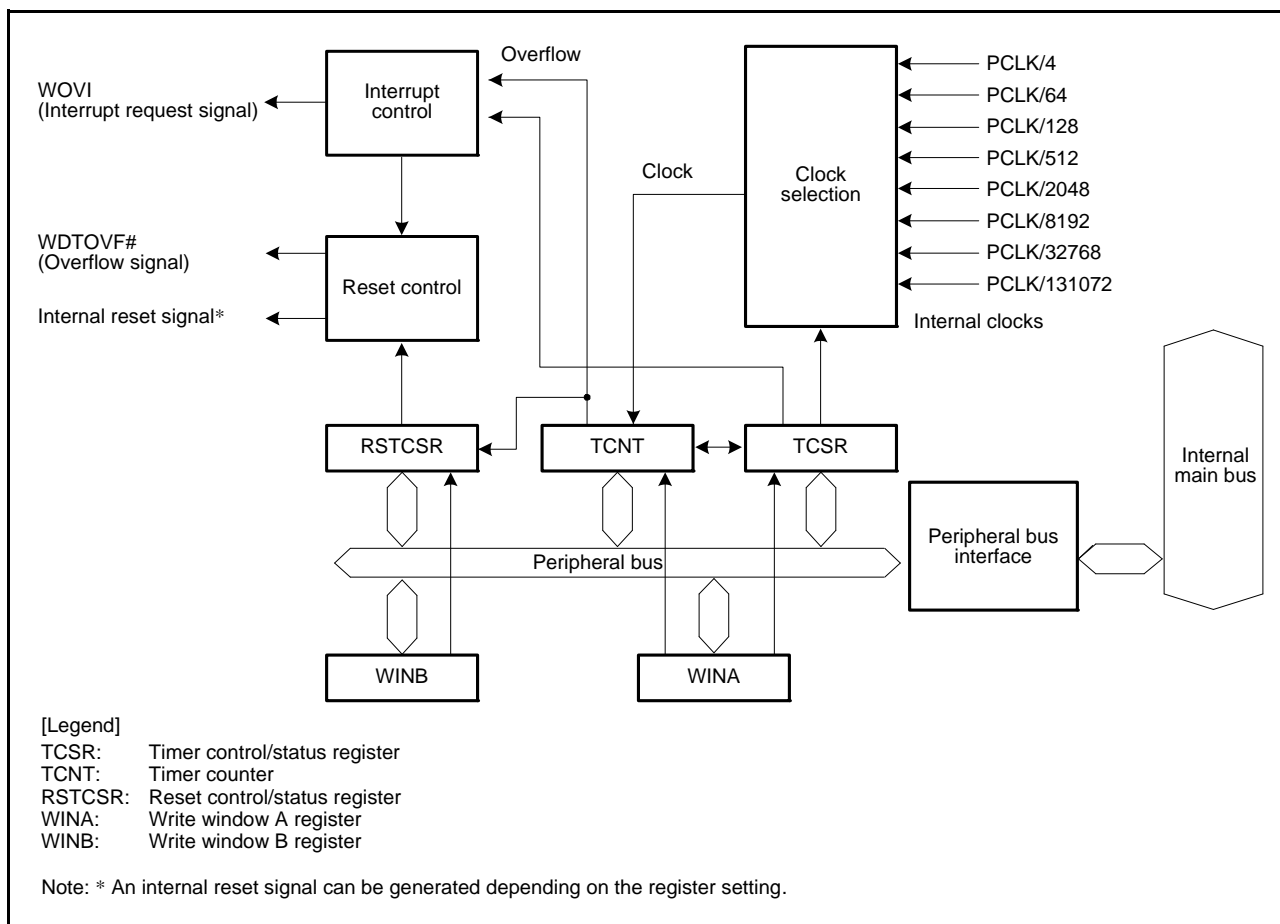


Figure 24.1 Block Diagram of WDT

Table 24.2 shows the input/output pin used for the WDT.

Table 24.2 Pin Configuration

Pin Name	I/O	Description
WDTOVF#	Output	Outputs a counter overflow signal in watchdog timer mode.

24.2 Register Descriptions

Table 24.3 lists the registers of the WDT.

Table 24.3 WDT Registers

Register Name	Symbol	Value after Reset	Address	Access Size
Timer control/status register	TCSR	x8h	0008 8028h*1	8
Timer counter	TCNT	00h	0008 8029h*1	8
Reset control/status register	RSTCSR	1Fh	0008 802Bh*1	8
Write window A register	WINA	—	0008 8028h*2	16
Write window B register	WINB	—	0008 802Ah*2	16

Note 1. Read-only register

Note 2. Write-only register

24.2.1 Timer Counter (TCNT)

Address: 0008 8029h



TCNT is an 8-bit up-counter for the internal clock.

TCNT is initialized to 00h when the TME bit in TCSR is set to 0.

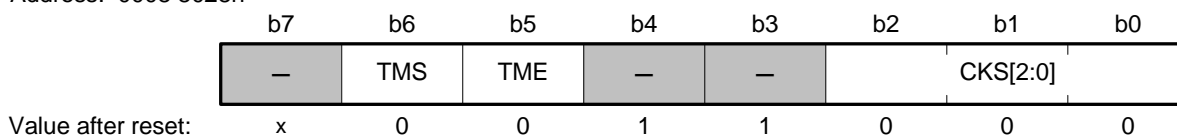
To read this counter, use 8-bit access.

To write to this counter, write data in WINA by 16-bit access.

For details, see section 24.5.1, Notes on Register Access.

24.2.2 Timer Control/Status Register (TCSR)

Address: 0008 8028h



[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKS[2:0]	Clock Select	b2 b0 0 0 0: PCLK/4 (cycle: 20.5 μ s) 0 0 1: PCLK/64 (cycle: 327.7 μ s) 0 1 0: PCLK/128 (cycle: 655.4 μ s) 0 1 1: PCLK/512 (cycle: 2.6 ms) 1 0 0: PCLK/2048 (cycle: 10.5 ms) 1 0 1: PCLK/8192 (cycle: 41.9 ms) 1 1 0: PCLK/32768 (cycle: 167.8 ms) 1 1 1: PCLK/131072 (cycle: 671.1 ms) Note: The overflow cycle for PCLK = 50 MHz is indicated in parentheses.	R/W
b4, b3	—	Reserved	These bits are always read as 1. The write value should always be 1.	R/W
b5	TME	Timer Enable	0: TCNT stops counting and is initialized to 00h. 1: TCNT starts counting.	R/W
b6	TMS	Timer Mode Select	0: Interval timer mode When TCNT overflows, an interval timer interrupt (WOVI) is requested. 1: Watchdog timer mode When TCNT overflows, WDTOVF# is output.	R/W
b7	—	Reserved	If read, an undefined value will be read. The write value should always be 1.	R/W

TCSR selects the clock source to be input to TCNT, and the timer mode.

To read this register, use 8-bit access.

To write to this register, write data to WINA by 16-bit access.

For details, see section 24.5.1, Notes on Register Access.

CKS[2:0] Bits (Clock Select)

These bits select the clock source to be input to TCNT.

TME Bit (Timer Enable)

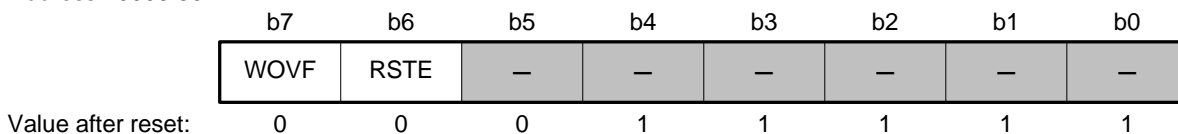
Selects whether TCNT starts or stops counting. When this bit is set to 1, TCNT starts counting. When this bit is cleared to 0, TCNT stops counting and is initialized to 00h.

TMS Bit (Timer Select)

Selects whether the WDT is used as a watchdog timer or interval timer.

24.2.3 Reset Control/Status Register (RSTCSR)

Address: 0008 802Bh



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are always read as 1. The write value should always be 1.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b6	RSTE	Reset Enable	0: The LSI is not reset internally when TCNT overflows in watchdog timer mode. (TCNT and TCSR of the WDT are reset.) 1: The LSI is internally reset when TCNT overflows in watchdog timer mode.	R/W
b7	WOVF	Watchdog Timer Overflow Flag	0: TCNT has not overflowed in watchdog timer mode. 1: TCNT has overflowed in watchdog timer mode.	R/(W)*

Note : * Only 0 can be written to this bit, to clear the flag.

RSTCSR controls the generation of the internal reset signal when TCNT overflows, and selects the type of internal reset signal.

RSTCSR is initialized to 1Fh by a reset signal from the RES# pin or a deep software standby reset, but not by the WDT internal reset signal caused by a WDToverflow.

To read this register, use 8-bit access.

To write to this register, write data in WINB by 16-bit access.

For details, see section 24.5.1, Notes on Register Access.

RSTE Bit (Reset Enable)

Selects whether or not this LSI is internally reset when TCNT overflows in watchdog timer mode.

WOVF Flag (Watchdog Timer Overflow)

Indicates that TCNT overflows in watchdog timer mode. This bit cannot be set to 1 in interval timer mode.

[Setting condition]

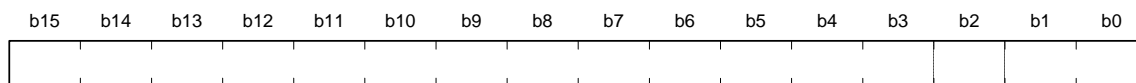
- When TCNT overflows (changed form FFh to 00h) in watchdog timer mode

[Clearing condition]

- Reading RSTCSR when WOVF = 1, and then writing 0 to WOVF

24.2.4 Write Window A Register (WINA)

Address: 0008 8028h



Value after reset: — — — — — — — — — — — — — — —

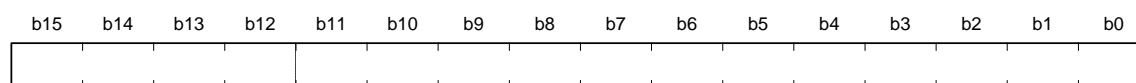
WINA is a write-only register for writing to TCNT and TCSR.

The writing method varies between TCNT and TCSR. For details, see section 24.5.1, Notes on Register Access.

To write to this register, use 16-bit access.

24.2.5 Write Window B Register (WINB)

Address: 0008 802Ah



Value after reset: — — — — — — — — — — — — — — —

WINB is a write-only register for writing to RSTCSR.

The writing method varies between writing 0 to the WOVF flag in RSTCSR and writing the RSTE bit in RSTCSR. For details, see section 24.5.1, Notes on Register Access. To write to this register, use 16-bit access.

24.3 Operation

24.3.1 Watchdog Timer Mode

To use the WDT in watchdog timer mode, set the TCSR.TMS bit to 1 (watchdog timer mode) and the TCSR.TME bit to 1 (TCNT starts counting).

During watchdog timer operation, if TCNT overflows without being rewritten because of a system crash or another error, the WDTOVF# signal is output. This ensures that TCNT does not overflow while the system is operating normally. Software must prevent TCNT overflows by rewriting the TCNT value (normally 00h is written) before overflow occurs. This WDTOVF# signal can be used to reset the LSI internally in watchdog timer mode.

If TCNT counter overflows in the watchdog timer mode, the RSTCSR.WOVF bit is set to 1. In addition, if TCNT counter overflows while the RSTCSR.RST bit is set to 1, a signal that resets the LSI internally is generated at the same time as the WDTOVF# signal is output. If a reset caused by a signal input to the RES# pin occurs at the same time as a reset caused by a WDT overflow, the RES# pin reset has priority and the WO VF flag in RSTCSR is cleared to 0.

The WDTOVF# signal is output for 257 cycles of PCLK when RSTE = 1, and for 256 cycles of PCLK when RSTE = 0. The internal reset signal is output for 1027 cycles of PCLK.

When RSTE = 1, an internal reset signal is generated. Since the system clock control register (SCKCR) is initialized, the multiplication ratio of PCLK becomes the initial value.

When RSTE = 0, an internal reset signal is not generated. Neither SCKCR nor the multiplication ratio of PCLK is changed.

When TCNT overflows in watchdog timer mode, the WO VF flag is set to 1. If TCNT overflows when the RSTE bit is set to 1, an internal reset signal is generated for the entire RX62N/RX621.

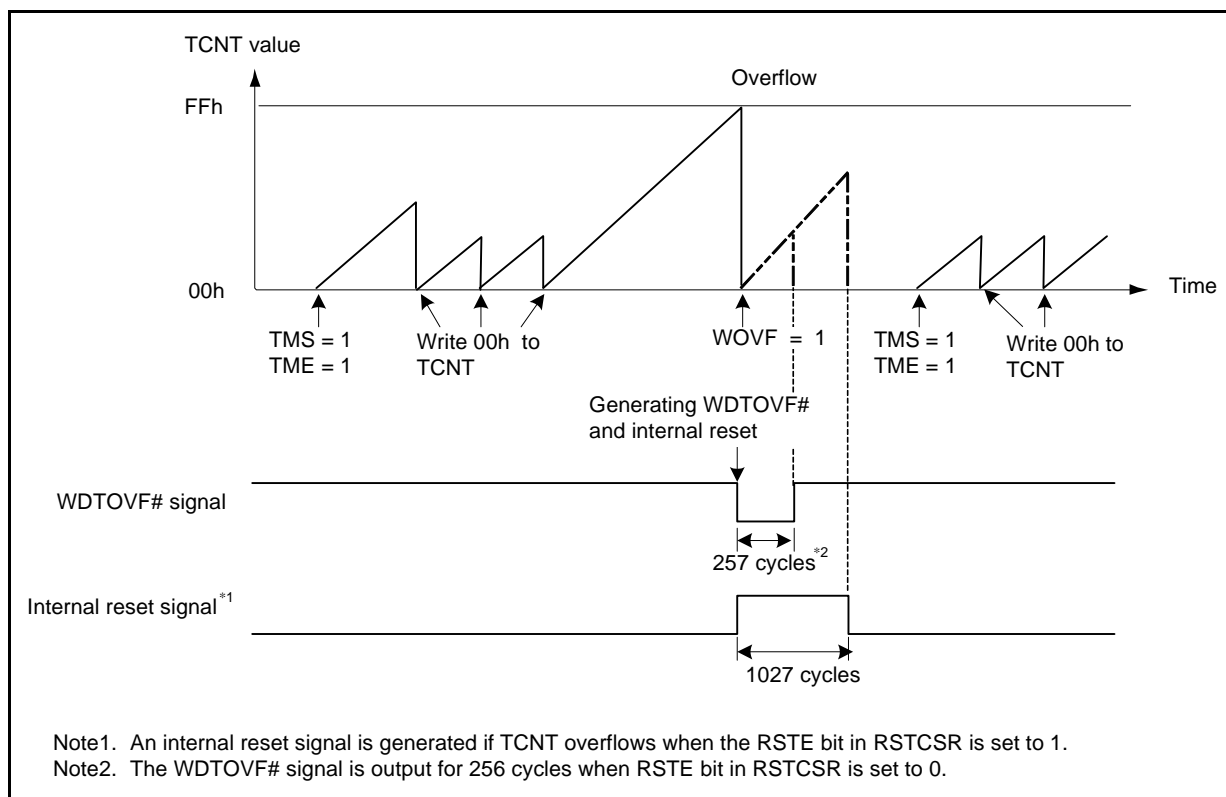


Figure 24.2 Operation in Watchdog Timer Mode

24.3.2 Interval Timer Mode

To use the WDT as an interval timer, set the TCSR.TMS bit to 0 (interval timer mode) and the TCSR.TME bit to 1 (TCNT starts counting).

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is generated each time TCNT overflows. Therefore, an interrupt can be generated at intervals.

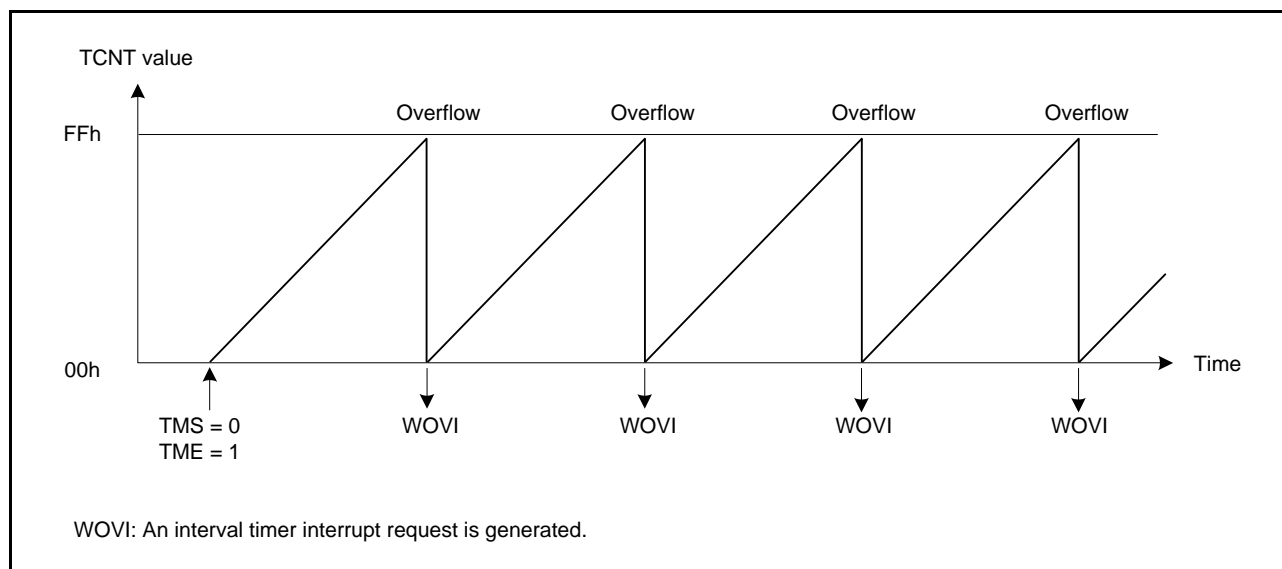


Figure 24.3 Operation in Interval Timer Mode

24.4 Interrupt Source

During interval timer mode operation, a TCNT overflow generates an interval timer interrupt (WOVI). For details, see section 11, Interrupt Control Unit (ICUa).

Table 24.4 WDT Interrupt Source

Name	Interrupt Source	Interrupt Status Flag	DTC Activation	DMACA Activation
WOVI	TCNT overflow	IR096.IR	Not possible	Not possible

24.5 Usage Notes

24.5.1 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR differ from other registers in being more difficult to write to.

(1) Writing to TCNT Counter, TCSR Register, and RSTCSR Register

When writing to TCNT and TCSR, be sure to use a word transfer instruction to the write window A register (WINA) (00088028h). For writing, TCNT and TCSR are assigned to the same address. Accordingly, perform data transfer as shown in Figure 24.4.

To write to TCNT, set 5Ah in the upper byte and data in the lower byte, and perform data transfer to TCNT.

To write to TCSR, set A5h in the upper byte and data in the lower byte, and perform data transfer to TCSR.

When writing to RSTCSR, use a word transfer instruction to the write window B register (WINB) (0008802Ah).

The method of writing 0 to the WOVF flag in RSTCSR differs from that of writing to the RSTE bit in RSTCSR. Perform data transfer as shown in Figure 24.4.

To write 0 to the WOVF flag, set A5h in the upper byte and 00h in the lower byte and write data in 16 bits, as shown in Figure 24.4. This writing has no effect on the RSTE bit.

To write to the RSTE bit, set 5Ah in the upper byte and the RSTCSR register write data in the lower byte and write data in 16 bits, as shown in Figure 24.4. This writing has no effect on the WOVF flag.

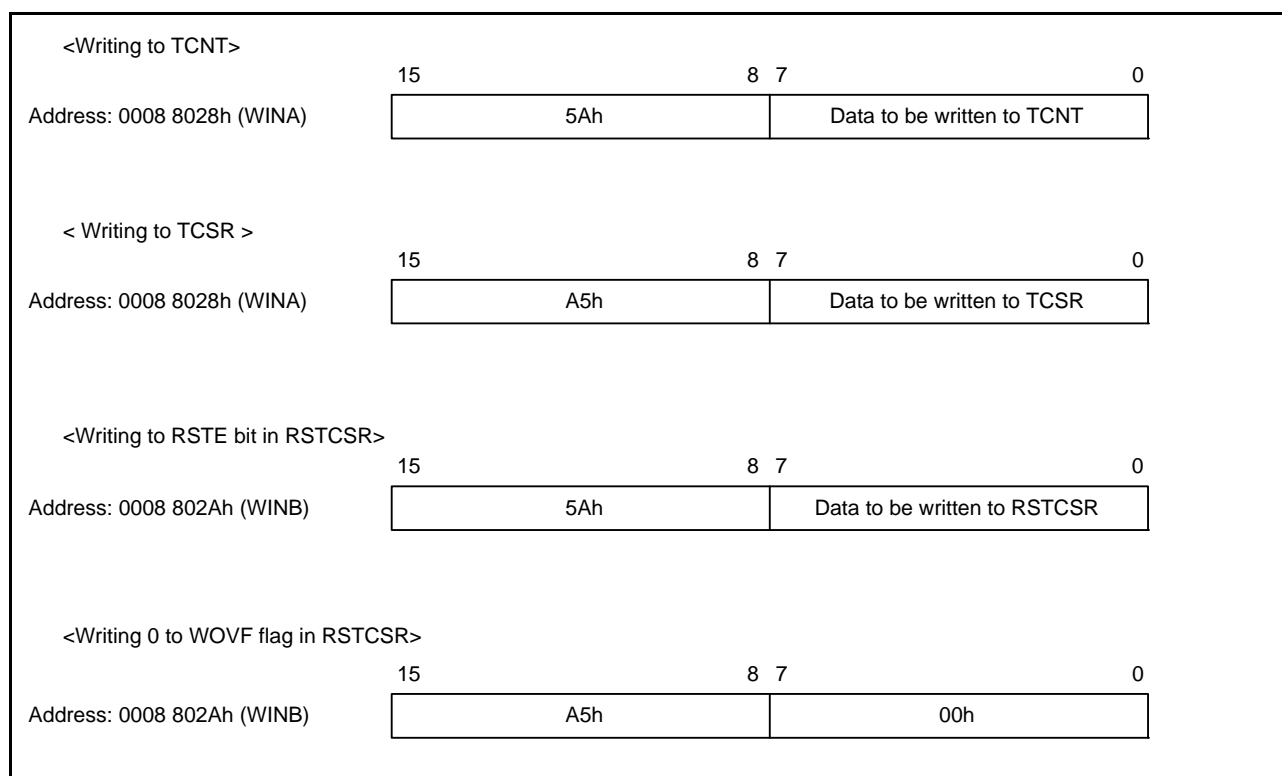


Figure 24.4 Writing to TCNT, TCSR, and RSTCSR

(2) Reading from TCNT Counter, TCSR Register, and RSTCSR Register

These counter and registers can be read from in the same way as other registers.

TCSR is assigned to address 00088028h, TCNT to address 00088029h, and RSTCSR to address 0008802Ah. For reading, use 8-bit access.

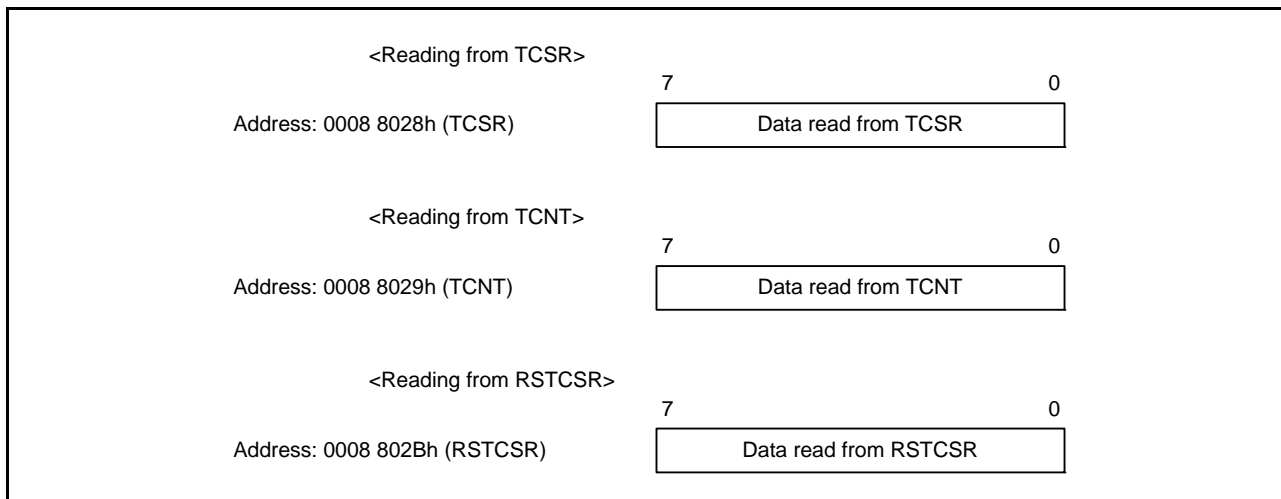


Figure 24.5 Reading from TCNT, TCSR, and RSTCSR

24.5.2 Conflict between Timer Counter (TCNT) Write and Increment

If a TCNT clock pulse is generated during a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 24.6 shows this operation.

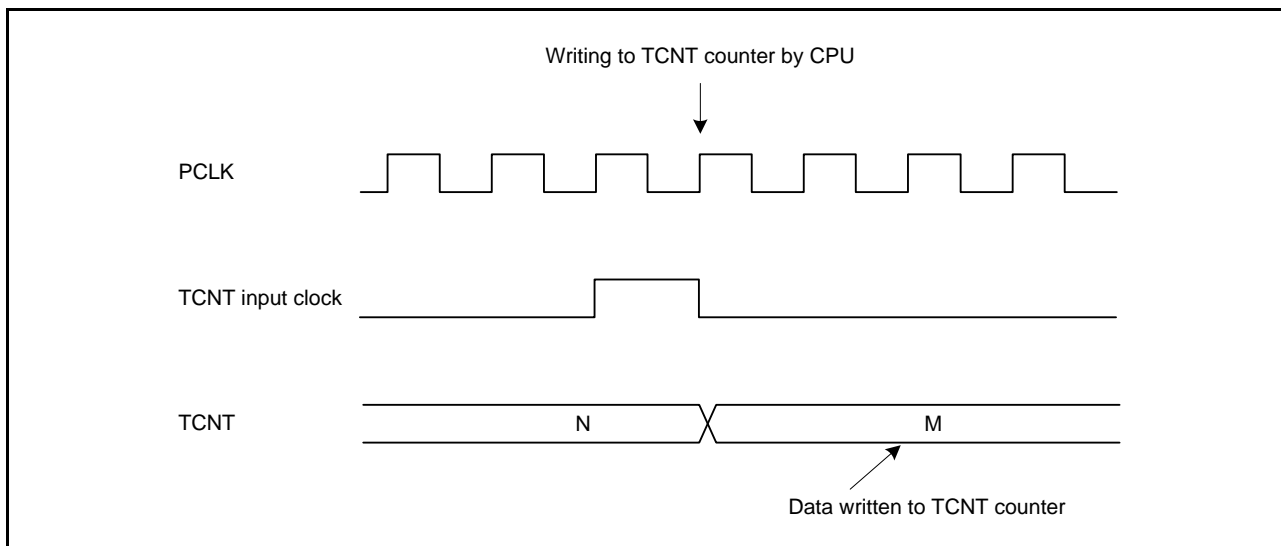


Figure 24.6 Conflict between TCNT Write and Increment

24.5.3 Changing Values of Bits CKS[2:0]

If bits CKS[2:0] bits in TCSR are written to while the WDT is operating, errors could occur in the incrementation. The watchdog timer must be stopped (by clearing the TME bit in TCSR to 0) before the values of CKS[2:0] bits in TCSR are changed.

24.5.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the timer mode is switched from watchdog timer mode to interval timer mode while the watchdog timer is operating, errors could occur in the operation. The watchdog timer must be stopped (by clearing the TME bit in TCSR to 0) before switching the timer mode.

24.5.5 Internal Reset in Watchdog Timer Mode

LSI is not reset internally if TCNT overflows while the RSTE bit in RSTCSR is cleared to 0 during watchdog time mode operation, but TCNT and the TCSR of the watchdog timer are reset.

TCNT, TCSR, and RSTCSR cannot be written to while the WDTOVF# signal is low. Also note that a read of the WOVF flag in RSTCSR is not recognized during this period. To clear the WOVF flag, therefore, read RSTCSR after the WDTOVF# signal goes high, then write 0 to the WOVF flag.

24.5.6 System Reset by WDTOVF# Signal

If the WDTOVF# signal is input to the RES# pin, the LSI will not be initialized correctly. Make sure that the WDTOVF# signal is not input logically to the RES# pin. To reset the entire system by means of the WDTOVF# signal, use a circuit like that shown in Figure 24.7.

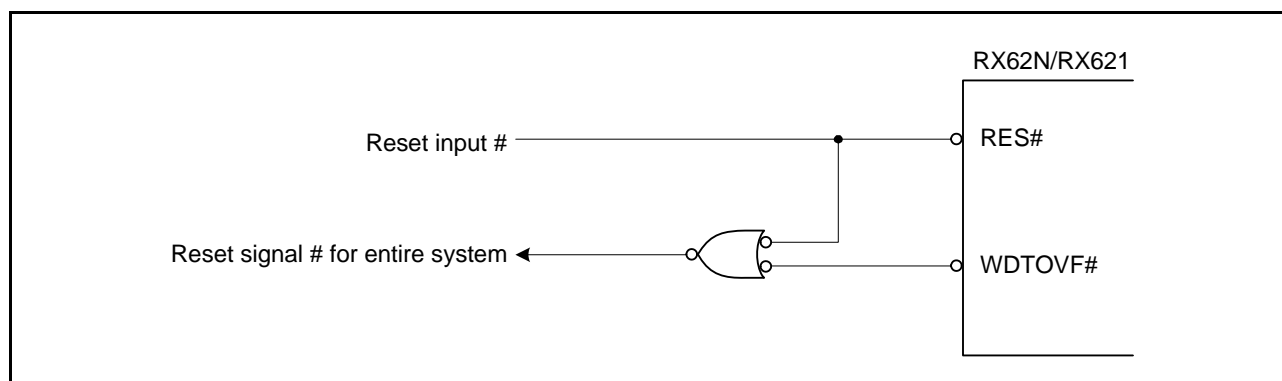


Figure 24.7 Circuit for System Reset by WDTOVF# Signal (Example)

24.5.7 Transition to Watchdog Timer Mode or Software Standby Mode

When the WDT operates in watchdog timer mode, a transition to software standby mode is not made even when the WAIT instruction is executed when the software standby bit (SSBY in SBYCR) in the standby control register is set to 1. Instead, a transition to sleep mode or all-module clock-stop mode is made.

To transit to software standby mode, the WAIT instruction must be executed after halting the watchdog timer (clearing the TME bit in TCSR to 0).

When the WDT operates in interval timer mode, a transition to software standby mode is made through execution of the WAIT instruction when the SSBY bit is set to 1. For details, see section 9, Low Power Consumption.

25. Independent Watchdog Timer (IWDT)

The independent watchdog timer (IWDT) is a watchdog timer for use independently of the conventional watchdog timer, which is for detecting programs entering runaway execution and system crashes.

The IWDT incorporates a 14-bit down-counter and resets the system if counting leads to an underflow of the value in the counter. The IWDT also has a refresh facility.

Note: The value in the IWDT must be refreshed before the counter underflows. For details, see section 25.3.3, Control of Refreshing

25.1 Overview

The specifications of the IWDT are given in Table 25.1. Figure 25.1 is a block diagram of the IWDT.

Table 25.1 Specifications of the IWDT

Item	Specifications
Clock for counting	IWDTCLK, IWDTCLK/16, IWDTCLK/32, IWDTCLK/64, IWDTCLK/128, IWDTCLK/256
Counter operation	Counting down by a 14-bit down-counter
Conditions for starting the counter	Counting can be started by refreshing the down-counter (write FFh after 00h has been written to the IWDTRR register).
Conditions for stopping the counter	<ul style="list-style-type: none"> Pin reset (the down-counter and other registers return to their initial values) Generation of an underflow
Reset-output sources of the IWDT	Underflow of the down-counter
Reading the value of the IWDT counter	The value reached in counting by the down-counter can be read out from a register (the IWDTSR).

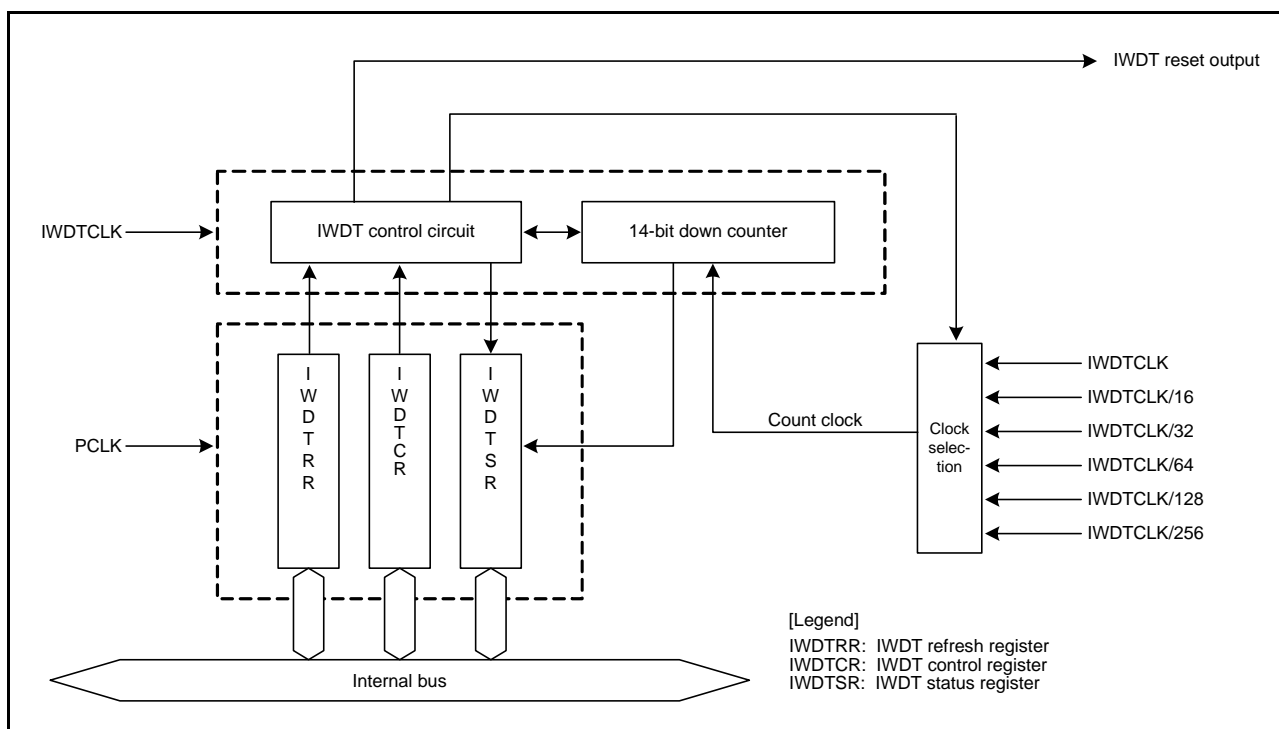


Figure 25.1 Block Diagram of the IWDT

25.2 Register Descriptions

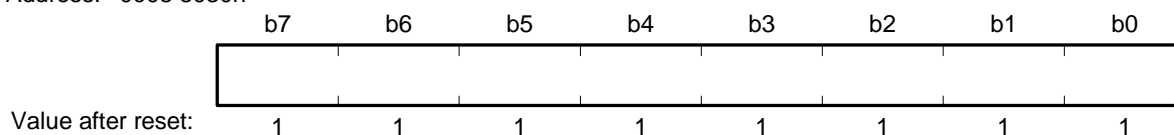
Table 25.2 lists the registers of the IWDT.

Table 25.2 IWDT Registers

Register Name	Symbol	Value after Reset	Address	Access Size
IWDT refresh register	IWDTRR	FFh	0008 8030h	8
IWDT control register	IWDTCR	3303h	0008 8032h	16
IWDT status register	IWDTSR	0000h	0008 8034h	16

25.2.1 IWDT Refresh Register (IWDTRR)

Address: 0008 8030h



IWDTRR refreshes the down-counter of the IWDT.

The down-counter of the IWDT is refreshed by writing 00h and then writing FFh to the IWDTRR (refreshing). After the counter has been refreshed, it starts counting down from the value selected by the TOPS[1:0] bits in the IWDT control register (IWDTCR).

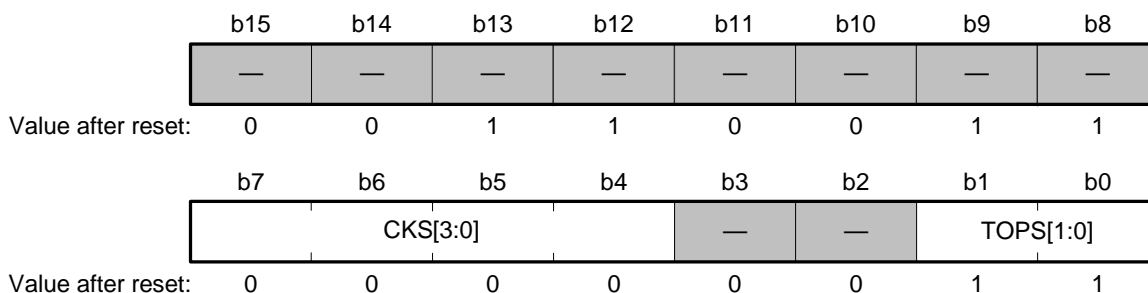
Furthermore, in the first reset operation after release from the reset state, counting down starts from the value selected by the TOPS[1:0] bits in IWDTCR.

Writing a value other than FFh after 00h invalidates the writing of 00h. To enable refreshing of the counter, 00h must be written to this register again, and this must be followed by writing of FFh.

Although reading the register after the writing of 00h will produce the value 00h, reading the register after writing any value other than 00h will always produce the value FFh.

25.2.2 IWDT Control Register (IWDTCR)

Address: 0008 8032h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Time-out Period Selection	b1 b0 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b4	CKS[3:0]	Clock Selection	b7 b4 0 0 --: IWDTCLK 0 1 0 0: IWDTCLK/16 0 1 0 1: IWDTCLK/32 0 1 1 0: IWDTCLK/64 0 1 1 1: IWDTCLK/128 1 -- --: IWDTCLK/256	R/W
b9, b8	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

IWDTCR specifies the time-out period for the underflow of the down-counter, and count clock.

Writing to the IWDTCR is only possible once during the period between release from the reset state and the first time the counter is refreshed. Writing to the IWDTCR is not possible after a refresh operation; furthermore, writing to the IWDTCR two or more times is not possible. This is because writing to the IWDTCR has been locked.

The writing to IWDTCR is unlocked by the reset source of the IWDT. With other reset sources, writing to the IWDTCR is not unlocked. For details, see section 6, Resets.

TOPS[1:0] Bits (Time-out Period Selection)

The TOPS[1:0] bits select the time-out period (period until the down-counter underflows) among 1024, 4096, 8192, or 16384 cycles, with taking the count clock specified by the CKS[3:0] bits as one cycle.

The combination of the settings of the CKS[3:0] and TOPS[1:0] bits determines the time (in cycles of the IWDTCLK) for the counter to underflow after being refreshed.

The relations between the settings of the CKS[3:0] and TOPS[1:0] bits, and the number of cycles of the IWDTCLK are shown in Table 25.3.

Table 25.3 Settings and Time-out Periods

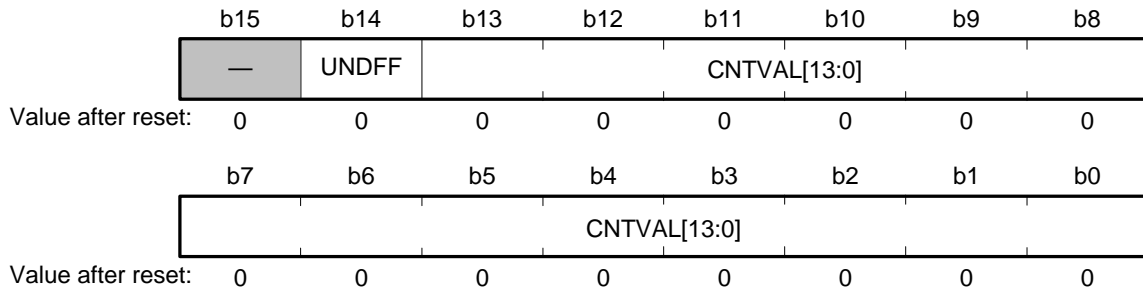
CKS[3:0]				TOPS[1:0]		Count Clock	Time-out Period (Number of Cycles)	Cycles of the IWDTCLK Signal
0	0	—	—	0	0	IWDTCLK	1024	1024
				0	1		4096	4096
				1	0		8192	8192
				1	1		16384	16384
0	1	0	0	0	0	IWDTCLK/16	1024	16384
				0	1		4096	65536
				1	0		8192	131072
				1	1		16384	262144
0	1	0	1	0	0	IWDTCLK/32	1024	32768
				0	1		4096	131072
				1	0		8192	262144
				1	1		16384	524288
0	1	1	0	0	0	IWDTCLK/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
0	1	1	1	0	0	IWDTCLK/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
1	—	—	—	0	0	IWDTCLK/256	1024	262144
				0	1		4096	1048576
				1	0		8192	2097152
				1	1		16384	4194304

CKS[3:0] Bits (Clock Selection)

These bits select the count clock for down counting among IWDTCLK, IWDTCLK/16, IWDTCLK/32, IWDTCLK/64, IWDTCLK/128, or IWDTCLK/256. In combination with the setting of the TOPS[1:0] bits, a counting period between 1024 and 4194304 cycles of the IWDTCLK signal is selected for the IWDT.

25.2.3 IWDT Status Register (IWDTSR)

Address: 0008 8034h



Bits	Symbol	Name	Function	R/W
b13 to b0	CNTVAL [13:0]	Down-Counter	Value counted by the down-counter	R
b14	UNDF	Underflow Flag	1: Underflow 0: No underflow	R/W
b15	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W

IWDTSR checks the counter value of the down-counter, and the generation state of the underflow.

IWDTSR is initialized by the reset source of the IWDT. With other reset sources, IWDTSR is not initialized. For details, see section 6, Resets.

CNTVAL[13:0] Bits (Down-Counter)

These bits can be read to confirm the value of the down-counter.

UNDF Flag (Underflow Flag)

This bit can be read to confirm the state of the down-counter in terms of whether or not an underflow has occurred.

The value "1" indicates that the down-counter has underflowed. The value "0" indicates that the down-counter has not underflowed.

Clear the UNDF bit by writing "0" to it. Writing "1" has no effect.

25.3 Description of Operation

25.3.1 Count Operation of the Down-Counter

Settings for the count clock and the time-out period are made in the IWDTCR after release from the reset state. Refreshing the down-counter then starts it counting down from the value selected by the setting of the IWDTCR.TOPS[1:0] bits.

After that, as long as the program continues in normal operation, the value in the counter is re-set when the counter is refreshed, and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the down-counter underflows because refreshing the down-counter is not possible due to the program having entered crashed execution, the IWDT outputs the reset signal.

After assertion of the reset signal, the down-counter is initialized (all bits cleared to "0") and kept in that state. After the system has been re-booted, counting down is again started by refreshing the counter.

An example of operation under the below conditions is shown in Figure 25.2.

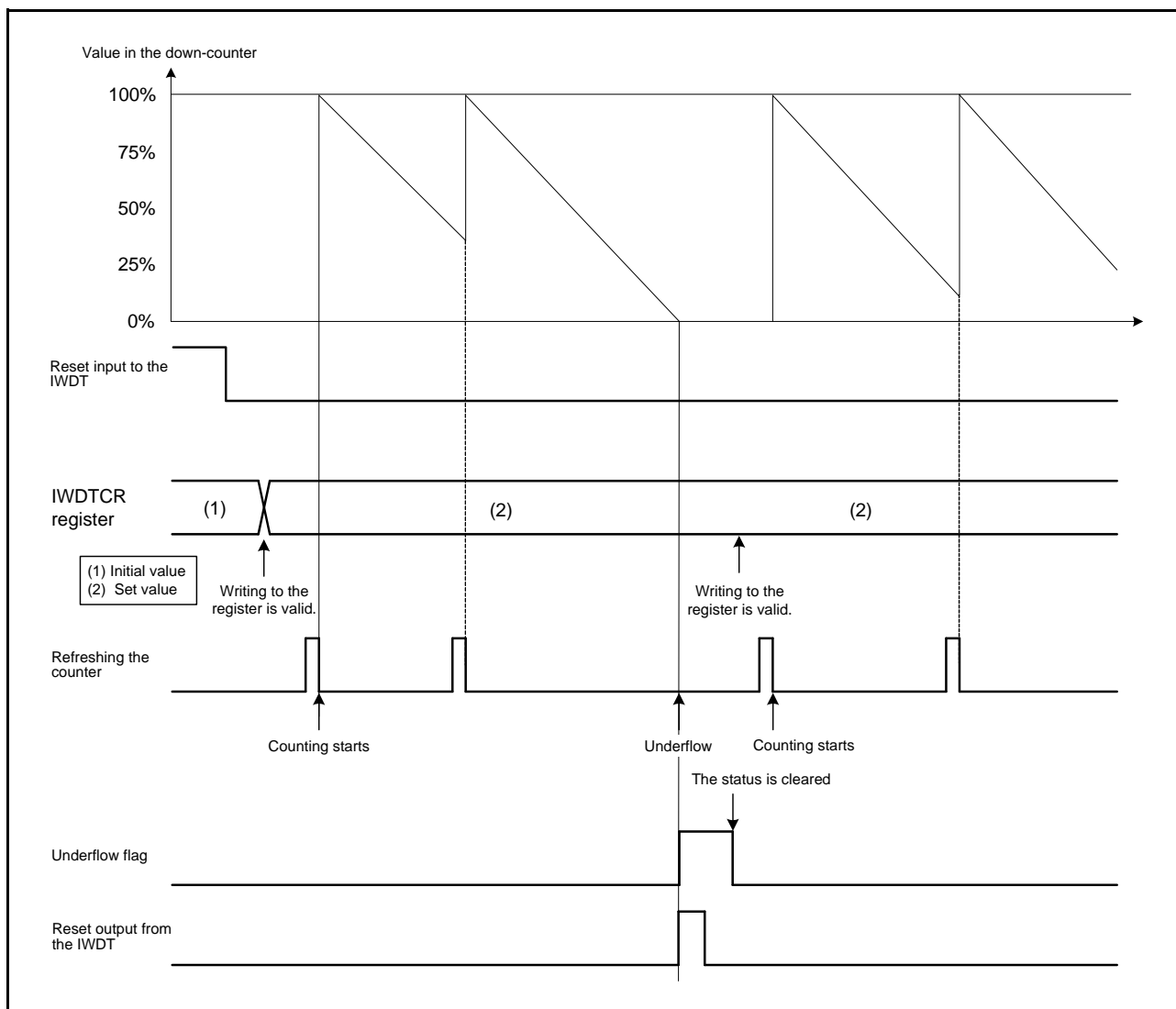


Figure 25.2 Example of Down-Counter Operation

25.3.2 Control over Writing to the IWDT Control Register (IWDTCR)

Writing to the IWDTCR is only possible for once after the reset state is canceled.

Writing to the IWDTCR makes the value of the internal register-lock signal of the IWDT become "1", locking the IWDTCR against subsequent attempts at writing.

The writing to IWDTCR is unlocked by the reset source of the IWDT. With other reset sources, writing to the IWDTCR is not unlocked. For details, see section 6, Resets.

Figure 25.3 is a chart of the waveforms in response to writing to the IWDTCR.

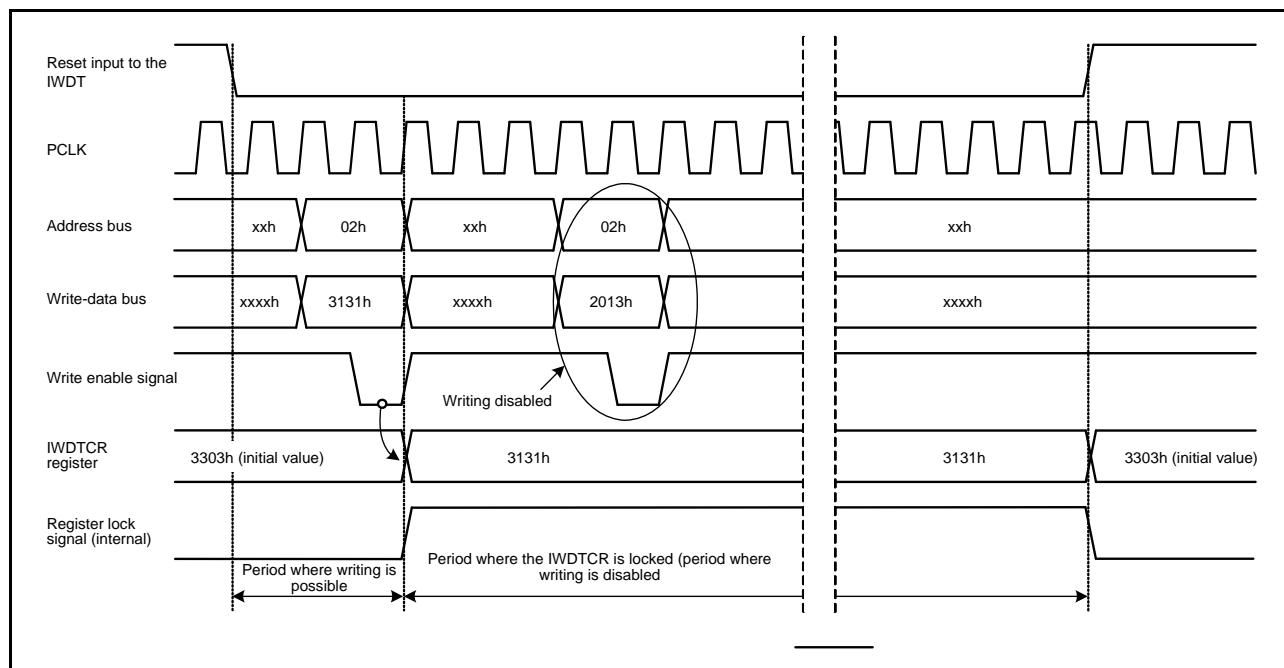


Figure 25.3 Control Waveforms in Response to Writing to the IWDTCR

25.3.3 Control of Refreshing

The IWDT starts operation (start of the down-counter) or the down-counter is refreshed by writing the values 00h and then FFh to the IWDT refresh register (IWDTRR). Write operations other than these (in order) are invalid. After invalid writing, correct refreshing is performed by writing 00h and then FFh to the IWDT refresh register (IWDTRR).

Though writing 00h → 00h is invalid, if FFh is written after that, writing 00h → FFh will be valid. Thus, writing 00h → 00h → FFh is valid. Moreover, if the first writing value is not 00h, writing will be valid if the operation contains the set of writing 00h → FFh.

Sample sequences of writing that are invalid for refreshing the counter

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (a value other than FFh) → FFh

After writing of FFh to the IWDTRR register, refreshing the down-counter requires up to four cycles of the signal for counting (the IWDTCR.CKS[3:0] bits determine how many cycles of the signal from the IWDTCLK make up one cycle for counting). Therefore, writing FFh to the IWDTRR should be completed four cycles before the counter underflows. The value of the counter is checked by the counter bits (IWDTSR.CNTVAL[13:0]).

Figure 25.4 shows refresh-operation waveforms for the count clock of the IWDTCLK.

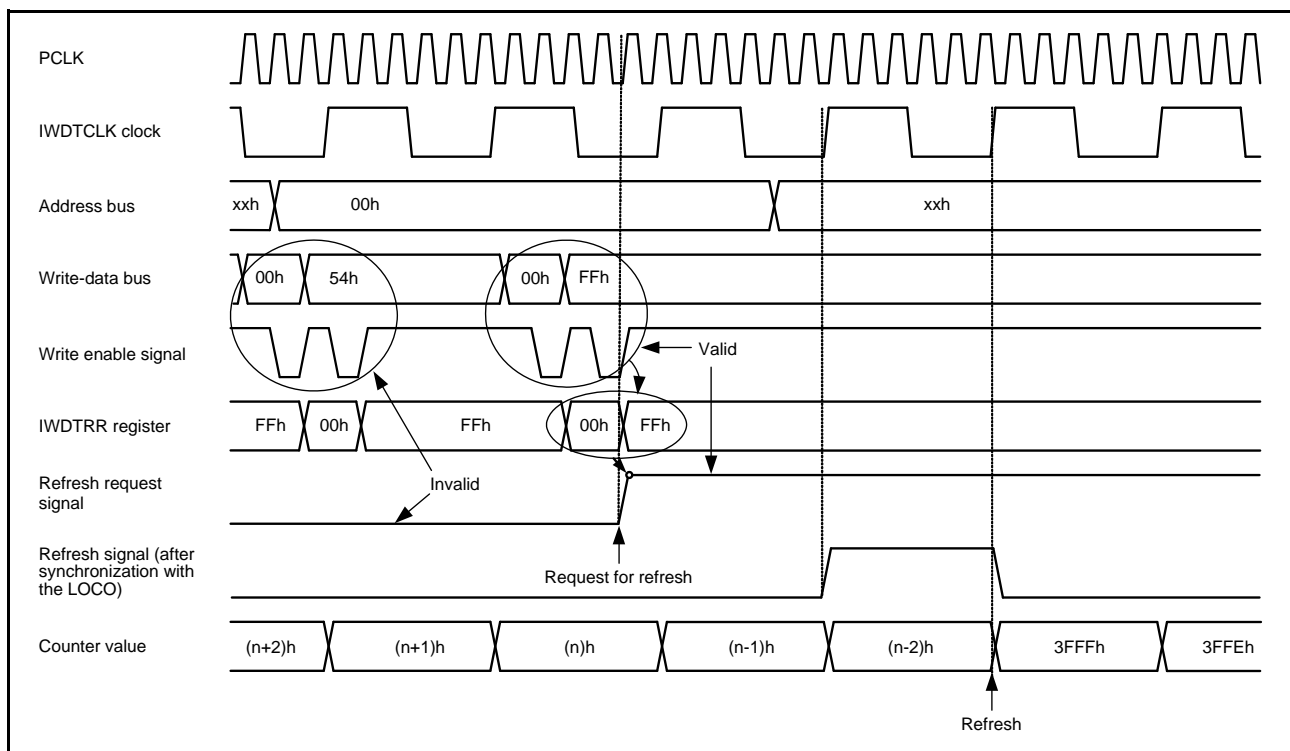


Figure 25.4 Refresh Operation Waveform (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

25.3.4 Status Flags

Reset requests from the refresh-error flag (IWDTSR.REFEF) and underflow flag (IWDTSR.UNDF) are retained after output of the reset signal from the IWDT.

Thus, after release from the reset state, read the IWDTSR.REFEF and UNDF bits to check for these reset requests.

For both flags, writing "0" clears the bit and writing "1" has no effect.

Leaving the status flags set does not affect operation. If the flags are not cleared, at the time of the next reset from the IWDT, the earlier reset request is cleared and the new reset request is written.

25.4 Usage Notes

25.4.1 Limitation on Transitions to Low-Power-Consumption Modes

Writing to the IWDTCR or refresh operations brings the IWDT into use. When the IWDT is in use, even executing a WAIT instruction while the SSBY bit in the SBYCR is set to 1 will not cause a transition to software standby mode; instead, the transition will be to sleep mode or all-module-clock-stop mode. Furthermore, the reset source for the IWDT triggers release from the given mode, while the other reset sources do not. For details, see section 6, Resets.

26. Ethernet Controller (ETHERC)

26.1 Overview

The RX62N Group has an on-chip Ethernet controller (ETHERC) conforming to the Ethernet or IEEE802.3 MAC (Media Access Control) layer standard. Connecting a physical-layer LSI (PHY-LSI) complying with this standard enables the ETHERC to perform transmission and reception of Ethernet/IEEE802.3 frames. The ETHERC has one MAC layer interface port. The ETHERC is connected to the Ethernet direct memory access controller for Ethernet controller (EDMAC) inside this LSI, and carries out high-speed data transfer to and from the memory.

Table 26.1 shows the specifications of the ETHERC, Figure 26.1 shows a configuration of the ETHERC, and Table 26.2 shows the pin configuration of the ETHERC.

Table 26.1 Specifications of ETHERC

Item	Description
Protocol	• Flow control conforming to IEEE802.3x
Data transmission and reception	• Transmission and reception of Ethernet/IEEE802.3 frames
Transfer rate	• Supports 10- and 100-Mbps transfer
Mode	• Supports full-duplex and half-duplex modes
Interface	• Conforms to the MII (Media Independent Interface) and RMII (Reduced Media Independent Interface) specifications of the IEEE802.3u standard
Function	• Magic Packet™* detection and Wake-On-LAN (WOL) signal output

Note: * Magic Packet™ is a registered trademark of Advanced Micro Devices, Inc.

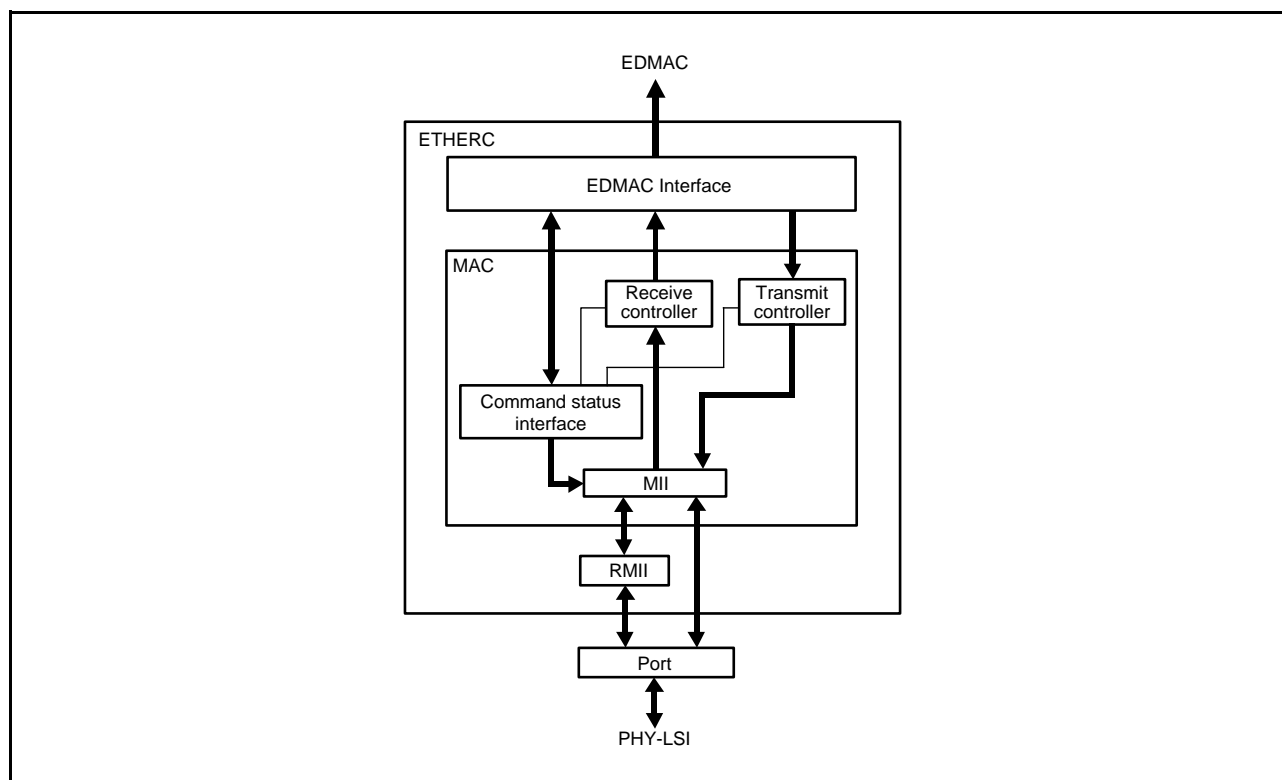


Figure 26.1 Configuration of ETHERC

Table 26.2 Pin Configuration (1) (MII Mode)

Name	I/O	Function
ET_TX_CLK*	Input	Transmit clock ET_TX_EN, ET_ETXD3 to ET_ETXD0, and ET_TX_ER timing reference signal
ET_RX_CLK*	Input	Receive clock ET_RX_DV, ET_ERXD3 to ET_ERXD0, and ET_RX_ER timing reference signal
ET_TX_EN*	Output	Transmit enable signal Indicates that transmit data is ready on MII_TXD3 to MII_TXD0
ET_ETXD3 to ET_ETXD0*	Output	4-bit transmit data
ET_TX_ER*	Output	Sends error state occurred during data reception to the PHY-LSI
ET_RX_DV*	Input	Indicates that valid receive data is on ET_ERXD3 to ET_ERXD0
ET_ERXD3 to ET_ERXD0*	Input	4-bit receive data
ET_RX_ER*	Input	Receive error Identifies error state occurred during data reception
ET_CRD*	Input	Carrier detection signal
ET_COL*	Input	Collision detection signal
ET_MDC*	Output	Reference clock signal for information transfer via ET_MDIO
ET_MDIO*	I/O	Bidirectional signal for exchange of management information between this LSI and PHY-LSI
ET_LINKSTA	Input	Inputs link status from PHY-LSI
ET_EXOUT	Output	External output pin
ET_WOL	Output	Wake-On-LAN signal indicating reception of Magic Packet™

Note : * MII signal conforming to IEEE802.3u

Table 26.3 Pin Configuration (2) (RMII Mode)

Name	I/O	Function
ET_MDC	Output	Reference clock signal for information transfer via ET_MDIO
ET_MDIO	I/O	Bidirectional signal for exchange of management information between this LSI and PHY-LSI
ET_WOL	Output	Wake-On-LAN signal indicating reception of Magic Packet™
ET_LINKSTA	Input	Inputs link status from PHY-LSI
ET_EXOUT	Output	External output pin
REF50CK*1	Input	RMII_TXD_EN, RMII_TXD1 to RMII_TXD0, RMII_CRD_DV, RMII_RXD1 to RMII_RXD0, and RMII_RX_ER timing reference signal
RMII_TXD1 and RMII_TXD0*1	Output	2-bit transmit data
RMII_TXD_EN*1	Output	Indicates that transmit data is ready on RMII_TXD1 and RMII_TXD0
RMII_RXD1 and RMII_RXD0*1	Input	2-bit receive data
RMII_RX_ER*1	Input	Identifies error state occurred during data reception
RMII_CRD_DV*1	Input	Carrier detection signal/indication of valid receive data on RMII_RXD1 and RMII_RXD0

Note 1. *RMII signals

26.2 Register Descriptions

Table 26.4 lists the registers of the ETHERC.

Table 26.4 Registers of ETHERC

Register Name	Symbol	Value after Reset	Address	Access Size
ETHERC mode register	ECMR	0000 0000h	000C 0100h	32
Receive frame length register	RFLR	0000 0000h	000C 0108h	32
ETHERC status register	ECSR	0000 0000h	000C 0110h	32
ETHERC interrupt permission register	ECSIPR	0000 0000h	000C 0118h	32
PHY interface register	PIR	0000 0000h	000C 0120h	32
PHY status register	PSR	0000 0000h	000C 0128h	32
Random number generation counter upper limit setting register	RDMLR	0000 0000h	000C 0140h	32
IPG register	IPGR	0000 0014h	000C 0150h	32
Automatic PAUSE frame register	APR	0000 0000h	000C 0154h	32
Manual PAUSE frame register	MPR	0000 0000h	000C 0158h	32
PAUSE Frame receive counter register	RFCF	0000 0000h	000C 0160h	32
Automatic PAUSE frame retransmit count register	TPAUSER	0000 0000h	000C 0164h	32
PAUSE frame retransmit counter register	TPAUSECR	0000 0000h	000C 0168h	32
Broadcast frame receive count setting register	BCFRR	0000 0000h	000C 016Ch	32
MAC address high register	MAHR	0000 0000h	000C 01C0h	32
MAC address low register	MALR	0000 0000h	000C 01C8h	32
Transmit retry over counter register	TROCR	0000 0000h	000C 01D0h	32
Delayed collision detect counter register	CDCR	0000 0000h	000C 01D4h	32
Lost carrier counter register	LCCR	0000 0000h	000C 01D8h	32
Carrier not detect counter register	CNDCR	0000 0000h	000C 01DCh	32
CRC error frame receive counter register	CEFCR	0000 0000h	000C 01E4h	32
Frame receive error counter register	FRECR	0000 0000h	000C 01E8h	32
Too-short frame receive counter register	TSFRCR	0000 0000h	000C 01ECh	32
Too-long frame receive counter register	TLFRCR	0000 0000h	000C 01F0h	32
Residual-bit frame receive counter register	RFCR	0000 0000h	000C 01F4h	32
Multicast address frame receive counter register	MAFCR	0000 0000h	000C 01F8h	32

26.2.1 ETHERC Mode Register (ECMR)

Address: 000C 0100h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—	—	—	—	TPC	ZPF	PFR	RXF	TXF
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
—	—	—	PRCEF	—	—	MPDE	—	—	RE	TE	—	ILB	RTM	DM	PRM
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	PRM	Promiscuous Mode	0: ETHERC performs normal operation 1: ETHERC performs promiscuous mode operation	R/W
b1	DM	Duplex Mode	0: Half-duplex transfer is specified. 1: Full-duplex transfer is specified.	R/W
b2	RTM	Transmission/Reception Rate	0: 10 Mbps 1: 100 Mbps	R/W
b3	ILB	Internal Loop Back Mode	0: Normal data transmission/reception is performed. 1: Data loopback is performed inside the MAC in the ETHERC when DM = 1.	R/W
b4	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b5	TE	Transmission Enable	0: Transmitting function is disabled. 1: Transmitting function is enabled.	R/W
b6	RE	Reception Enable	0: Receiving function is disabled. 1: Receiving function is enabled.	R/W
b8, b7	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b9	MPDE	Magic Packet™ Detection Enable	0: Magic Packet™ detection is not enabled. 1: Magic Packet™ detection is enabled.	R/W
b11, b10	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b12	PRCEF	CRC Error Frame Reception Enable	0: A frame with a CRC error is received as a frame with an error. 1: A frame with a CRC error is received as a frame without an error.	R/W
b15 to b13	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b16	TXF	Operating Mode for Transmitting Port Flow Control	0: PAUSE frame detection is disabled. (Automatic PAUSE frame is not transmitted) 1: Flow control for the transmitting port is enabled. (Automatic PAUSE frame is transmitted as required)	R/W
b17	RXF	Operating Mode for Receiving Port Flow Control	0: PAUSE frame detection is disabled 1: Flow control for the receiving port is enabled	R/W
b18	PFR	PAUSE Frame Receive Mode	0: PAUSE frame is not transferred to EDMAC 1: PAUSE frame is transferred to EDMAC	R/W
b19	ZPF	PAUSE Frame Usage with TIME = 0 Enable	0: Control of a PAUSE frame whose TIME parameter value is 0 is disabled 1: Control of a PAUSE frame whose TIME parameter value is 0 is enabled	R/W
b20	TPC	PAUSE Frame Transmission	0: PAUSE frame is not transmitted in a PAUSE period 1: PAUSE frame is transmitted even in a PAUSE period	R/W
b31 to b21	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

ECMR specifies the operating mode of the ETHERC. The settings in ECMR should be made in the initialization process after a reset in most cases.

The operating mode setting must not be changed while the transmitting and receiving functions are enabled. To switch the operating mode, return the ETHERC and EDMAC to their initial states by means of the software reset bit (SWR) in the EDMAC mode register (EDMR) of the EDMAC before making settings again.

PRM Bit (Promiscuous Mode)

Setting the PRM bit enables all Ethernet frames to be received. "All Ethernet frames" means all receivable frames, irrespective of differences or enabled/disabled status of destination address, broadcast address, multicast bit, etc.

RTM Bit (Transmission/Reception Rate)

This bit specifies the transmission and reception bit rate when RMII is selected.

TE Bit (Transmission Enable)

If a switch is made from transmitting function enabled (TE = 1) to disabled (TE = 0) while a frame is being transmitted, the transmitting function will be enabled until transmission of the corresponding frame is completed.

RE Bit (Reception Enable)

If a switch is made from receiving function enabled (RE = 1) to disabled (RE = 0) while a frame is being received, the receiving function will be enabled until reception of the corresponding frame is completed.

MPDE Bit (Magic Packet™ Detection Enable)

This bit enables or disables Magic Packet™ detection by hardware to allow activation from the Ethernet.

ZPF Bit (PAUSE Frame Usage with TIME = 0 Enable)

When the ZPF bit is set to 0, the next frame is not transmitted until the time specified by the Timer value has elapsed. On receiving a PAUSE frame with a Timer value of 0, the PAUSE frame is discarded.

When the ZPF bit is cleared to 0, if the data size in the receive FIFO becomes smaller than the setting of the flow control start FIFO threshold setting register (FCFTR) of the EDMAC before the time specified by the Timer value elapses, an automatic PAUSE frame with a Timer value of 0 is transmitted. On receiving a PAUSE frame with a Timer value of 0, the transmission wait state is canceled.

26.2.2 ETHERC Status Register (ECSR)

Address: 000C 0110h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	BFR	PSRTO	—	LCHNG	MPD	ICD
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	ICD	Illegal Carrier Detection	0: PHY-LSI has not detected an illegal carrier on the line 1: PHY-LSI has detected an illegal carrier on the line	R/W
b1	MPD	Magic Packet™ Detection	0: Magic Packet™ has not been detected 1: Magic Packet™ has been detected	R/W
b2	LCHNG	Link Signal Change	0: Change in the LINKSTA signal has not been detected 1: Change in the LINKSTA signal has been detected (high to low or low to high)	R/W
b3	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b4	PSRTO	PAUSE Frame Retransmit Retry Over	0: PAUSE frame retransmit count has not exceeded the upper limit 1: PAUSE frame retransmit count has exceeded the upper limit	R/W
b5	BFR	Continuous Broadcast Frame Reception	Continuous Broadcast Frame Reception Interrupt (Interrupt Source) Indicates that Broadcast frames have been received continuously.	R/W
b31 to b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

ECSR indicates the status in the ETHERC. Each state can be notified to the CPU by interrupts. When 1 is written to the PSRTO, LCHNG, MPD, and ICD bits, the corresponding flags can be cleared. Writing 0 does not affect the flags. For bits that generate interrupts, the interrupt can be enabled or disabled by the corresponding bit in the ETHERC interrupt permission register (ECSIPR).

The interrupts generated due to ECSR are indicated in the ETHERC status register source bit (ECI) in ETHERC/EDMAC status register (EESR) of the EDMAC.

ICD Bit (Illegal Carrier Detection)

This bit indicates that the PHY-LSI has detected an illegal carrier on the line. More specifically, the ICD bit is set to 1 when the signals transmitted from the PHY-LSI to the RX62N through the ET_RX_DV, ET_RX_ER, and ET_ERXD3 to ET_ERXD0 pins are 0, 1, and 1110, respectively (see Figure 26.9). If the signal input from the PHY-LSI changes in a period shorter than the period required for recognition by software, correct information may not be obtained. Refer to the timing specification of the PHY-LSI used.

LCHNG Bit (Link Signal Change)

This bit indicates that the ET_LINKSTA signal input from the PHY-LSI has changed from high to low or low to high.

To check the current Link state, refer to the LNKSTA pin status bit (LMON) in the PHY status register (PSR).

PSRTO Bit (PAUSE Frame Retransmit Retry Over)

This bit indicates whether the retransmit count for retransmitting a PAUSE frame when flow control is enabled has exceeded the retransmit upper limit set in the automatic PAUSE frame retransmit count register (TPAUSER).

26.2.3 ETHERC Interrupt Permission Register (ECSIPR)

Address: 000C 0118h

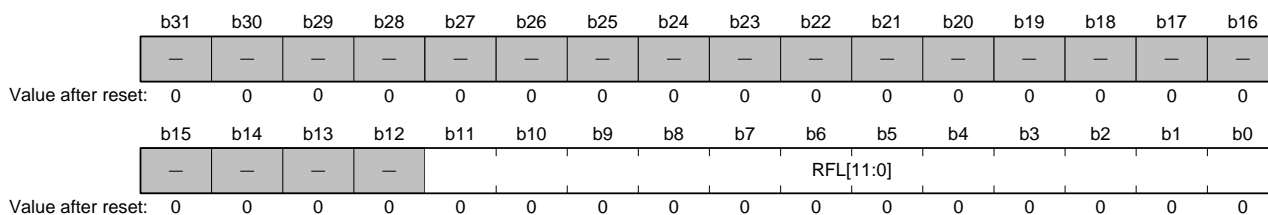
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	BFSIPR	PSRTOIP	—	LCHNGIP	MPDIP	ICDIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ICDIP	Illegal Carrier Detect Interrupt Enable	0: Interrupt notification by the ICD bit is disabled 1: Interrupt notification by the ICD bit is enabled	R/W
b1	MPDIP	Magic Packet™ Detect Interrupt Enable	0: Interrupt notification by the MPD bit is disabled 1: Interrupt notification by the MPD bit is enabled	R/W
b2	LCHNGIP	LINK Signal Change Interrupt Enable	0: Interrupt notification by the LCHNG bit is disabled 1: Interrupt notification by the LCHNG bit is enabled	R/W
b3	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b4	PSRTOIP	PAUSE Frame Retransmit Retry Over Interrupt Enable	0: Interrupt notification by the PSRTO bit is disabled 1: Interrupt notification by the PSRTO bit is enabled	R/W
b5	BFSIPR	Continuous Broadcast Frame Reception Interrupt Enable	0: Interrupt notification by the corresponding bit in ECSR is disabled 1: Interrupt notification by the corresponding bit in ECSR is enabled	R/W
b31 to b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

ECSIPR enables or disables the interrupt sources indicated by ECSR. Each bit can disable or enable interrupts corresponding to the bits in ECSR.

26.2.4 Receive Frame Length Register (RFLR)

Address: 000C 0108h



Bit	Symbol	Bit Name	Description	R/W
b11 to b0	RFL[11:0]	Receive Frame Length 11 to 0	000h to 5EEh: 1,518 bytes 5EFh: 1,519 bytes 5F0h: 1,520 bytes : : 7FFh: 2,047 bytes 800h to FFFh: 2,048 bytes The frame data described here refers to all fields from the destination address up to the CRC data, but actually, frame contents from the destination address up to the data are transferred to memory; CRC data is not included in the transfer. When the received data exceeds the specified length, the part of data that exceeds the specified length is discarded.	R/W
b31 to b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RFLR specifies the maximum frame length (in bytes) that can be received by the RX62N. The RFLR settings must not be changed while the receiving function is enabled.

RFL[11:0] Bits (Receive Frame Length 11 to 0)

The RFL[11:0] bits specify a value for frame length to be checked. When the received data exceeds the specified value, a frame length error occurs.

26.2.5 PHY Interface Register (PIR)

Address: 000C 0120h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
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	MDC	MII/RMII Management Data Clock	The value set in the MDC bit is output from the ET_MDC pin to supply the MII/RMII with the management data clock.	R/W
b1	MMD	MII/RMII Management Mode	0: Read direction is specified 1: Write direction is specified	R/W
b2	MDO	MII/RMII Management Data-Out	Stores the data for output from the ET_MDIO pin. The data are output on the ET_MDIO pin when the MMD bit is 1 (specifying the direction as writing) but not when the MMD bit is 0 (specifying the direction as reading).	R/W
b3	MDI	MII/RMII Management Data-In	Indicates the level of the ET_MDIO pin. The write value should always be 0.	R/W
b31 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

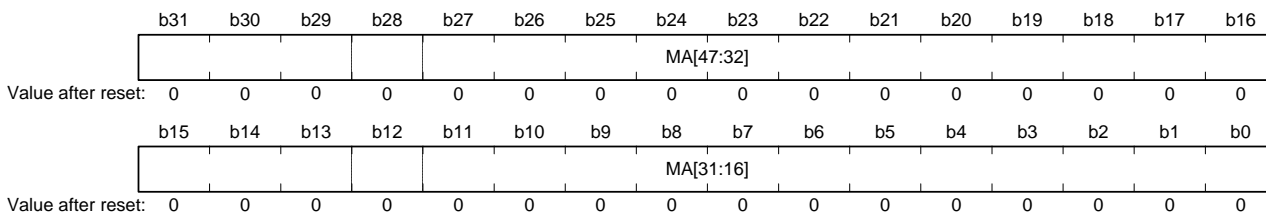
PIR provides a means of accessing the PHY-LSI internal registers via the MII/RMII.

MDC Bit (MII/RMII Management Data Clock)

The value set in the MDC bit is output from the ET_MDC pin to supply the MII/RMII with the management data clock. For the method of accessing the MII/RMII registers, see section 26.3.4, Accessing MII/RMII Registers .

26.2.6 MAC Address High Register (MAHR)

Address: 000C 01C0h

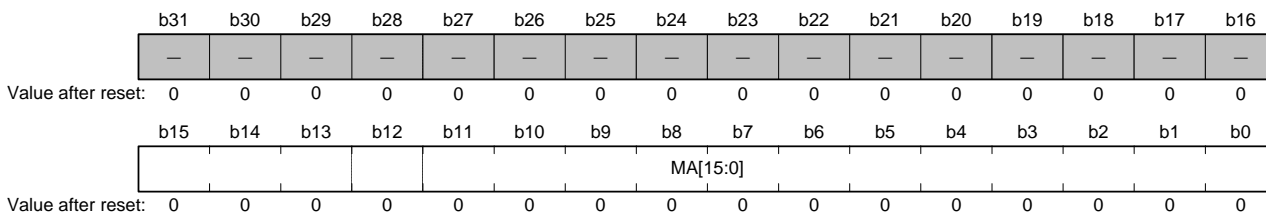


Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MA[47:16]	MAC Address Bits 47 to 16	These bits specify the upper 32 bits of the MAC address. For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set 01234567h in MAHR.	R/W

MAHR specifies the upper 32 bits of the 48-bit MAC address. The settings in MAHR should be made in the initialization process after a reset in most cases. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. Return the ETHERC and EDMAC to their initial states by means of the software reset bit (SWR) in the EDMAC mode register (EDMR) of the EDMAC before making settings again.

26.2.7 MAC Address Low Register (MALR)

Address: 000C 01C8h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MA[15:0]	MAC Address Bits 15 to 0	These bits specify the lower 16 bits of the MAC address. For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set 89ABh in MALR.	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

MALR specifies the lower 16 bits of the 48-bit MAC address. The settings in MALR should be made in the initialization process after a reset in most cases. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. Return the ETHERC and EDMAC to their initial states by means of the software reset bit (SWR) in the EDMAC mode register (EDMR) before making settings again.

26.2.8 PHY Status Register (PSR)

Address: 000C 0128h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LMON
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	LMON	LINKSTA Pin Status	The Link status can be read by connecting the Link signal output from the PHY-LSI to the EX_LINKSTA pin. For the polarity, refer to the specifications of the PHY-LSI to be connected.	R
b31 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R

PSR is used to read interface signals from the PHY-LSI.

26.2.9 Transmit Retry Over Counter Register (TROCR)

Address: 000C 01D0h

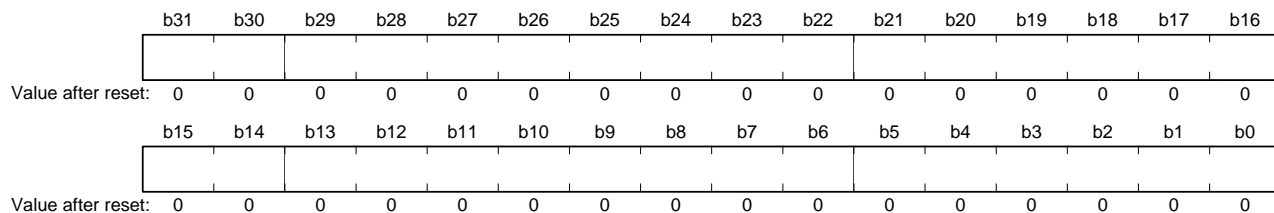
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the number of frames that were unable to be transmitted in 16 transmission attempts including the retransfer.	R/W

TROCR is a counter that indicates the number of frames that were unable to be transmitted in 16 transmission attempts including the retransfer. When 16 transmission attempts have failed, TROCR is incremented by 1. When the value in TROCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to TROCR with any value.

26.2.10 Delayed Collision Detect Counter Register (CDCR)

Address: 000C 01D4h

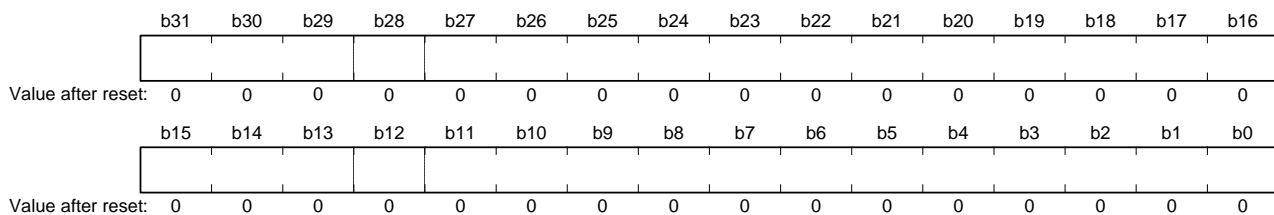


Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the number of all delayed collisions after the start of data transmission.	R/W

CDCR is a counter that indicates the number of all delayed collisions that occurred on the line after the start of data transmission. When the value in CDCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to CDCR with any value.

26.2.11 Lost Carrier Counter Register (LCCR)

Address: 000C 01D8h

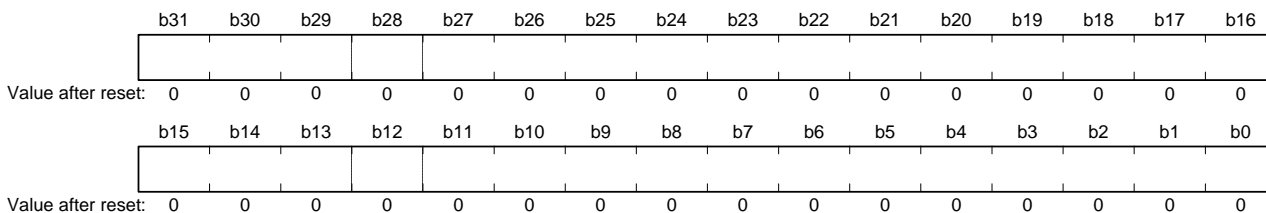


Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the number of times the carrier was lost during data transmission.	R/W

LCCR is a counter that indicates the number of times the carrier was lost during data transmission. When the value in LCCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to LCCR with any value.

26.2.12 Carrier Not Detect Counter Register (CNDCR)

Address: 000C 01DCh

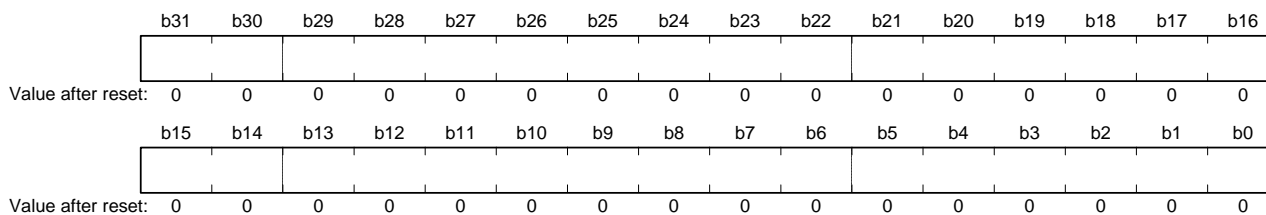


Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the number of times carrier was not detected.	R/W

CNDCR is a counter that indicates the number of times carrier was not detected during preamble transmission. When the value in CNDCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to CNDCR with any value.

26.2.13 CRC Error Frame Receive Counter Register (CEFCR)

Address: 000C 01E4h

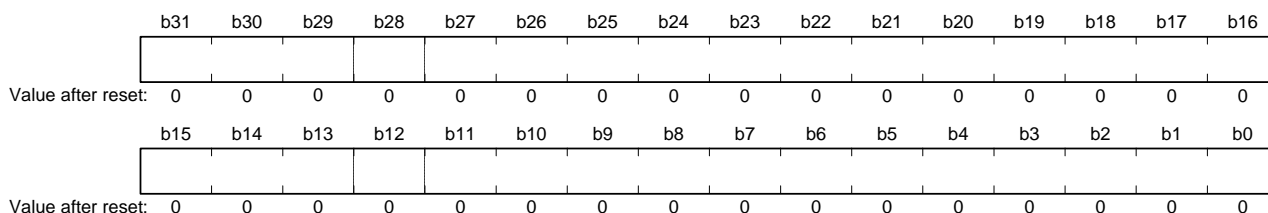


Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the number of CRC error frames received.	R/W

CEFCR is a counter that indicates the number of times a frame with a CRC error was received. When the value in CEFCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to the CEFCR register with any value.

26.2.14 Frame Receive Error Counter Register (FRECR)

Address: 000C 01E8h

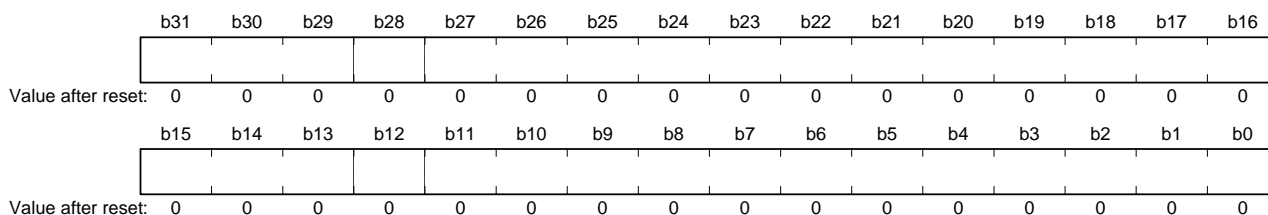


Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the number of errors during frame reception.	R/W

FRECR is a counter that indicates the number of frames for which a receive error was generated by the signal input to the ET_RX_ER pin from the PHY-LSI. FRECR is incremented each time the ET_RX_ER signal becomes active. When the value in FRECR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to FRECR with any value.

26.2.15 Too-Short Frame Receive Counter Register (TSFRCR)

Address: 000C 01ECh

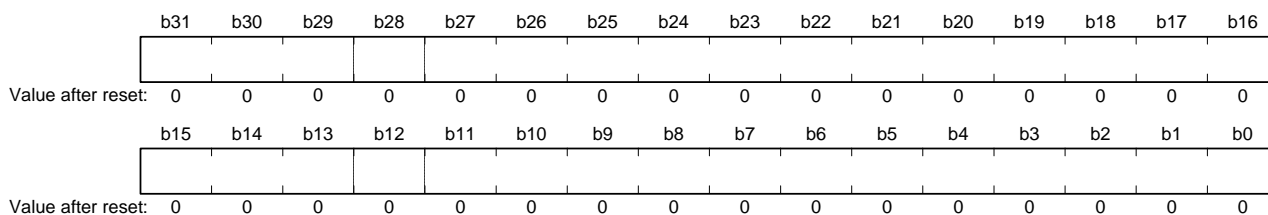


Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the number of frames received with a length of less than 64 bytes.	R/W

TSFRCR is a counter that indicates the number of frames received with a length less than 64 bytes. When the value in TSFRCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to TSFRCR with any value.

26.2.16 Too-Long Frame Receive Counter Register (TLFRCR)

Address: 000C 01F0h

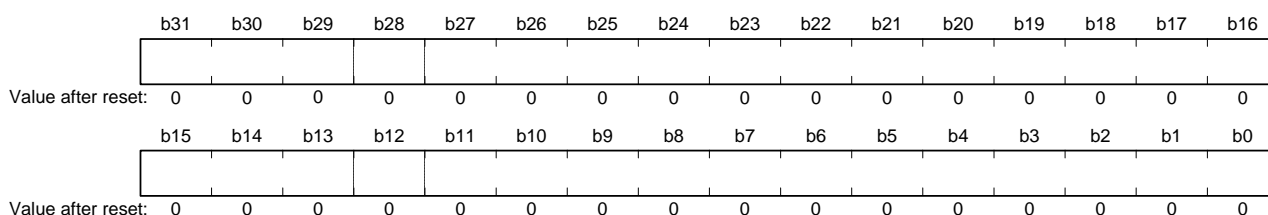


Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the number of frames received with a length exceeding the value in RFLR.	R/W

TLFRCR is a counter that indicates the number of frames received with a length exceeding the value specified by RFLR. When the value in TLFRCR reaches FFFFFFFFh, the counter stops incrementing. TLFRCR is not incremented when a frame containing residual bits is received. In this case, the reception of the frame is indicated in the residual-bit frame receive counter register (RFCR). The counter value is cleared to 0 by a write to TLFRCR with any value.

26.2.17 Residual-Bit Frame Receive Counter Register (RFCR)

Address: 000C 01F4h

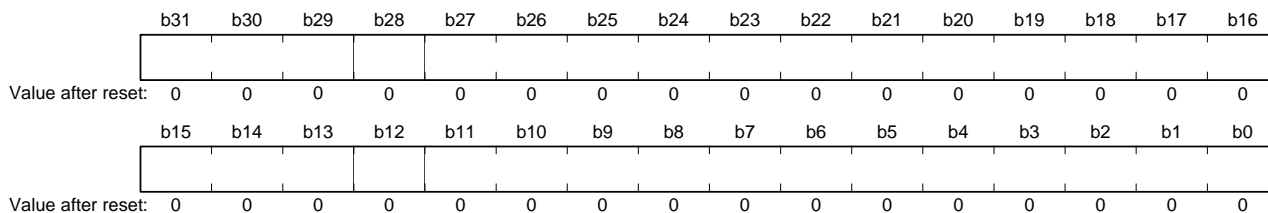


Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the number of frames received containing residual bits.	R/W

RFCR is a counter that indicates the number of frames received containing residual bits (less than an 8-bit unit). When the value in RFCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to RFCR with any value.

26.2.18 Multicast Address Frame Receive Counter Register (MAFCR)

Address: 000C 01F8h

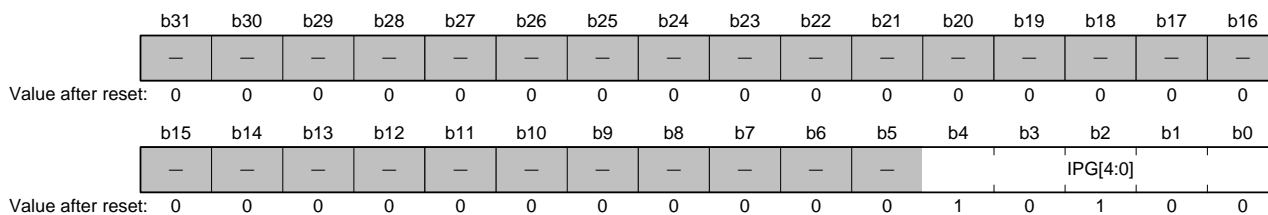


Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the number of multicast frames received.	R/W

MAFCR is a counter that indicates the number of frames received with a specified multicast address. When the value in MAFCR reaches FFFFFFFFh, the counter stops incrementing. The counter value is cleared to 0 by a write to MAFCR with any value.

26.2.19 IPG Register (IPGR)

Address: 000C 0150h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IPG[4:0]	Inter Packet Gap	00h: 16-bit time 01h: 20-bit time : : 14h: 96-bit time (initial value) : : 1Fh: 140-bit time	R/W
b31 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

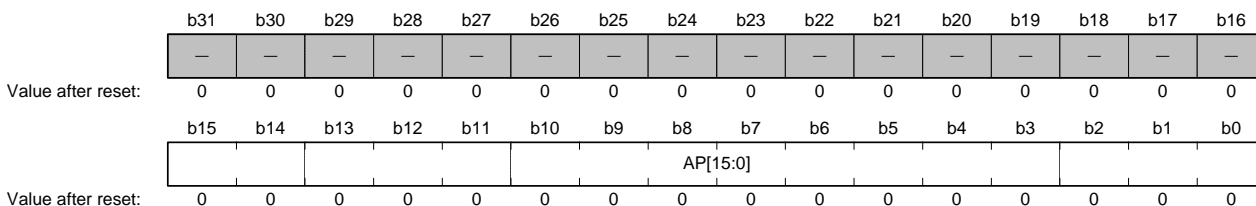
IPGR sets the IPG (Inter Packet Gap) value. This setting must not be changed while the transmitting and receiving functions of ECMR are enabled. (For details, refer to section 26.3.6, Operation by IPG Setting.)

IPG[4:0] Bits (Inter Packet Gap)

These bits specify the IPG value in 4-bit time units.

26.2.20 Automatic PAUSE Frame Register (APR)

Address: 000C 0154h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	AP[15:0]	Automatic PAUSE	These bits set the TIME parameter value of an automatic PAUSE frame.	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

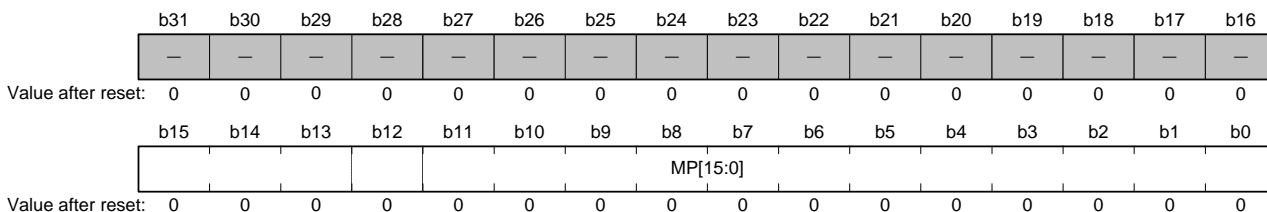
APR specifies the TIME parameter value of an automatic PAUSE frame. When an automatic PAUSE frame is transmitted, the value set in APR is used as the TIME parameter of the PAUSE frame. This setting must not be changed while the transmitting and receiving functions of APR are enabled.

AP[15:0] Bits (Automatic PAUSE)

These bits set the TIME parameter value of an automatic PAUSE frame. One bit is equivalent to 512 bit-time.

26.2.21 Manual PAUSE Frame Register (MPR)

Address: 000C 0158h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MP[15:0]	Manual PAUSE	These bits set the TIME parameter value of a manual PAUSE frame.	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

MPR specifies the TIME parameter value of a manual PAUSE frame. When a manual PAUSE frame is transmitted, the value set in MPR is used as the TIME parameter of the PAUSE frame.

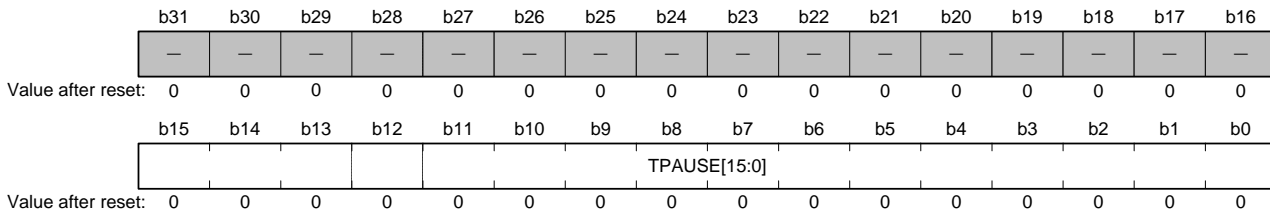
Write to the MPR register while transmission is enabled.

MP[15:0] Bits (Manual PAUSE)

These bits set the TIME parameter value of a manual PAUSE frame. One bit is equivalent to 512 bit-time. The read value is undefined.

26.2.22 Automatic PAUSE Frame Retransmit Count Register (TPAUSER)

Address: 000C 0164h

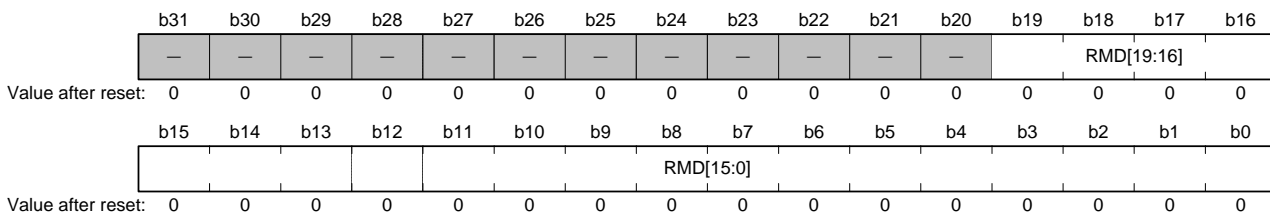


Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TPAUSE[15:0]	Upper Limit for Automatic PAUSE Frame Retransmission	0000h: Retransmit count is unlimited 0001h: Retransmit count is 1 : : FFFFh: Retransmit count is 65,535	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

TPAUSER specifies the upper limit for the number of times to retransmit an automatic PAUSE frame. The settings in TPAUSER must not be changed while the transmitting function is enabled.

26.2.23 Random Number Generation Counter Upper Limit Setting Register (RDMLR)

Address: 000C 0140h



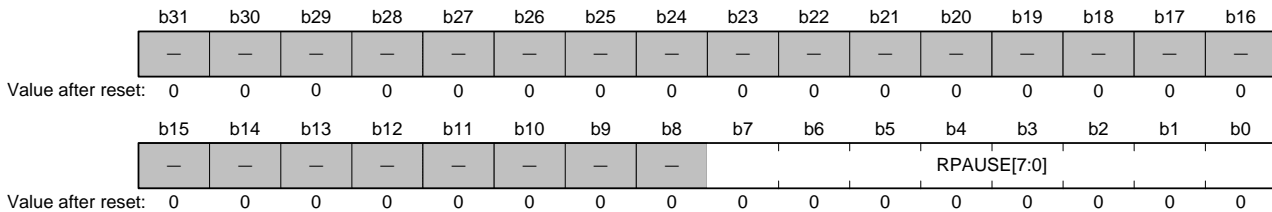
Bit	Symbol	Bit Name	Description	R/W
b19 to b0	RMD[19:0]	Upper Limit for Counter Used in Random Number Generation Block	00000h: Setting for normal operation 00001h to FFFFh: Upper limit for the counter	R/W
b31 to b20	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note: The operation of the random number generation block depends on the setting in RDMLR. Accordingly, special attention should be paid when setting a value other than 0.

RDMLR specifies the upper limit for the counter used in the random number generation block. This setting must not be changed while the transmitting and receiving functions of RDMLR are enabled.

26.2.24 PAUSE Frame Receive Counter Register (RFCF)

Address: 000C 0160h

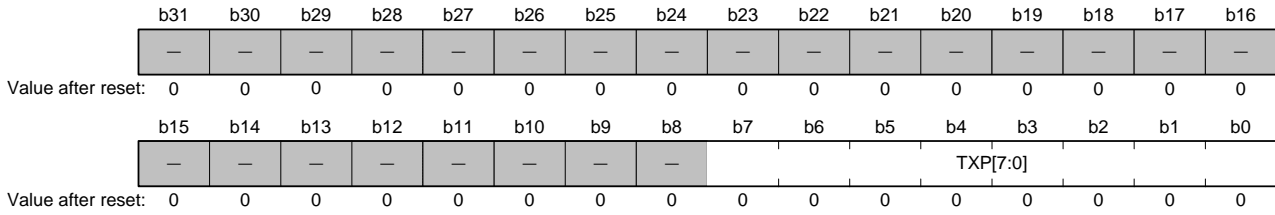


Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RPAUSE[7:0]	PAUSE Frame Receive Count	Receive counter	R
b31 to b8	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RFCF is a counter that indicates the number of times a PAUSE frame was received.

26.2.25 PAUSE Frame Retransmit Counter Register (TPAUSECR)

Address: 000C 0168h

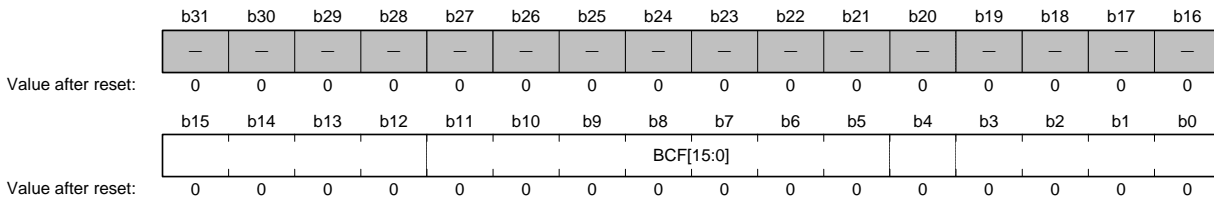


Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TXP[7:0]	PAUSE Frame Retransmit Count	Number of times a PAUSE frame was retransmitted	R
b31 to b8	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

TPAUSECR is a counter that indicates the number of times a PAUSE frame was retransmitted.

26.2.26 Broadcast Frame Receive Count Setting Register (BCFRR)

Address: 000C 016Ch



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	BCF[15:0]	Receive Count for Continuous Broadcast Frames	0000h: No limitation for receive count 0001h: 1 frame can be received : : FFFFh: 65,535 frames can be received	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

BCFRR specifies the number of Broadcast frames that can be received continuously. This setting must not be changed while the transmitting and receiving functions of BCFRR are enabled.

BCF[15:0] Bits (Receive Count for Continuous Broadcast Frames)

If the destination address (DA) can receive a frame with a Broadcast address up to the number of times set in these bits and frames have been received for more times than the specified count, the excess Broadcast frames are discarded.

26.3 Operation

The following gives an outline of the ETHERC operations.

The ETHERC supports flow control functions conforming to IEEE802.3x, and can transmit and receive PAUSE frames used for the control.

26.3.1 Transmission

The ETHERC transmitter assembles transmit data into a frame and outputs it to the MII/RMII when there is a transmit request from the transmit EDMAC. The data transmitted via the MII/RMII is transmitted to the lines by the PHY-LSI. Figure 26.2 shows the state transitions in the ETHERC transmitter.

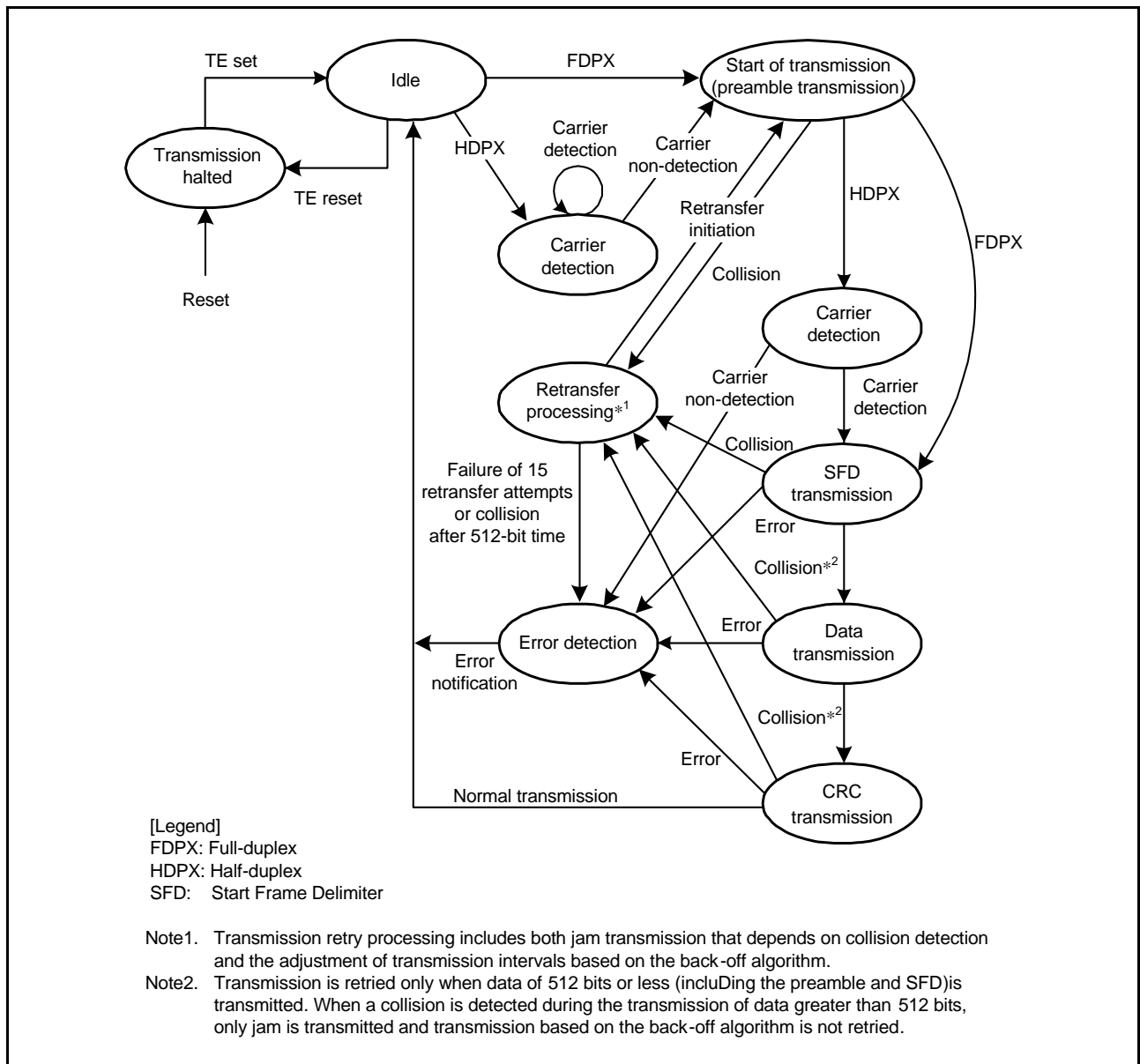


Figure 26.2 ETHERC Transmitter State Transitions

1. When the transmit enable (ECMR.TE) bit is set, the transmitter enters the transmit idle state.
2. When a transmit request is issued by the transmit EDMAC, the ETHERC sends the preamble to the MII/RMII after carrier detection and a transmission delay equivalent to the frame interval time. If full-duplex transfer is selected, which does not require carrier detection, the preamble is sent as soon as a transmit request is issued by the transmit EDMAC.
3. The transmitter sends the SFD, data, and CRC sequentially. At the end of transmission, the transmit EDMAC generates a transmission complete interrupt (TC). If a collision or the carrier-not-detected state occurs during data transmission, it is reported as an interrupt.
4. After waiting for the frame interval time, the transmitter enters the idle state, and if there is more transmit data, continues transmission.

26.3.2 Reception

The ETHERC receiver separates the frame from the MII/RMII into preamble, SFD, data, and CRC, and the fields from DA (destination address) to the CRC data are transferred to the receive EDMAC. Figure 26.3 shows the state transitions of the ETHERC receiver.

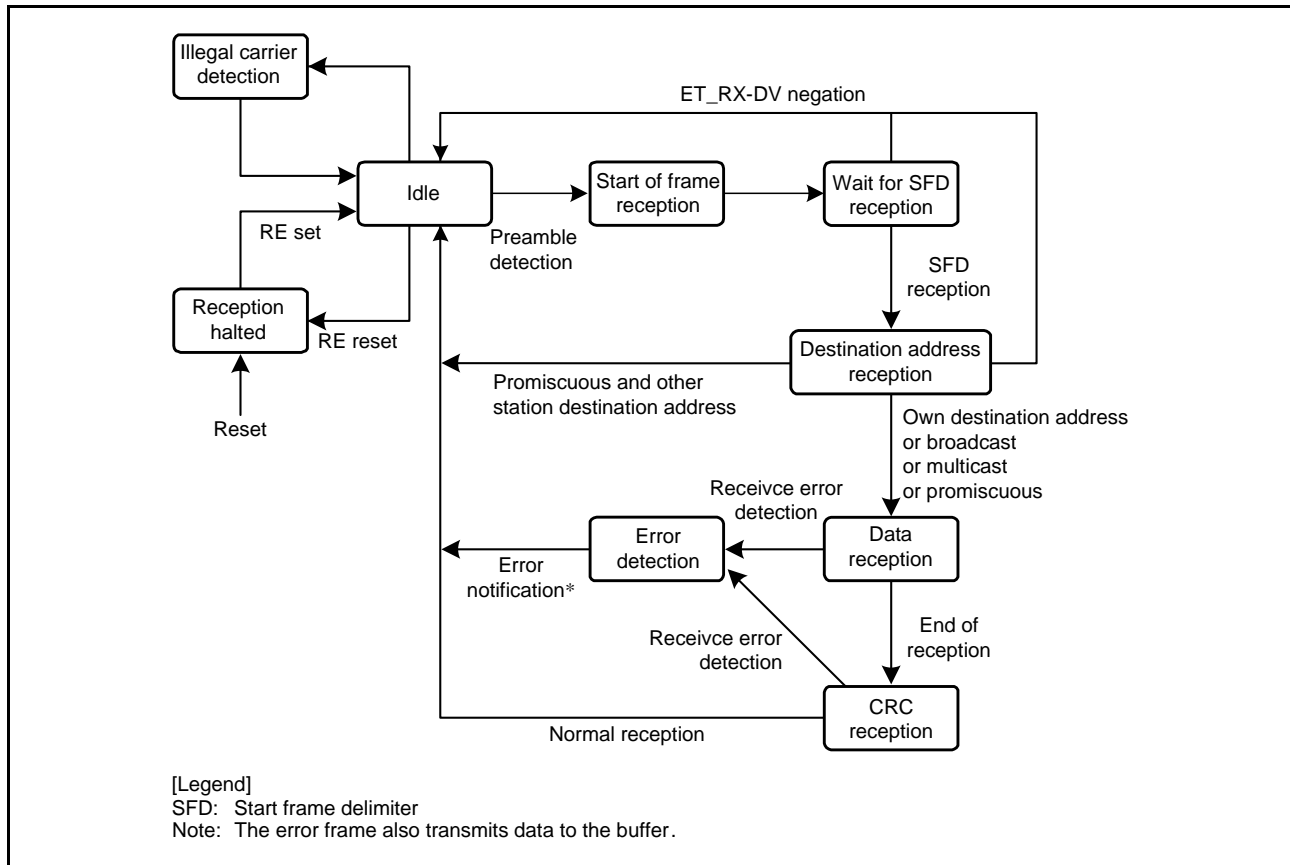


Figure 26.3 ETHERC Receiver State Transitions

1. When the receive enable (ECMR.RE) bit is set, the receiver enters the receive idle state.
2. Upon detecting an SFD (start frame delimiter) after a receive packet preamble, the receiver starts receive processing. It discards a frame with an invalid pattern.
3. In normal mode, if the destination address of the frame matches the RX62N address, or if the broadcast or multicast frame type is specified, the receiver starts data reception. In promiscuous mode, the receiver starts reception for any type of frame.
4. After data reception from the MII/RMII, the receiver carries out a CRC check in the frame data field. The result is indicated as a status bit in the descriptor after the frame data has been written to memory. The receiver reports an error status in the case of an abnormality.
5. After one frame has been received, if the receive enable bit is set (ECMR.RE = 1) in the ETHERC mode register, the receiver prepares to receive the next frame.

26.3.3 Frame Timing

26.3.3.1 MII Frame Timing

The MII Frame timing is shown in Figure 26.4 to Figure 26.9.

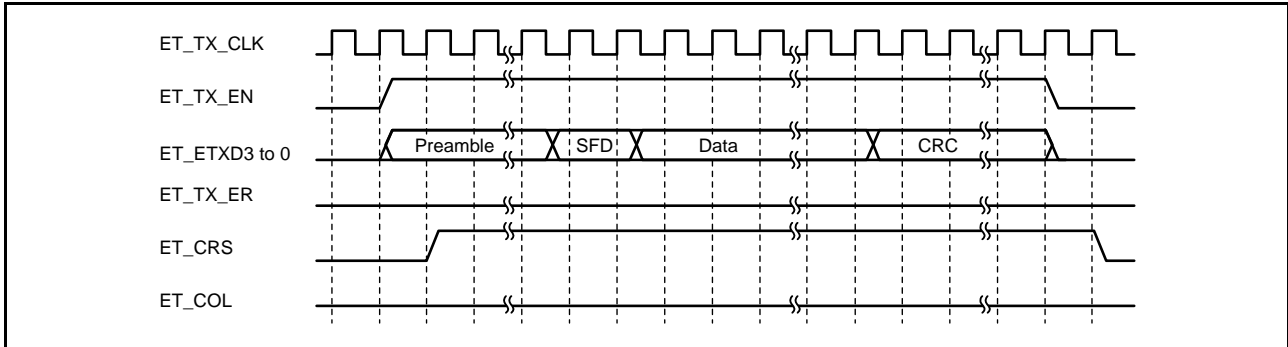


Figure 26.4 MII Frame Transmit Timing (Normal Transmission)

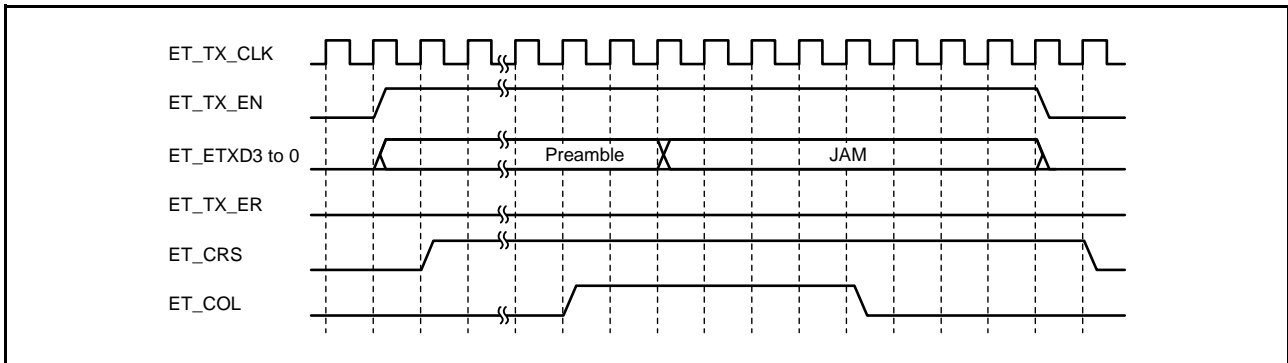


Figure 26.5 MII Frame Transmit Timing (Collision)

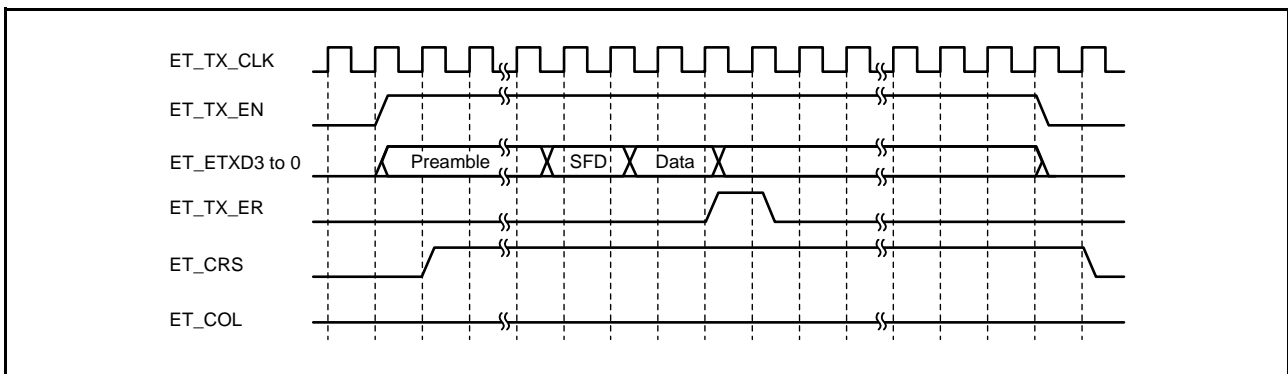


Figure 26.6 MII Frame Transmit Timing (Transmit Error)

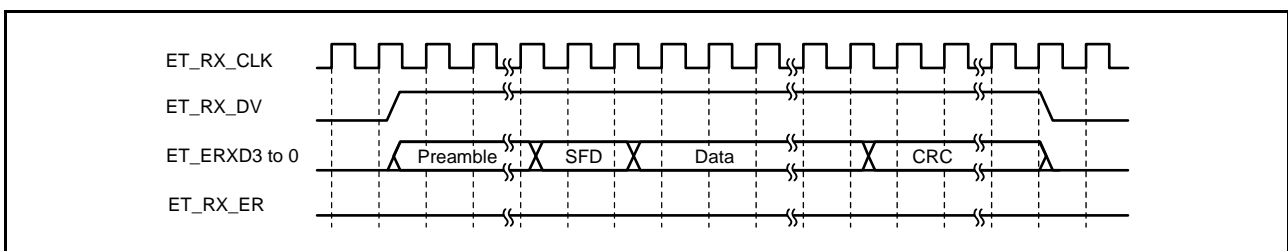


Figure 26.7 MII Frame Receive Timing (Normal Reception)

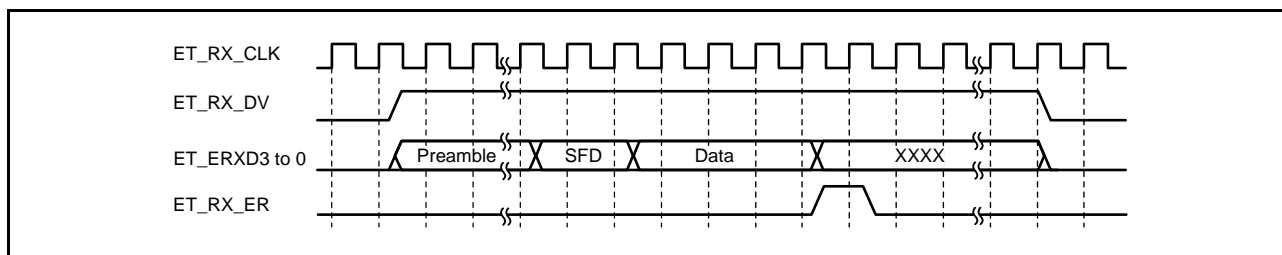


Figure 26.8 MII Frame Receive Timing (Reception Error (1): Receive Error Notification)

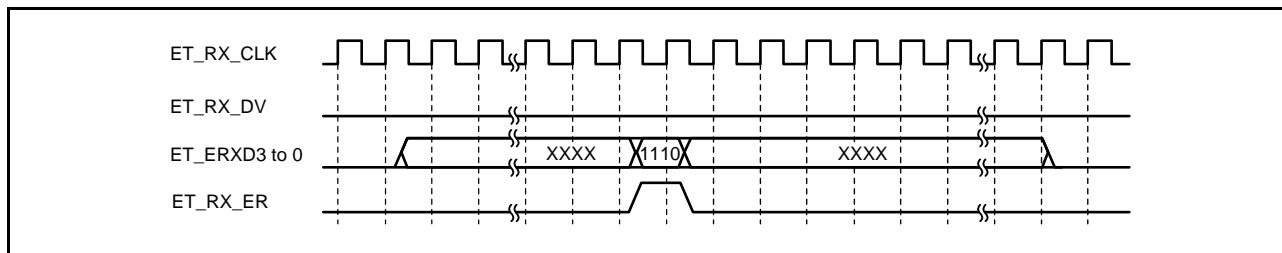


Figure 26.9 MII Frame Receive Timing (Reception Error (2): Carrier Error Notification)

26.3.3.2 RMII Frame Timing

The RMII Frame timing is shown in Figure 26.10 to Figure 26.12.

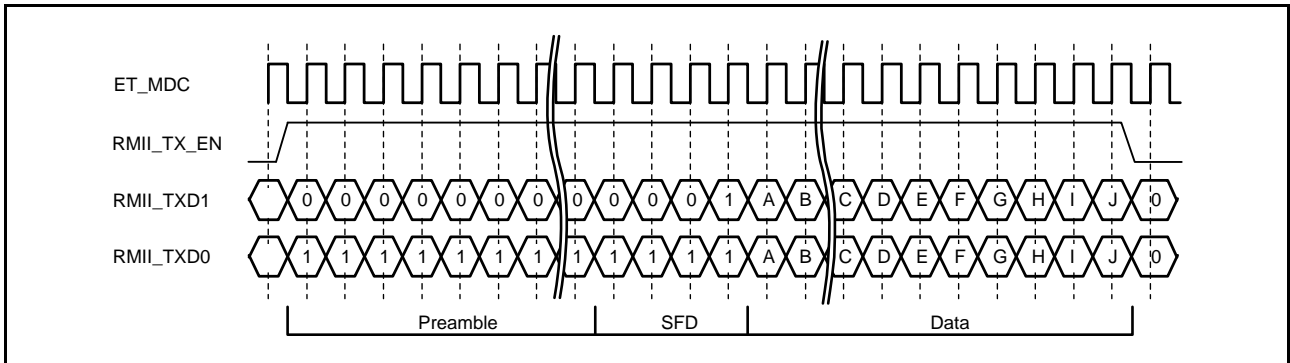


Figure 26.10 RMII Frame Transmit Timing (Normal Transmission)

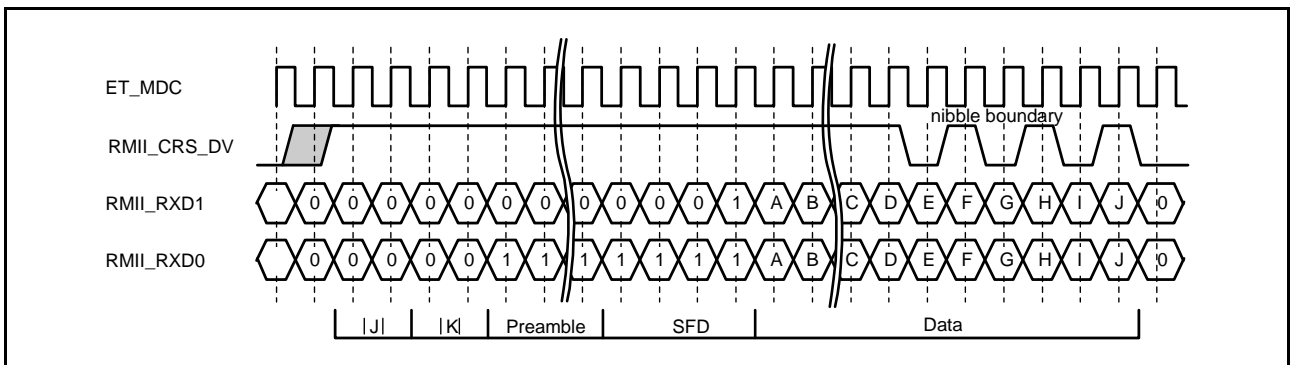


Figure 26.11 RMII Frame Receive Timing (Normal Reception)

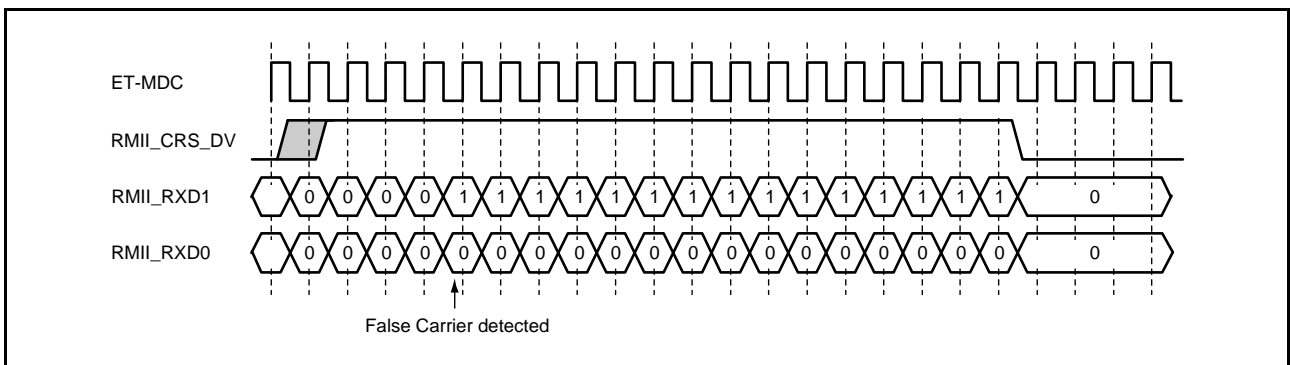


Figure 26.12 RMII Frame Receive Timing (False Carrier Detected)

26.3.4 Accessing MII/RMII Registers

The MII/RMII registers in the PHY-LSI are accessed via the PHY interface register (PIR) in the RX62N. Connection is made as a serial interface in accordance with the MII/RMII frame format.

26.3.4.1 MII/RMII Management Frame Format

The format of an MII/RMII management frame is shown in Figure 26.13. To access an MII/RMII register, a management frame should be implemented by the program in accordance with the procedures shown in section 26.3.4.2, MII/RMII Register Access Procedure.

Access Type	MII/RMII Management Frame							
Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
Number of bits	32	2	2	5	5	2	16	—
Read	1..1	01	10	00001	RRRRR	Z0	D..D	—
Write	1..1	01	01	00001	RRRRR	10	D..D	X

[Legend]
 PRE: 32 consecutive 1b
 ST: Write 01b to indicate the start of frame
 OP: Write the code indicating access type
 PHYAD: Write 0001b if the PHY-LSI address is 1 (sequential write starting with the MSB). This value depends on the PHY-LSI address.
 REGAD: Write 0001b if the register address is 1 (sequential write starting with the MSB). This value depends on the PHY-LSI register address.
 TA: Time for switching data transmission source on MII/RMII interface
 (a) For write: Write 10b
 (b) For read: Bus release (notation: Z0) performed
 DATA: 16-bit data. Sequentially write or read from MSB
 (a) For write: Write 16-bit data
 (b) For read: 16-bit data read
 IDLE: Wait time until next MII management format input
 (a) For write: Independent bus release (notation: X) performed
 (b) For read: Bus already released in TA; control unnecessary

Figure 26.13 MII/RMII Management Frame Format

26.3.4.2 MII/RMII Register Access Procedure

The program should access MII/RMII registers via the PHY interface register (PIR). Access is implemented by a combination of 1-bit-unit data write, 1-bit-unit data read, bus release, and independent bus release. Figure 26.14 to Figure 26.17 show the MII/RMII register access timing. The timing will differ depending on the PHY-LSI type.

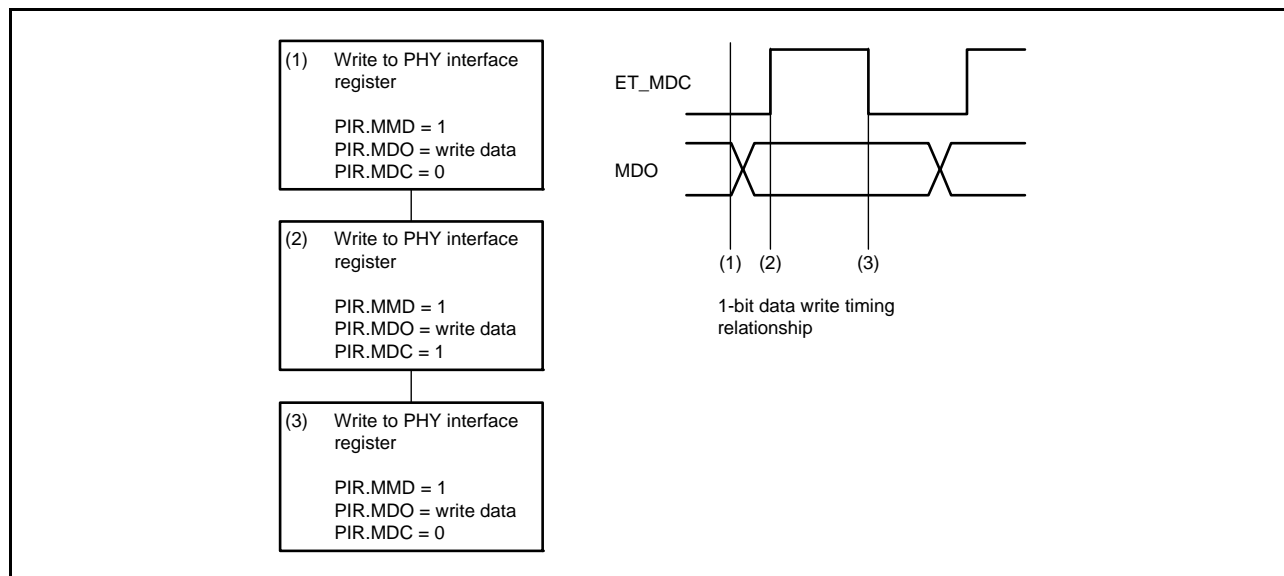


Figure 26.14 Flowchart of 1-Bit Data Write

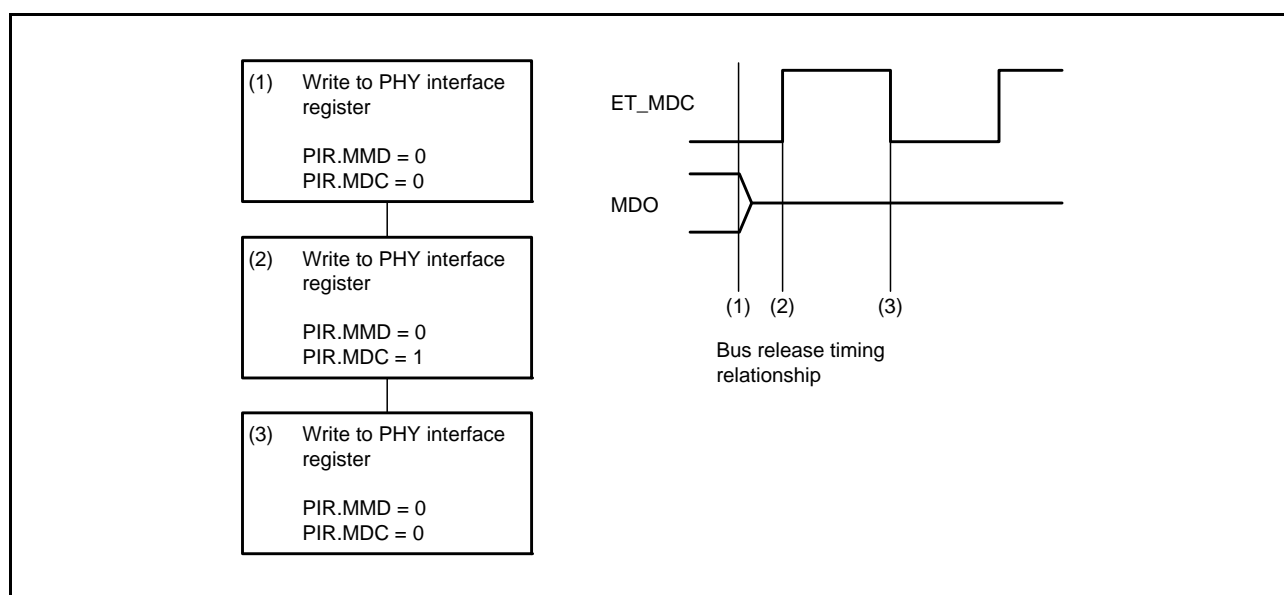


Figure 26.15 Flowchart of Bus Release (TA in Read in Figure 26.13)

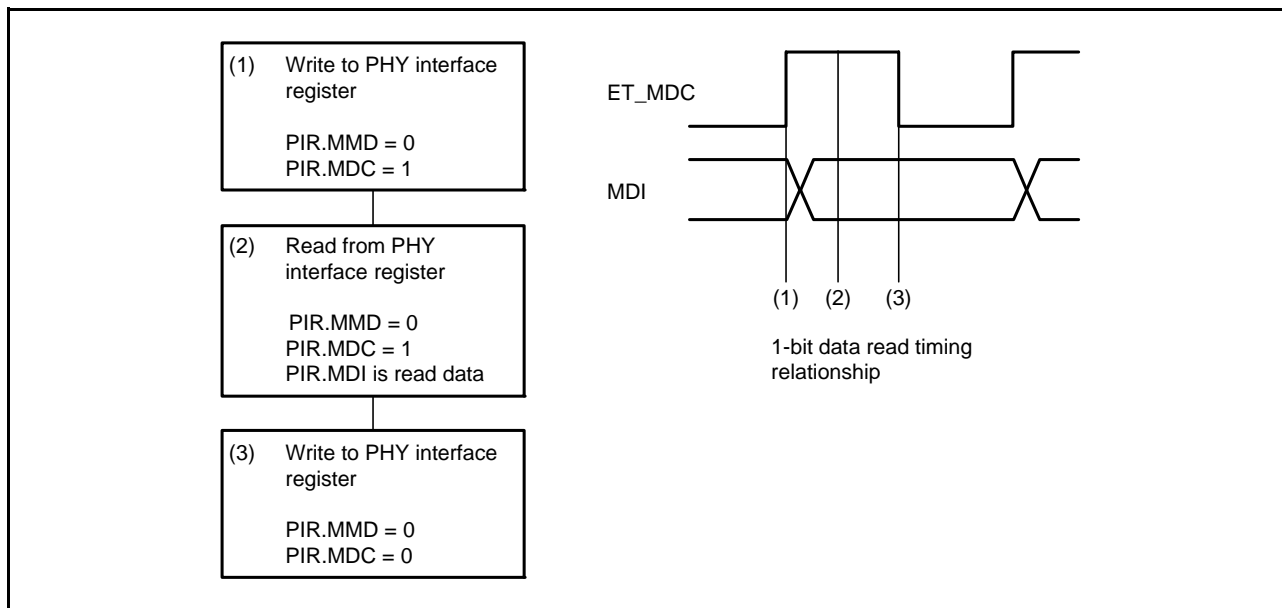


Figure 26.16 Flowchart of 1-Bit Data Read

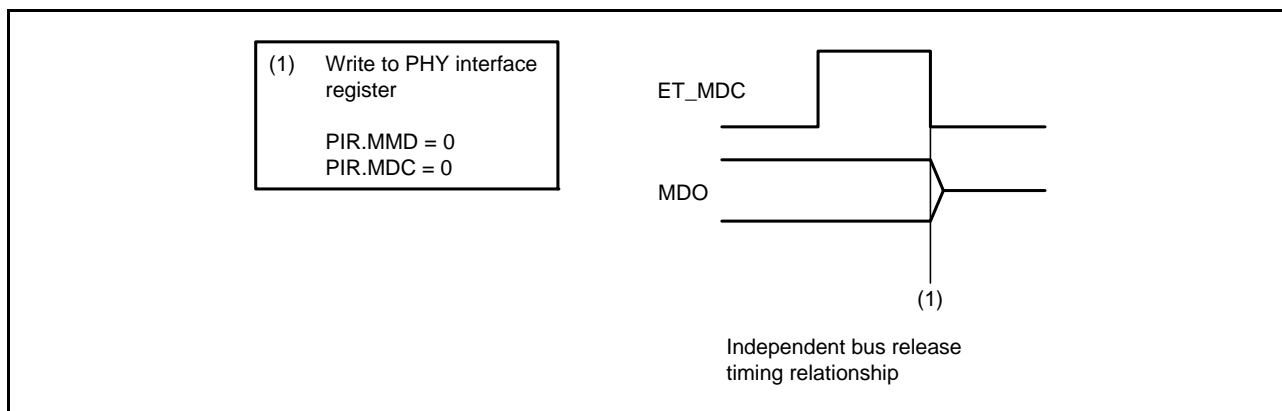


Figure 26.17 Flowchart of Independent Bus Release (IDLE in Write in Figure 26.13)

26.3.5 Magic Packet™ Detection

The ETHERC has a Magic Packet™ detection function. This function provides a Wake-On-LAN (WOL) facility that activates various peripheral devices connected to a LAN from the host device or other source. This makes it possible to construct a system in which a peripheral device receives a Magic Packet™ sent from the host device or other source, and activates itself. When the Magic Packet™ is detected, data is stored in the receive FIFO through the broadcast packet that has been received and the ETHERC is notified of the receiving status. To return to normal operation from the interrupt processing, initialize the ETHERC and EDMAC by using the software reset (SWR) bit in the EDMAC mode register (EDMR) in the EDMAC.

With a Magic Packet™, reception is performed regardless of the destination address. As a result, this function becomes valid and the ET_WOL pin is enabled only when a match occurs with the destination address specified by the format in the Magic Packet™. Further information on Magic Packets™ can be found in the technical documentation published by Advanced Micro Devices, Inc.

The procedure for using the WOL function with the RX62N is as follows.

1. Disable interrupt source output by means of the interrupt enable/mask registers.
2. Set the Magic Packet™ detection enable bit (MPDE) in the ETHERC mode register (ECMR).
3. Set the Magic Packet™ detection interrupt enable bit (MPDIP) to the enabled state in the ETHERC interrupt permission register (ECSIPR).
4. If necessary, set the CPU operating mode to sleep mode or set peripheral modules to module standby mode.
5. When a Magic Packet™ is detected, an interrupt is sent to the CPU. The ET_WOL pin notifies peripheral LSIs that the Magic Packet™ has been detected.

26.3.6 Operation by IPG Setting

The ETHERC has a function to change the non-transmission period, IPG (Inter Packet Gap), between transmit frames. By changing the value in the IPG setting register (IPGR), the transmission efficiency can be raised or lowered from the standard value. IPG settings are prescribed in the IEEE802.3 standard. When changing settings, adequately check that each device can operate smoothly on the same network.

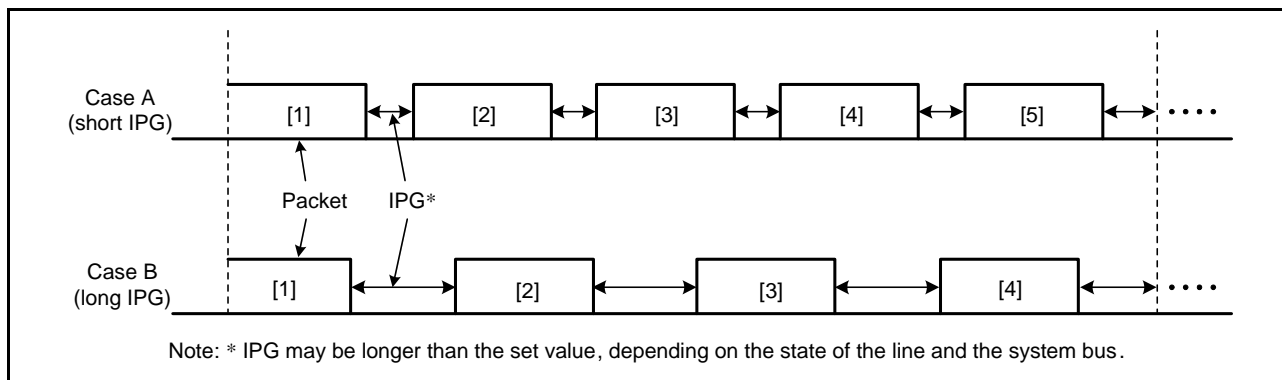


Figure 26.18 Relationship between IPG Setting and Transmission Efficiency

26.3.7 Flow Control

The ETHERC supports flow control functions conforming to IEEE802.3x for full-duplex operation. The flow control can be applied to both receive and transmit operations. For flow control, the following procedures can be used to transmit PAUSE frames.

26.3.7.1 Automatic PAUSE Frame Transmission

For receive frames, PAUSE frames are automatically transmitted when the amount of data written to the receive FIFO (in EDMAC) reaches the value set in the flow control start FIFO threshold setting register (FCFTR) of the EDMAC. The TIME parameter included in the PAUSE frame is set by the automatic PAUSE frame register (APR). The automatic PAUSE frame transmission is repeated until the amount of data in the receive FIFO becomes less than the value set in FCFTR as the receive data is read from the FIFO. Using the automatic PAUSE frame retransmit count register (TPAUSER), the upper limit of the PAUSE frame retransmission count can also be set. In this case, PAUSE frame transmission is repeated until the amount of receive FIFO data becomes less than the FCFTR value or the number of transmits reaches the TPAUSER value.

The automatic PAUSE frame transmission is enabled when the operating mode bit for transmitting port flow control (TXF) in the ETHERC mode register (ECMR) is 1.

26.3.7.2 Manual PAUSE Frame Transmission

PAUSE frames are transmitted under software control. When the Timer value is written to the manual PAUSE frame register (MPR), manual PAUSE frame transmission is started. With this method, PAUSE frame transmission is carried out only once.

26.3.7.3 PAUSE Frame Reception

After a PAUSE frame is received, the next frame is not transmitted until the time indicated by the Timer value elapses. However, the transmission of the current frame is continued. PAUSE frame reception is enabled when the operating mode bit for receiving port flow control (RXF) in the ETHERC mode register (ECMR) is set to 1.

26.4 Connection to PHY-LSI

Figure 26.19 and Figure 26.20 show examples of connection to a PHY-LSI.

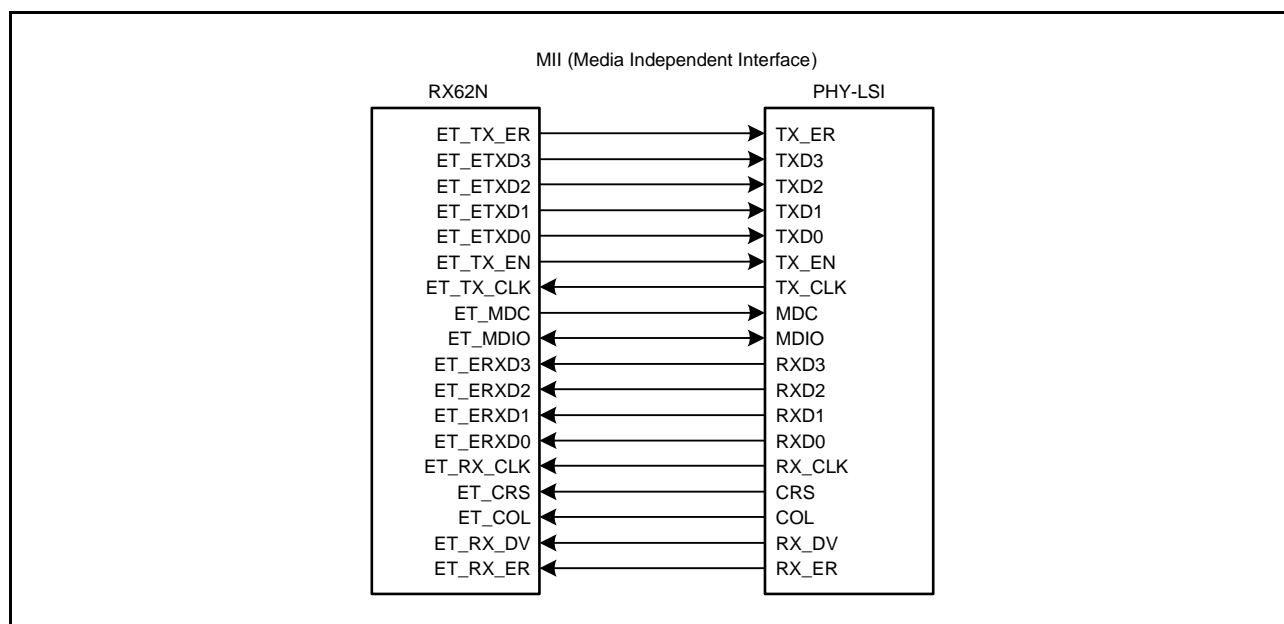


Figure 26.19 Example of Connection to PHY-LSI (MII)

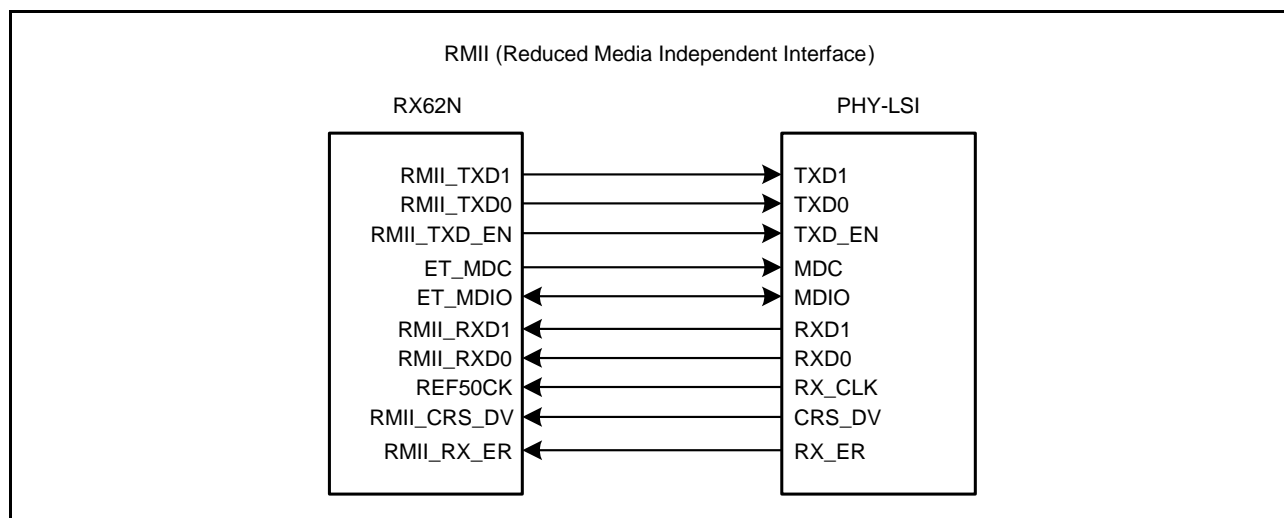


Figure 26.20 Example of Connection to PHY-LSI (RMII)

26.5 Usage Notes

Note the following when using the ETHERC.

26.5.1 Conditions for Setting LCHNG Bit

The ECSR.LCHNG bit may be set even when the input level on the ET_LINKSTA pin has not changed. It may be set when the ET_LINKSTA pin is enabled by the IOPORT.PFENET bit or when a high is applied to the ET_LINKSTA pin while the ETHERC/EDMAC is released from the software reset state by the SWR bit in EDMR in the EDMAC. This is because the ET_LINKSTA signal is internally fixed low regardless of the external pin level when the ET_LINKSTA pin is not enabled by the IOPORT.PFENET bit or while the ETHERC/EDMAC is in the software reset state.

In order not to request the LINK signal change interrupt accidentally, clear the ECSR.LCHNG bit before setting the ECSIPR.LCHNGIP.

26.5.2 RMII_RX_ER Pin Input when RMII is Selected

When RMII is selected, if the receive error signal from the PHY is only one cycle of the RMII 50-MHz reference clock, it is not recognized as an error signal.

27. Ethernet Controller Direct Memory Access Controller (EDMAC)

27.1 Overview

The RX62N Group has an on-chip direct memory access controller (EDMAC) directly connected to the Ethernet controller (ETHERC). The EDMAC controls the most part of the buffer management by using descriptors. This reduces the load on the CPU, thus enabling efficient data transmission and reception.

Table 27.1 shows the EDMAC specifications, and Figure 27.1 shows the configuration of the EDMAC, and the descriptors and transmit/receive buffers in memory.

Table 27.1 Specifications of EDMAC

Item	Description
Data transmission and reception	<ul style="list-style-type: none"> • Descriptor management system • Supports single-frame/multi-buffer operation
Function	<ul style="list-style-type: none"> • Achieves efficient system bus utilization through the use of DMA block transfer (32-byte units) • Transmit/receive frame status information is indicated in descriptors • Padding can be inserted in receive data

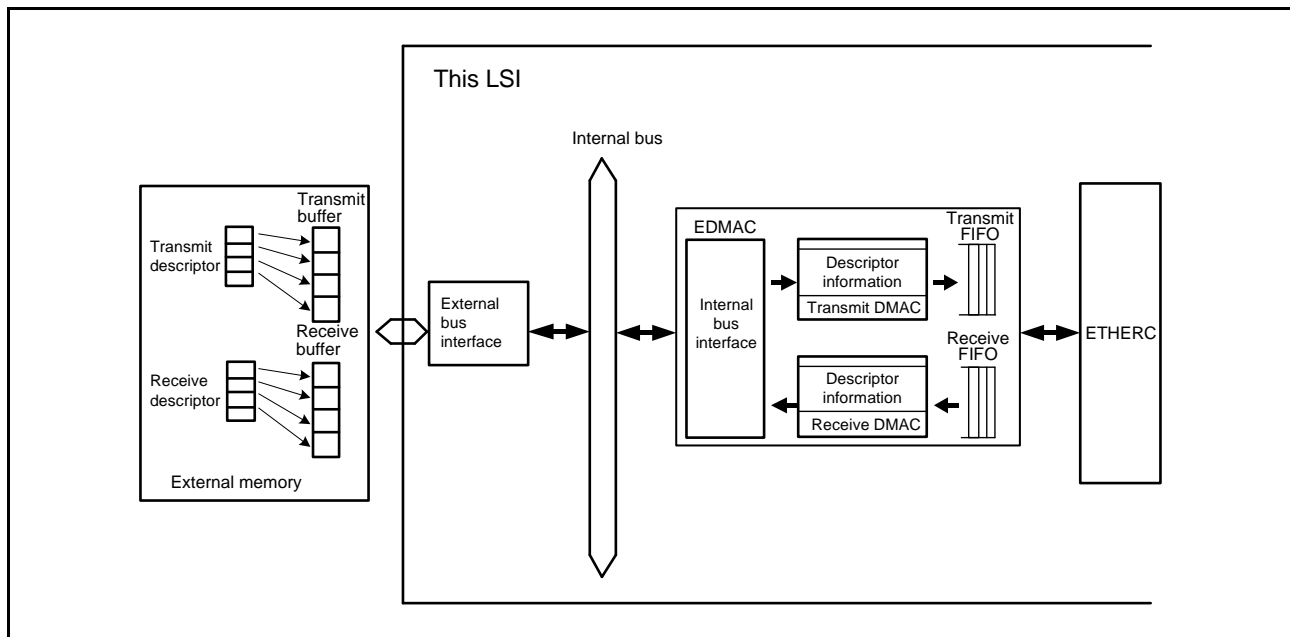


Figure 27.1 Configuration of EDMAC, and Descriptors and Buffers

27.2 Register Descriptions

Table 27.2 lists the registers of the EDMAC.

Table 27.2 Registers of EDMAC

Register Name	Symbol	Value after Reset	Address	Access Size
EDMAC mode register	EDMR	0000 0000h	000C 0000h	32
EDMAC transmit request register	EDTRR	0000 0000h	000C 0008h	32
EDMAC receive request register	EDRRR	0000 0000h	000C 0010h	32
Transmit descriptor list start address register	TDLAR	0000 0000h	000C 0018h	32
Receive descriptor list start address register	RDLAR	0000 0000h	000C 0020h	32
ETHERC/EDMAC status register	EESR	0000 0000h	000C 0028h	32
ETHERC/EDMAC status interrupt permission register	EESIPR	0000 0000h	000C 0030h	32
Transmit/receive status copy enable register	TRSCER	0000 0000h	000C 0038h	32
Receive missed-frame counter register	RMFCR	0000 0000h	000C 0040h	32
Transmit FIFO threshold register	TFTR	0000 0000h	000C 0048h	32
FIFO depth register	FDR	0000 0000h	000C 0050h	32
Receiving method control register	RMCR	0000 0000h	000C 0058h	32
Transmit FIFO underrun counter	TFUCR	0000 0000h	000C 0064h	32
Receive FIFO overflow counter	RFOCR	0000 0000h	000C 0068h	32
Receive buffer write address register	RBWAR	0000 0000h	000C 00C8h	32
Receive descriptor fetch address register	RDFAR	0000 0000h	000C 00CCh	32
Transmit buffer read address register	TBRAR	0000 0000h	000C 00D4h	32
Transmit descriptor fetch address register	TDFAR	0000 0000h	000C 00D8h	32
Flow control start FIFO threshold setting register	FCFTR	0007 0007h	000C 0070h	32
Receive data padding insert register	RPADIR	0000 0000h	000C 0078h	32
Transmit interrupt setting register	TRIMD	0000 0000h	000C 007Ch	32
Independent output signal setting register	IOSR	0000 0000h	000C 006Ch	32

27.2.1 EDMAC Mode Register (EDMR)

Address: 000C 0000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	DE	DL[1:0]	—	—	—	—	SWR
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	SWR	Software Reset	[Writing] 0: Disabled 1: ETHERC and EDMAC are reset*1	R/W
b3 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b5, b4	DL[1:0]	Transmit/Receive Descriptor Length	b5 b4 0 0: 16 bytes 0 1: 32 bytes 1 0: 64 bytes 1 1: 16 bytes	R/W
b6	DE	Big/Little Endian Mode*2	0: Big endian mode (longword access) 1: Little endian mode (longword access)	R/W
b31 to b7	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. TDLAR, RMFCR, TFUCR, and RFOCR are not reset.

Note 2. The setting applies to transmit and receive data. The setting does not apply to transmit/receive descriptors or registers (only big endian mode is available).

EDMR specifies EDMAC operating mode.

EDMR should usually be set at initialization after a reset. If the ETHERC and EDMAC are initialized with EDMR during data transmission, abnormal data may be transmitted on the line.

It is prohibited to modify the operating mode while transmission or reception function is enabled. Before modifying the operating mode, the ETHERC and EDMAC should be initialized by setting the SWR bit.

Note that complete initialization of the ETHERC and EDMAC modules takes 64 cycles of the internal bus clock of the EDMAC. Therefore, access to registers in the ETHERC or EDMAC should only proceed after that period has elapsed.

27.2.2 EDMAC Transmit Request Register (EDTTR)

Address: 000C 0008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TR
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	TR	Transmit Request	0: Transmission-halted state Writing 0 does not stop transmission. Termination of transmission is controlled by the active bit of the transmit descriptor. 1: Transmission start The EDMAC starts reading the target descriptor and sends a frame whose transmission active bit is set to 1.	R/W
b31 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

EDTTR issues transmit directives to the EDMAC.

After having transmitted one frame, the EDMAC reads the next descriptor. If the transmit descriptor active bit in this descriptor is set (active), the EDMAC continues transmission. Otherwise, the EDMAC clears the TR bit and stops the transmit DMAC operation.

27.2.3 EDMAC Receive Request Register (EDRRR)

Address: 000C 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	RR
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	RR	Receive Request	0: Receiving function is disabled* 1: Receive descriptor is read, and the EDMAC becomes ready to receive	R/W
b31 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note : * If the receiving function is disabled during frame reception, write-back is not performed successfully to the receive descriptor. Following pointers to read a receive descriptor become abnormal and the EDMAC cannot operate successfully. In this case, to make EDMAC reception enabled again, execute a software reset by the EDMR.SWR bit.
To disable the EDMAC receiving function without executing a software reset, specify the ECMR.RE bit in the ETHERC. Next, after the EDMAC has completed the reception and write-back to the receive descriptor has been confirmed, disable the receiving function using EDRRR.

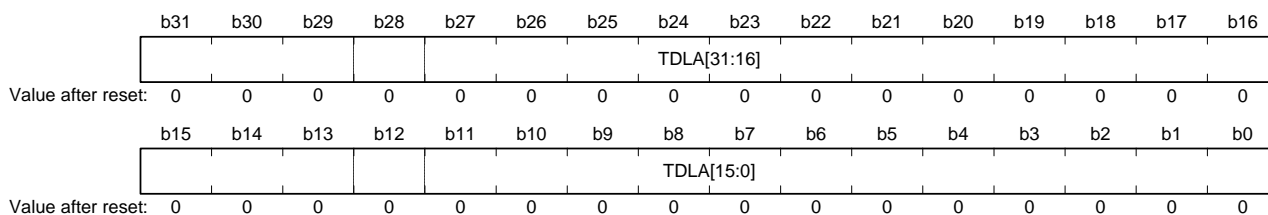
EDRRR issues receive directives to the EDMAC.

After writing 1 to the RR bit in this register, the EDMAC reads the receive descriptor. If the receive descriptor active bit of this receive descriptor is set to 1 (active), the EDMAC becomes ready for a receive request from the ETHERC.

After data has been received for the receive buffer size, the EDMAC reads the next receive descriptor and becomes ready for frame reception. If the receive descriptor active bit of that receive descriptor is set to 0 (inactive), the EDMAC clears the RR bit and stops receive DMAC operation.

27.2.4 Transmit Descriptor List Start Address Register (TDLAR)

Address: 000C 0018h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TDLA[31:0]	Transmit Descriptor Start Address	16-byte boundary: TDLA[3:0] = 0000 32-byte boundary: TDLA[4:0] = 00000 64-byte boundary: TDLA[5:0] = 000000	R/W

TDLAR specifies the start address of the transmit descriptor list.

Descriptors have a boundary configuration in accordance with the descriptor length indicated by the EDMR.DL[1:0] bits.

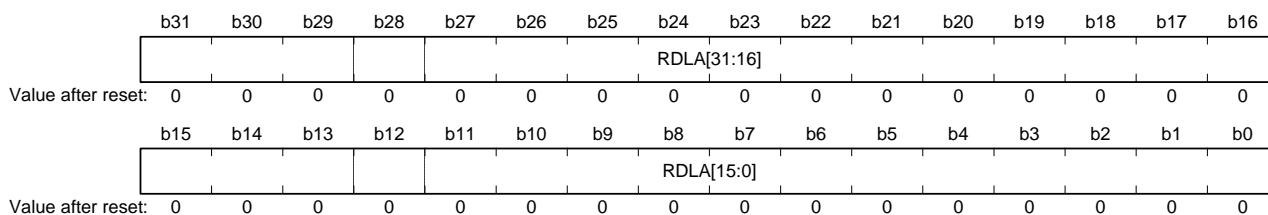
TDLAR must not be modified during transmission. Modifications to TDLAR should only be made in the transmission-halted state specified by the EDTRR.TR bit (= 0).

TDLA[31:0] Bits (Transmit Descriptor Start Address)

The lower bits should be set according to the specified descriptor length.

27.2.5 Receive Descriptor List Start Address Register (RDLAR)

Address: 000C 0020h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RDLA[31:0]	Receive Descriptor Start Address	16-byte boundary: RDLA[3:0] = 0000 32-byte boundary: RDLA[4:0] = 00000 64-byte boundary: RDLA[5:0] = 000000	R/W

RDLAR specifies the start address of the receive descriptor list.

Descriptors have a boundary configuration in accordance with the descriptor length indicated by the EDMR.DL[1:0] bits.

RDLAR must not be modified during reception. Modifications to RDLAR should only be made while reception is disabled by the EDRRR.RR bit (= 0).

RDLA[31:0] Bits (Receive Descriptor Start Address)

The lower bits should be set according to the specified descriptor length.

27.2.6 ETHERC/EDMAC Status Register (EESR)

Address: 000C 0028h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	TWB	—	—	—	TABT	RABT	RFCOF	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
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
—	—	—	—	CND	DLC	CD	TRO	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
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	CERF	CRC Error on Received Frame	0: CRC error has not been detected 1: CRC error has been detected	R/W
b1	PRE	PHY-LSI Receive Error	0: PHY-LSI receive error has not been detected 1: PHY-LSI receive error has been detected	R/W
b2	RTSF	Receive Too-Short Frame	0: Too-short frame has not been received 1: Too-short frame has been received	R/W
b3	RTLF	Receive Too-Long Frame	0: Too-long frame has not been received 1: Too-long frame has been received	R/W
b4	RRF	Receive Residual-Bit Frame	0: Residual-bit frame has not been received 1: Residual-bit frame has been received	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	RMAF	Receive Multicast Address Frame	0: Multicast address frame has not been received 1: Multicast address frame has been received	R/W
b8	TRO	Transmit Retry Over	0: Transmit retry-over condition has not been detected 1: Transmit retry-over condition has been detected	R/W
b9	CD	Delayed Collision Detect	0: Delayed collision has not been detected 1: Delayed collision has been detected	R/W
b10	DLC	Detect Loss of Carrier	0: Loss of carrier has not been detected 1: Loss of carrier has been detected	R/W
b11	CND	Carrier Not Detect	0: A carrier is detected when transmission starts 1: A carrier is not detected	R/W
b15 to b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b16	RFOF	Receive FIFO Overflow	0: Overflow has not occurred 1: Overflow has occurred	R/W
b17	RDE	Receive Descriptor Empty	0: Receive descriptor active bit RD0.RACT = 1 detected 1: Receive descriptor active bit RD0.RACT = 0 detected	R/W
b18	FR	Frame Reception	0: Frame has not been received 1: Frame has been received	R/W
b19	TFUF	Transmit FIFO Underflow	0: Underflow has not occurred 1: Underflow has occurred	R/W
b20	TDE	Transmit Descriptor Empty	0: Transmit descriptor active bit TD0.TACT = 1 detected 1: Transmit descriptor active bit TD0.TACT = 0 detected	R/W
b21	TC	Frame Transmit Complete	0: Transfer not complete, or no transfer directive issued 1: Transfer complete	R/W
b22	ECI	ETHERC Status Register Source	0: ETHERC status interrupt source has not been detected 1: ETHERC status interrupt source has been detected	R*1
b23	ADE	Address Error	0: Illegal memory address has not been detected (normal operation) 1: Illegal memory address has been detected*2	R/W
b24	RFCOF	Receive Frame Counter Overflow	0: Receive frame counter has not overflowed 1: Receive frame counter has overflowed	R/W

Bit	Symbol	Bit Name	Description	R/W
b25	RABT	Receive Abort Detect	0:Frame reception has not been aborted or no reception directive has been issued 1:Frame reception has been aborted	R/W
b26	TABT	Transmit Abort Detect	0:Frame transmission has not been aborted or no transmission directive has been issued 1:Frame transmission has been aborted	R/W
b29 to b27	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b30	TWB	Write-Back Complete	0: Write-back has not been completed, or no transmission directive has been issued 1: Write-back has been completed	R/W
b31	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. The ECI bit is read-only. When the source in ECSR in the ETHERC is cleared, the ECI bit is also cleared.

Note 2. When an address error is detected, the EDMAC halts transmission or reception. To resume the operation, execute a software reset with the EDMR.SWR bit.

EESR shows communications status on the EDMAC and the ETHERC.

The information in EESR is reported in the form of interrupt sources. Individual bits are cleared by writing 1 (however, the ECI bit is a read-only bit that is not cleared by writing 1) and are not affected by writing 0. Each interrupt source can be masked by means of the corresponding bit in the ETHERC/EDMAC status interrupt permission register (EESIPR).

RTSF Bit (Receive Too-Short Frame)

This bit indicates that a frame of fewer than 64 bytes has been received.

RTLF Bit (Receive Too-Long Frame)

This bit indicates that a frame whose byte size exceeds the upper limit for the receive frame length set by the receive frame length register (RFLR) in the ETHERC has been received.

TRO Bit (Transmit Retry Over)

This bit indicates that a retry-over condition has occurred during frame transmission. A total of 16 transmission retries including 15 retries based on the back-off algorithm have failed after the ETHERC transmission starts.

CD Bit (Delayed Collision Detect)

This bit indicates that a delayed collision has been detected during frame transmission.

DLC Bit (Detect Loss of Carrier)

This bit indicates that loss of the carrier has been detected during frame transmission.

RFOF Bit (Receive FIFO Overflow)

This bit indicates that the receive FIFO has overflowed during frame reception.

RDE Bit (Receive Descriptor Empty)

When receive descriptor empty (RDE = 1) occurs, reception can be resumed by setting the RD0.RACT bit of the receive descriptor to 1 and then restarting the receive operation.

FR Bit (Frame Reception)

This bit indicates that a frame has been received and the receive descriptor has been updated. The FR bit is set to 1 each time a frame is received.

TFUF Bit (Transmit FIFO Underflow)

This bit indicates that the transmit FIFO has underflowed during frame transmission. Incomplete data is sent onto the line.

TDE Bit (Transmit Descriptor Empty)

This bit indicates that the transmit descriptor active bit (TD0.TACT) of a transmit descriptor read by the EDMAC is not set if the previous descriptor does not represent the end of a frame in multi-buffer frame processing. As a result, an incomplete frame may be sent.

When transmit descriptor empty (TDE = 1) occurs, execute a software reset and initiate transmission. In this case, transmission starts from the address that is stored in TDLAR.

TC Bit (Frame Transmit Complete)

This bit indicates that all the data specified by the transmit descriptor has been transmitted from the ETHERC. The TC bit is set to 1, assuming the completion of transmission, when transmission of one frame is completed in single-frame/single-buffer operation or when the last data of a frame has been transmitted and the transmit descriptor active bit (TD0.TACT) of the next descriptor is not set in multi-buffer frame processing. After frame transmission, the EDMAC writes the transmission status back to the relevant descriptor.

ADE Bit (Address Error)

This bit indicates that the memory address that the EDMAC tried to transfer is found illegal.

RFCOF Bit (Receive Frame Counter Overflow)

This bit indicates that the frame counter in the receive FIFO has overflowed.

RABT Bit (Receive Abort Detect)

This bit indicates that the ETHERC has aborted receiving a frame because of failures during frame reception.

TABT Bit (Transmit Abort Detect)

This bit indicates that the ETHERC has aborted transmitting a frame because of failures during frame transmission.

TWB Bit (Write-Back Complete)

This bit indicates that write-back from the EDMAC to the corresponding descriptor after frame transmission has completed. This operation is enabled only when the transmit interrupt setting (TIS) bit in the transmit interrupt setting register (TRIMD) is set to 1.

27.2.7 ETHERC/EDMAC Status Interrupt Permission Register (EESIPR)

Address: 000C 0030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TWB IP	—	—	—	TABT IP	RABT IP	RFCOF IP	ADE IP	ECI IP	TC IP	TDE IP	TFUF IP	FR IP	RDE IP	RFOF IP
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
	—	—	—	—	CND IP	DLC IP	CD IP	TRO IP	RMAF IP	—	—	RRF IP	RTLFI IP	RTSFI IP	PRE IP	CERFI IP
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	CERFIP	CRC Error on Received Frame Interrupt Enable	0: CRC error interrupt is disabled 1: CRC error interrupt is enabled	R/W
b1	PREIP	PHY-LSI Receive Error Interrupt Enable	0: PHY-LSI receive error interrupt is disabled 1: PHY-LSI receive error interrupt is enabled	R/W
b2	RTSFIP	Receive Too-Short Frame Interrupt Enable	0: Receive too-short frame interrupt is disabled 1: Receive too-short frame interrupt is enabled	R/W
b3	RTLFIPI	Receive Too-Long Frame Interrupt Enable	0: Receive too-long frame interrupt is disabled 1: Receive too-long frame interrupt is enabled	R/W
b4	RRFIP	Receive Residual-Bit Frame Interrupt Enable	0: Receive residual-bit frame interrupt is disabled 1: Receive residual-bit frame interrupt is enabled	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	RMAFIP	Receive Multicast Address Frame Interrupt Enable	0: Receive multicast address frame interrupt is disabled 1: Receive multicast address frame interrupt is enabled	R/W
b8	TROIP	Transmit Retry Over Interrupt Enable	0: Transmit retry over interrupt is disabled 1: Transmit retry over interrupt is enabled	R/W
b9	CDIP	Delayed Collision Detect Interrupt Enable	0: Delayed collision detect interrupt is disabled 1: Delayed collision detect interrupt is enabled	R/W
b10	DLCIP	Detect Loss of Carrier Interrupt Enable	0: Detect loss of carrier interrupt is disabled 1: Detect loss of carrier interrupt is enabled	R/W
b11	CNDIP	Carrier Not Detect Interrupt Enable	0: Carrier not detect interrupt is disabled 1: Carrier not detect interrupt is enabled	R/W
b15 to b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b16	RFOFIP	Receive FIFO Overflow Interrupt Enable	0: Overflow interrupt is disabled 1: Overflow interrupt is enabled	R/W
b17	RDEIP	Receive Descriptor Empty Interrupt Enable	0: Receive descriptor empty interrupt is disabled 1: Receive descriptor empty interrupt is enabled	R/W
b18	FRIP	Frame Reception Interrupt Enable	0: Frame reception interrupt is disabled 1: Frame reception interrupt is enabled	R/W
b19	TFUFIP	Transmit FIFO Underflow Interrupt Enable	0: Underflow interrupt is disabled 1: Underflow interrupt is enabled	R/W
b20	TDEIP	Transmit Descriptor Empty	0: Transmit descriptor empty interrupt is disabled 1: Transmit descriptor empty interrupt is enabled	R/W
b21	TCIP	Frame Transmission Complete Interrupt Enable	0: Frame transmission complete interrupt is disabled 1: Frame transmission complete interrupt is enabled	R/W
b22	ECIIP	ETHERC Status Register Source Interrupt Enable	0: ETHERC status interrupt is disabled 1: ETHERC status interrupt is enabled	R/W
b23	ADEIP	Address Error Interrupt Enable	0: Address error interrupt is disabled 1: Address error interrupt is enabled	R/W

Bit	Symbol	Bit Name	Description	R/W
b24	RFCOFIP	Receive Frame Counter Overflow Interrupt Enable	0: Receive frame counter overflow interrupt is disabled 1: Receive frame counter overflow interrupt is enabled	R/W
b25	RABTIP	Receive Abort Detect Interrupt Enable	0: Receive abort detect interrupt is disabled 1: Receive abort detect interrupt is enabled	R/W
b26	TABTIP	Transmit Abort Detect Interrupt Enable	0: Transmit abort detect interrupt is disabled 1: Transmit abort detect interrupt is enabled	R/W
b29 to b27	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b30	TWBIP	Write-Back Complete Interrupt Enable	0: Write-back complete interrupt is disabled 1: Write-back complete interrupt is enabled	R/W
b31	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

EESIPR enables interrupts corresponding to individual bits in the ETHERC/EDMAC status register (EESR). An interrupt is enabled by writing 1 to the corresponding bit.

27.2.8 Transmit/Receive Status Copy Enable Register (TRSCER)

Address: 000C 0038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CND CE	DLC CE	CD CE	TRO CE	RMAF CE	—	—	RRF CE	RTLF CE	RTSF CE	PRE CE	CERF CE
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	CERFCE	CERF Bit Copy Directive	0: Reflects the EESR.CERF bit status in the RD0.RFS bit in the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RD0.RFS bit in the receive descriptor	R/W
b1	PRECE	PRE Bit Copy Directive	0: Reflects the EESR.PRE bit status in the RD0.RFS bit in the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RD0.RFS bit in the receive descriptor	R/W
b2	RTSFCE	RTSF Bit Copy Directive	0: Reflects the EESR.RTSF bit status in the RD0.RFS bit in the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RD0.RFS bit in the receive descriptor	R/W
b3	RTLFCE	RTLF Bit Copy Directive	0: Reflects the EESR.RTLF bit status in the RD0.RFS bit in the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RD0.RFS bit in the receive descriptor	R/W
b4	RRFCE	RRF Bit Copy Directive	0: Reflects the EESR.RRF bit status in the RD0.RFS bit in the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RD0.RFS bit in the receive descriptor	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	RMAFCE	RMAF Bit Copy Directive	0: Reflects the EESR.RMAF bit status in the RD0.RFS bit in the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RD0.RFS bit in the receive descriptor	R/W
b8	TROCE	TRO Bit Copy Directive	0: Reflects the EESR.TRO bit status in the TD0.TFS bit in the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TD0.TFS bit in the transmit descriptor	R/W
b9	CDCE	CD Bit Copy Directive	0: Reflects the EESR.CD bit status in the TD0.TFS bit in the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TD0.TFS bit in the transmit descriptor	R/W
b10	DLCCE	DLC Bit Copy Directive	0: Reflects the EESR.DLC bit status in the TD0.TFS bit in the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TD0.TFS bit in the transmit descriptor	R/W
b11	CNDCE	CND Bit Copy Directive	0: Reflects the EESR.CND bit status in the TD0.TFS bit in the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TD0.TFS bit in the transmit descriptor	R/W
b31 to b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

TRSCER specifies whether the information for the transmit and receive state reported by bits in the ETHERC/EDMAC status register (EESR) is to be reflected in the TFS25 to TFS0 or RFS26 to RFS0 bits of the corresponding descriptor.

The bits in TRSCER correspond to bits 11 to 0 in EESR. When a bit is cleared to 0, the transmit status (bits 11 to 8 in EESR) is reflected in the TFS3 to TFS0 bits of the transmit descriptor, and the receive status (bits 7 to 0 in EESR) is reflected in the RFS7 to RFS0 bits of the receive descriptor. When a bit is set to 1, the occurrence of the corresponding source is not reflected in the descriptor. After this LSI is reset, all bits are cleared to 0.

27.2.9 Receive Missed-Frame Counter Register (RMFCR)

Address: 000C 0040h

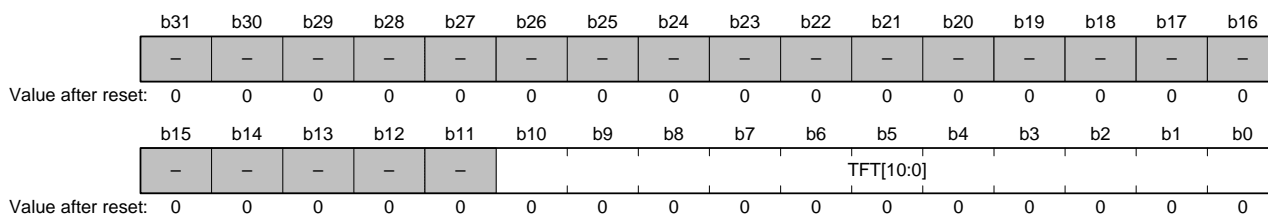
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MFC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MFC[15:0]	Missed-Frame Counter	These bits indicate the number of frames that are discarded and not transferred to the receive buffer during reception.	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RMFCR indicates the number of frames that could not be stored in the receive buffer and so were discarded during reception. When the receive FIFO overflows, the receive frames in the FIFO are discarded. The number of frames discarded at this time is counted. When the value in RMFCR reaches FFFFh, count-up is halted. The counter value is cleared to 0 by a write to RMFCR with any value.

27.2.10 Transmit FIFO Threshold Register (TFTR)

Address: 000C 0048h



Bit	Symbol	Bit Name	Description	R/W
b10 to b0	TFT[10:0]	Transmit FIFO Threshold	000h: Store and forward mode 001h to 00Ch: Setting prohibited 00Dh: 52 bytes 00Eh: 56 bytes : : 01Fh: 124 bytes 020h: 128 bytes : : 03Fh: 252 bytes 040h: 256 bytes : : 07Fh: 508 bytes 080h: 512 bytes : : 0FFh: 1020 bytes 100h: 1024 bytes : : 1FFh: 2044 bytes 200h: 2048 bytes 201h to 7FFh: Setting prohibited	R/W
b31 to b11	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

- Note 1. When starting transmission before one frame of data write has completed, take care no underflow occurs.
- Note 2. Operation cannot be guaranteed when the value set in TFTR is greater than the transmit FIFO size.
- Note 3. To prevent a transmit underflow, setting the initial value (store and forward modes) is recommended.

TFTR specifies the transmit FIFO threshold at which the first transmission is started. The actual threshold is four times the set value.

The ETHERC starts transmission when the amount of data in the transmit FIFO exceeds the number of bytes specified in TFTR, when the transmit FIFO is full, or when one frame of data write is performed. TFTR must be set in the transmission-halt state.

TFT[10:0] Bits (Transmit FIFO Threshold)

The transmit FIFO threshold must be set to a value smaller than the FIFO size specified by the FIFO depth register (FDR).

27.2.11 FIFO Depth Register (FDR)

Address: 000C 0050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	–	–	–			TFD[4:0]			–	–	–			RFD[4:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	RFD[4:0]	Receive FIFO Size	00000: 256 bytes 00001: 512 bytes 00010: 768 bytes 00011: 1024 bytes 00100: 1280 bytes 00101: 1536 bytes 00110: 1792 bytes 00111: 2048 bytes Other than the above: Setting prohibited	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b12 to b8	TFD[4:0]	Transmit FIFO Size	00000: 256 bytes 00001: 512 bytes 00010: 768 bytes 00011: 1024 bytes 00100: 1280 bytes 00101: 1536 bytes 00110: 1792 bytes 00111: 2048 bytes Other than the above: Setting prohibited	R/W
b31 to b13	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note: Operation cannot be guaranteed when the value set in FDR is greater than the transmit or receive FIFO size.

FDR specifies the sizes of the transmit and receive FIFOs.

To activate the ETHERC, set this register to 0000 0707h.

RFD[4:0] Bits (Receive FIFO Size)

These bits specify the size of the receive FIFO. The setting must not be changed after transmission/reception has started.

TFD[4:0] Bits (Transmit FIFO Size)

These bits specify the size of the transmit FIFO. The setting must not be changed after transmission/reception has started.

27.2.12 Receiving Method Control Register (RMCR)

Address: 000C 0058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	RNC	RNR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

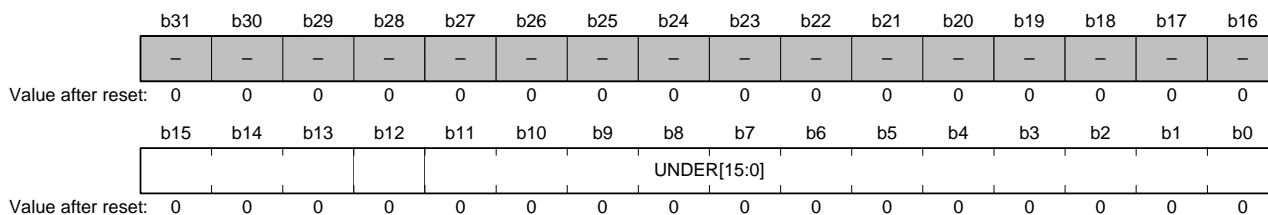
Bit	Symbol	Bit Name	Description	R/W
b0	RNR	Receive Request Bit Reset	0:Allows the hardware to reset the receive request bit (RR) in EDRRR automatically upon completion of reception of one frame. Control is possible for each frame. To receive the subsequent receive frame, the RR bit needs to be set again. 1:Allows the higher-level software to control the receive request bit (RR) in EDRRR. Once the RR bit is set to 1, the hardware continues to fetch the receive descriptor and receive frames automatically until the RR bit is cleared to 0. In other words, continuous reception of multiple frames is possible. It is recommended to set the RNR bit to 1 when continuous reception is used. However, when a receive descriptor empty is detected, the hardware clears the RR bit automatically.	R/W
b1	RNC	Receive Request Bit Non-Reset Mode	0:Nop 1:Allows the software to reset the receive request bit (RR) in EDRRR. In this case, even when the RD0.RACT bit in the fetched descriptor is 0 (receive descriptor empty), the RR bit is not automatically reset and the receive descriptor is continuously fetched to continue DMA transfers of the receive frames.	R/W
b31 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RMCR specifies the control method for the receive request (RR) bit in the EDMAC receive request register (EDRRR) while a frame is received.

RMCR must be set in the receiving-halted state.

27.2.13 Transmit FIFO Underrun Counter (TFUCR)

Address: 000C 0064h



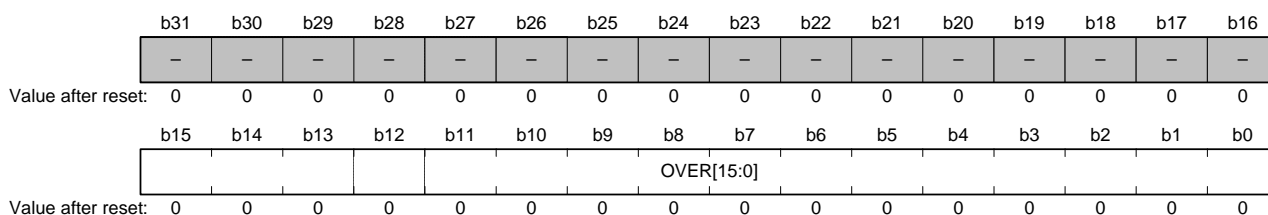
Bit	Symbol	Bit Name	Description	R/W
b15 to b0	UNDER[15:0]	Transmit FIFO Underflow Count	Indicates the count of underflows having occurred in the transmit FIFO. The counter stops when the count value reaches FFFFh.	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

TFUCR indicates the count of underruns having occurred in the transmit FIFO.

The count value is cleared to 0 by writing any value to this register.

27.2.14 Receive FIFO Overflow Counter (RFOCR)

Address: 000C 0068h



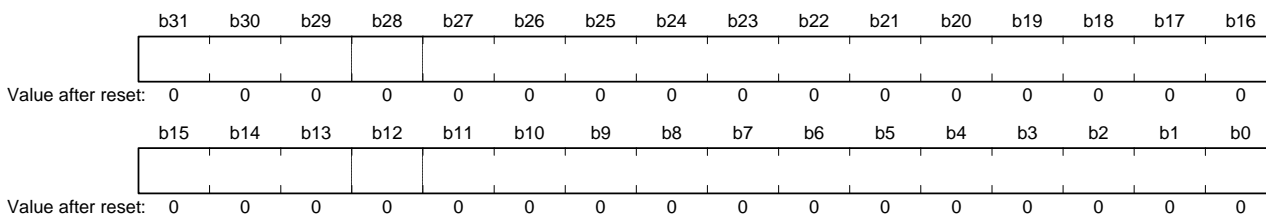
Bit	Symbol	Bit Name	Description	R/W
b15 to b0	OVER[15:0]	Receive FIFO Overflow Count	Indicates the count of overflows having occurred in the receive FIFO. The counter stops when the count value reaches FFFFh.	R/W
b31 to b16	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RFOCR indicates the count of overflows having occurred in the receive FIFO.

The count value is cleared to 0 by writing any value to this register.

27.2.15 Receive Buffer Write Address Register (RBWAR)

Address: 000C 00C8h



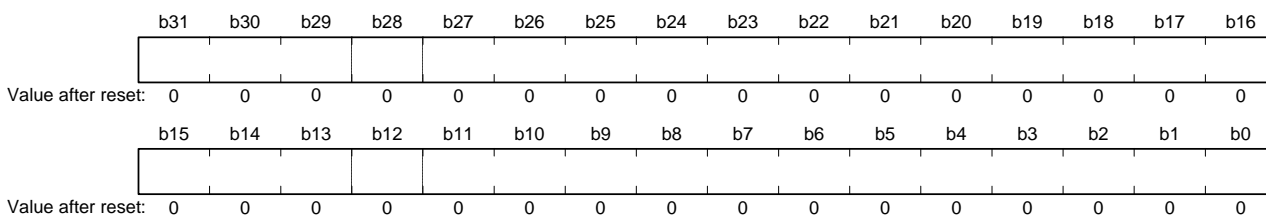
Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits can only be read. Writing is prohibited.	R

RBWAR stores the address of data to be written in the receive buffer by the EDMAC.

Which addresses in the receive buffer are processed by the EDMAC can be recognized by monitoring addresses indicated in RBWAR. The address to which the EDMAC is actually writing may be different from the value read from RBWAR.

27.2.16 Receive Descriptor Fetch Address Register (RDFAR)

Address: 000C 00CCh



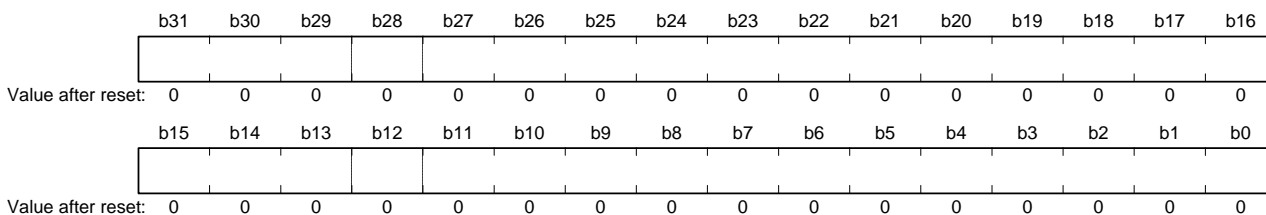
Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits can only be read. Writing is prohibited.	R

RDFAR stores the descriptor start address that is required when the EDMAC fetches descriptor information from the receive descriptor.

Which receive descriptor information is used for processing by the EDMAC can be recognized by monitoring addresses indicated in RDFAR. The address from which the EDMAC is actually fetching a descriptor may be different from the value read from RDFAR.

27.2.17 Transmit Buffer Read Address Register (TBRAR)

Address: 000C 00D4h



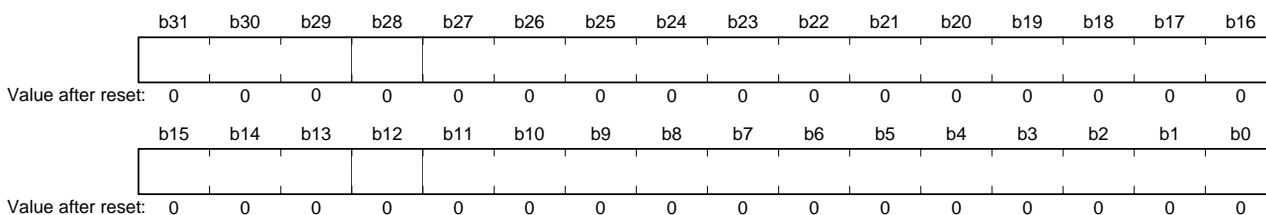
Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits can only be read. Writing is prohibited.	R

TBRAR stores the address of data to be read from the transmit buffer by the EDMAC.

Which addresses in the transmit buffer are processed by the EDMAC can be recognized by monitoring addresses indicated in TBRAR. The address from which the EDMAC is actually reading may be different from the value read from TBRAR.

27.2.18 Transmit Descriptor Fetch Address Register (TDFAR)

Address: 000C 00D8h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits can only be read. Writing is prohibited.	R

TDFAR stores the descriptor start address that is required when the EDMAC fetches descriptor information from the transmit descriptor.

Which transmit descriptor information is used for processing by the EDMAC can be recognized by monitoring addresses indicated in TDFAR. The address from which the EDMAC is actually fetching a descriptor may be different from the value read from TDFAR.

27.2.19 Flow Control Start FIFO Threshold Setting Register (FCFTR)

Address: 000C 0070h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
–	–	–	–	–	–	–	–	–	–	–	–	–	RFFO[2:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
–	–	–	–	–	–	–	–	–	–	–	–	–	RFDO[2:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RFFO[2:0]	Receive FIFO Overflow BSY Output Threshold	b2 b0 000: When 256 – 32 bytes of data is stored in the receive FIFO. 001: When 512 – 32 bytes of data is stored in the receive FIFO. : 110: When 1792 – 32 bytes of data is stored in the receive FIFO. 111: When 2048 – 64 bytes of data is stored in the receive FIFO.	R/W
b15 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b18 to b16	RFFO[2:0]	Receive Frame Count Overflow BSY Output Threshold	b18 b16 000: When two receive frames have been stored in the receive FIFO. 001: When four receive frames have been stored in the receive FIFO. 010: When six receive frames have been stored in the receive FIFO. : 110: When 14 receive frames have been stored in the receive FIFO. 111: When 16 receive frames have been stored in the receive FIFO.	R/W
b31 to b19	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

FCFTR specifies the flow control of the ETHERC (specifies the threshold of automatic PAUSE output).

The threshold can be set in terms of the data size in the receive FIFO (RFDO[2:0] bits) and the number of receive frames (RFFO[2:0] bits). Flow control is turned on when either of the data size in the receive FIFO or the number of receive frames satisfies the corresponding threshold condition.

If the same receive FIFO size as set by the FIFO depth register (FDR) is set in FCFTR when flow control is to be turned on according to the condition set in the RFDO[2:0] bits, flow control is turned on with (FIFO data size – 64) bytes. For instance, when the FDR.RFD[4:0] bits = 00111b and the FCFTR.RFDO[2:0] bits = 111, flow control is turned on when (2,048 – 64) bytes of data is stored in the receive FIFO. The value set in the RFDO[2:0] bits should be equal to or smaller than the value set in the FDR.RFD[4:0] bits.

27.2.20 Receive Data Padding Insert Register (RPADIR)

Address: 000C 0078h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PADS[1:0]		
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	
	—	—	—	—	—	—	—	—	—	—	PADR[5:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PADR[5:0]	Padding Slot	00h: Inserts the specified padding size immediately before the first byte of the receive data. 01h: Inserts the specified padding size immediately before the second byte of the receive data. : 3Eh: Inserts the specified padding size immediately before the 63rd byte of the receive data. 3Fh: Inserts the specified padding size immediately before the 64th byte of the receive data.	R/W
b15 to b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b17, b16	PADS[1:0]	Padding Size	b17 b16 0 0: No padding insertion 0 1: 1-byte insertion 1 0: 2-byte insertion 1 1: 3-byte insertion	R/W
b31 to b18	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

RPADIR specifies padding insertion in receive data.

Before modifying the settings of RPADIR, execute a software reset through the EDMR.SWR bit.

27.2.21 Transmit Interrupt Setting Register (TRIMD)

Address: 000C 007Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	–	–	–	–	–	–	–	–	–	–	–	TIM	–	–	–	TIS
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	TIS	Transmit Interrupt Setting	0:Interrupt not set No interrupt notification is sent in the mode selected by the TIM bit. When the TIS bit is 0, the TIM bit setting is invalid. 1:Interrupt set An interrupt notification is sent by setting the EESR.TWB bit to 1 in the mode selected by the TIM bit.	R/W
b3 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	TIM	Transmit Interrupt Mode	0:Per-transmit-frame mode An interrupt is issued upon write-back completion of each frame. 1:Interrupt mode An interrupt is issued upon write-back completion of the transmit descriptor with the TD0.TWBI bit set to 1.	R/W
b31 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

TRIMD specifies whether to notify write-back completion of each frame during transmission by means of the EESR.TWB bit and an interrupt.

27.2.22 Independent Output Signal Setting Register (IOSR)

Address: 000C 006Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	ELB
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	ELB	External Loopback Mode	0: The ET_EXOUT pin outputs low. 1: The ET_EXOUT pin outputs high.	R/W
b31 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

IOSR is a register to select the output level to the external output pin (ET_EXOUT) in external loopback mode. The ET_EXOUT pin can be used to specify loopback mode for the PHY-LSI. To use the loopback function of the PHY-LSI through this register, the PHY-LSI needs to be provided with the pin to be connected to the ET_EXOUT pin.

27.3 Operation

The EDMAC, connected to the ETHERC, allows efficient transfer of transmit/receive data between the ETHERC and memory (buffers) without CPU intervention. The EDMAC automatically reads the control information, called descriptors. The descriptors corresponding to each buffer hold buffer pointers and other information. The EDMAC reads transmit data from the transmit buffer and writes receive data to the receive buffer according to the control information. By arranging such multiple descriptors continuously (i.e., making a descriptor list), data can be transmitted or received sequentially.

27.3.1 Descriptor Lists and Data Buffers

By the communication program, a transmit descriptor list and a receive descriptor list should be created in memory space prior to transmission and reception. The start addresses of these lists should be set in the transmit descriptor list start address register and receive descriptor list start address register.

The start addresses of the descriptor lists should be placed on the address boundaries in accordance with the descriptor length specified in the EDMAC mode register (EDMR). Here, the start address of the transmit buffer can be placed on a longword, word, or byte boundary.

27.3.1.1 Transmit Descriptor

Figure 27.2 shows the relationship between a transmit descriptor and a transmit buffer. The transmit descriptor can relate one transmit frame to one transmit buffer (single-frame/single-buffer operation) or one frame to multiple transmit buffers (single-frame/multi-buffer operation).

When the transmit buffer length (TBL) is to be set to 1 to 16 bytes, the buffer address needs to be placed on a 32-byte boundary. When the transmit buffer length (TBL) is set to 0 byte, operation cannot be guaranteed.

Each transmit descriptor is cleared to 0000 0000h by a reset.

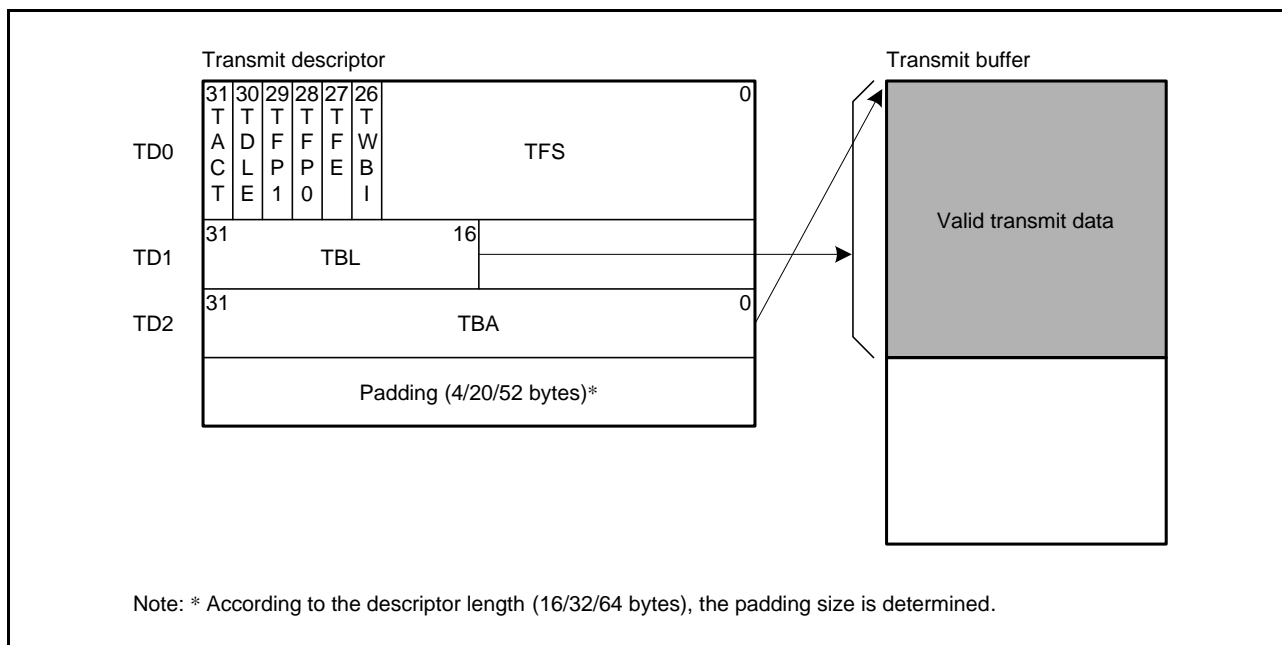


Figure 27.2 Relationship between Transmit Descriptor and Transmit Buffer

(1) Transmit Descriptor 0 (TD0)

Bit	Symbol	Bit Name	Description	R/W
<u>b25 to b0</u>	<u>TFS</u>	Transmit Frame Status	TFS25 to TFS9: Reserved (The write value should always be 0.) TFS8: Detect Transmit Abort When set to 1, this bit indicates that the abort signal is set to 1 during frame transmission (causing TFE to be set). TFS7 to TFS4: Reserved (The write value should always be 0.) TFS3: Detect No Carrier (corresponding to the EESR.CND bit) TFS2: Detect Loss of Carrier (corresponding to the EESR.DLC bit) TFS1: Detect Delayed Collision during Transmission (corresponding to the EESR.CD bit) TFS0: Transmit Retry Over (corresponding to the EESR.TRO bit) When set to 1, these bits indicate that TFS8 to TFS1 have been set to 1 during frame transmission. (Although TFE is normally set when these bits are set to 1, it can be prevented from being set by so setting TRSCER.)	R/W
b26	TWBI	Write-Back Completion Interrupt Notification	(This bit is valid when TRIMD is set so.) 0: Nop 1: An interrupt is generated upon completion of write-back to this descriptor.	R/W
<u>b27</u>	<u>TFE</u>	Transmit Frame Error	0: Frame transmission is continued (normal operation). 1: Frame transmission has been aborted.	R/W
b29, b28	TFP[1:0]	Transmit Frame Position	^{b29 b28} 0 0: Transmission of the frame of the transmit buffer specified by this descriptor is continued. (The frame is incomplete.) 0 1: The transmit buffer specified by this descriptor contains the end of the frame (The frame is complete.) 1 0: The transmit buffer specified by this descriptor is the start of the frame (The frame is incomplete.) 1 1: The contents in the transmit buffer specified by this descriptor correspond to one frame (single-frame/single-buffer).	R/W
b30	TDLE	Transmit Descriptor Ring End	When set to 1, the TDLE bit indicates that the corresponding descriptor is the last one of the descriptor ring.	R/W
<u>b31</u>	<u>TACT</u>	Transmit Descriptor Active	Indicates that the corresponding descriptor is active.	R/W

Note: The underlined bits are subject to write-back.

TD0 indicates the transmit frame status, informing frame transmission status.

TFE Bit (Transmit Frame Error)

When set to 1, the TFE bit indicates that an error is indicated by any of the TFS bits. (Through the TRSCER setting, it is possible to prevent this bit from being set by an event indicated by TFS7 to TFS0. It is impossible, however, if an event indicated by TFS7 to TFS0 also causes TFS8 to be set.)

TFP[1:0] Bits (Transmit Frame Position)

These bits relate the transmit buffer to the transmit frame. The settings of the TFP bits and the TBL bits should be logically correct in the consecutive descriptors.

TACT Bit (Transmit Descriptor Active)

This bit indicates that the corresponding descriptor is active. The TACT bit is set to 1 by software. This bit is cleared to 0 by hardware when a transmit frame has been completely transferred or when transmission has been aborted due to some cause.

(2) Transmit Descriptor 1 (TD1)

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b31 to b16	TBL	Transmit Buffer Length	Indicates the length of the relevant transmit buffer in terms of valid bytes.	R/W

TD1 indicates the length of the transmit buffer.

(3) Transmit Descriptor 2 (TD2)

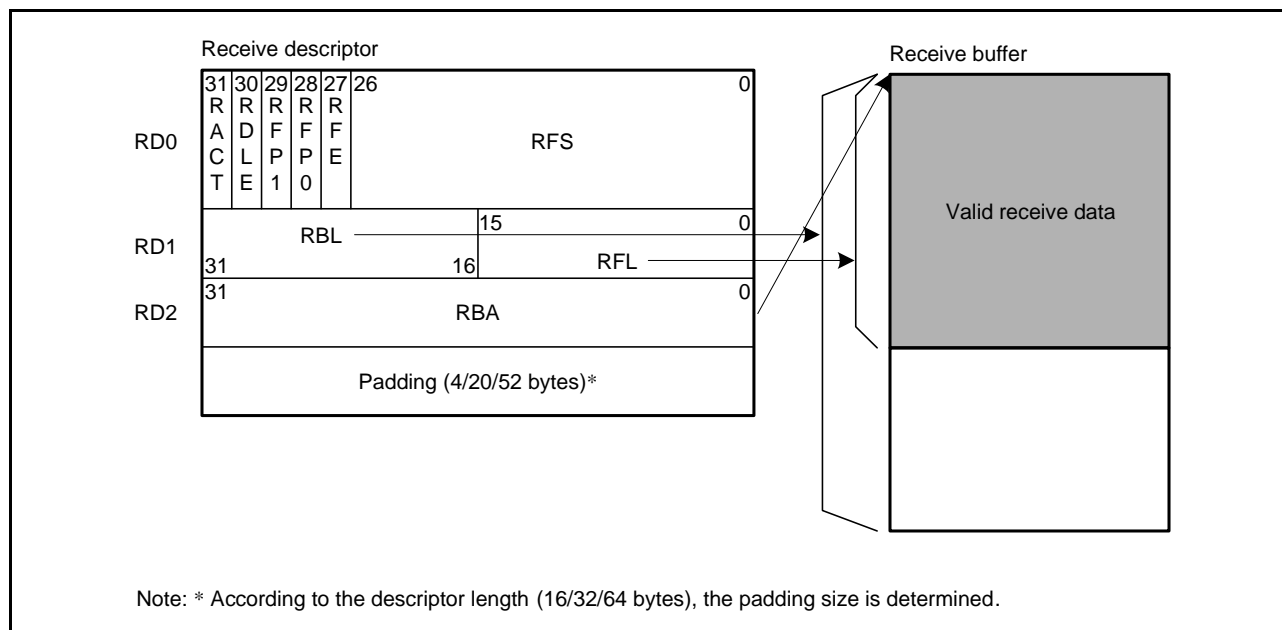
Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TBA	Transmit Buffer Address	Indicates the start address of the transmit buffer.	R/W

27.3.1.2 Receive Descriptor

Figure 27.3 shows the relationship between a receive descriptor and a receive buffer. The receive buffer address should be placed on a 32-byte boundary.

When the receive buffer length (RBL) is set to 0 byte, operation specified by the descriptor cannot be guaranteed.

Each receive descriptor is cleared to 0000 0000h by a reset.



Note: * According to the descriptor length (16/32/64 bytes), the padding size is determined.

Figure 27.3 Relationship between Receive Descriptor and Receive Buffer

(1) Receive Descriptor 0 (RD0)

Bit	Symbol	Bit Name	Description	R/W
b26 to b0	RFS	Receive Frame Status	RFS26 to RFS10: Reserved (The write value should always be 0.) RFS9:Receive FIFO Overflow (corresponding to the EESR.RFOF bit) When set to 1, this bit indicates that a receive FIFO overflow has occurred terminating the frame halfway and that the frame has been written back. RFS8:Detect Receive Abort (corresponding to the EESR.RABT bit) When set to 1, this bit indicates that any of bits RFS3 to RFS0 is set to 1. RFS7:Receive Multicast Address Frame (corresponding to the EESR.RMAF bit) RFS6 and RFS5: Reserved (The write value should always be 0.) RFS4:Receive Residual-Bit Frame (corresponding to the EESR.RRF bit) RFS3:Receive Too-Long Frame (corresponding to the EESR.RTLF bit) RFS2:Receive Too-Short Frame (corresponding to the EESR.RTSF bit) RFS1:PHY-LSI Receive Error (corresponding to the EESR.PRE bit) RFS0:CRC Error on Received Frame (corresponding to the EESR.CERF bit)	R/W
b27	RFE	Receive Frame Error	0: No error in RFS 1: An error has been indicated in RFS	R/W
b29, b28	RFP[1:0]	Receive Frame Position	b29 b28 0 0: Reception of the frame of the receive buffer specified by this descriptor is continued. (The frame is incomplete.) 0 1: The receive buffer specified by this descriptor contains the end of the frame (The frame is complete.) 1 0: The receive buffer specified by this descriptor is the start of the frame (The frame is incomplete.) 1 1: The contents in the receive buffer specified by this descriptor correspond to one frame (single-frame/single-buffer).	R/W
b30	RDLE	Receive Descriptor Ring End	When set to 1, the RDLE bit indicates that the corresponding descriptor is the last one of the descriptor ring.	R/W
b31	RACT	Receive Descriptor Active	Indicates that the corresponding descriptor is active.	R/W

Note: The underlined bits are subject to write-back.

RD0 indicates the receive frame status, informing frame reception status.

RFE Bit (Receive Frame Error)

When set to 1, the RFE bit indicates that an error is indicated by any of the RFS bits. (Through the TRSCER setting, it is possible to prevent this bit from being set by an event indicated by RFS7 to RFS0. It is impossible, however, if an event indicated by RFS7 to RFS0 also causes RFS8 to be set.)

RFP[1:0] Bits (Receive Frame Position)

These bits relate the receive buffer to the receive frame.

RACT Bit (Receive Descriptor Active)

This bit indicates that the corresponding descriptor is active. The RACT bit is set to 1 by software. This bit is cleared to 0 by hardware when an entire receive frame has been completely transferred to the buffer address specified by RD2 or when the receive buffer becomes full.

(2) Receive Descriptor 1 (RD1)

Bit	Symbol	Bit Name	Description	R/W
-----	--------	----------	-------------	-----

b15 to b0	RFL	Receive Data Length	Indicates the length of (number of bytes in) a receive frame stored in the buffer. The number of bytes for padding insertion specified by RPADIR is excluded. These bits are written back to the descriptor containing the end of a frame.	R/W
b31 to b16	RBL	Receive Buffer Length	Indicates the length of the relevant receive buffer in terms of bytes. The buffer length should be set to $n \times 32$.	R/W

Note: The underlined bits are subject to write-back.

RD1 indicates the receive buffer length and receive frame length.

(3) Receive Descriptor 2 (RD2)

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RBA	Receive Buffer Address	Indicates the start address of the receive buffer. The buffer address should be set on a 32-byte boundary.	R/W

27.3.2 Transmission

When the EDTRR.TR bit is set to 1 while the transmission function is enabled, the EDMAC reads the descriptor following the previously used descriptor from the transmit descriptor list (or the descriptor indicated by the transmit descriptor start address register (TDLAR) at the initial start time). If the TD0.TACT bit of the read descriptor is set to 1 (active), the EDMAC sequentially reads transmit frame data from the transmit buffer start address specified by TD2 for transfer to the ETHERC. The ETHERC creates a transmit frame and starts transmission to the MII. After DMA transfer of data equivalent to the buffer length specified in the descriptor, the following processing is carried out according to the TD0.TFP value.

- TD0.TFP = 00 or 10 (frame continuation)
Descriptor write-back (writing to the TD0.TACT bit) is performed after DMA transfer.
- TD0.TFP = 01 or 11 (frame end)
Descriptor write-back (writing to the TD0.TACT bit and status bits) is performed after completion of frame transmission.

As long as the TD0.TACT bit of a read descriptor is set to 1 (active), the reading of EDMAC descriptors and the transmission of frames continue. When a descriptor with the TD0.TACT bit cleared to 0 (inactive) is read, the EDMAC clears the EDTRR.TR bit to 0 and completes transmit processing.

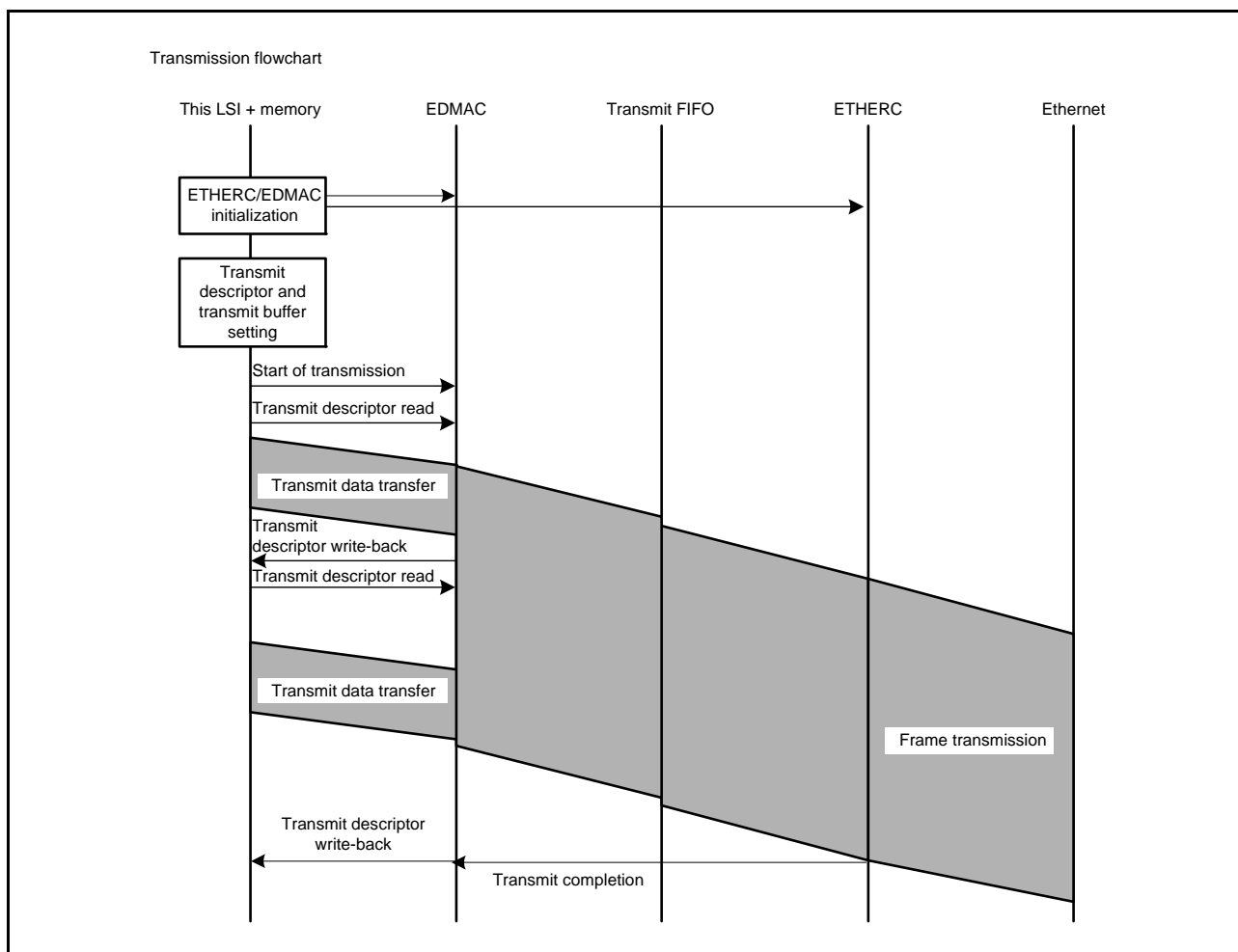


Figure 27.4 Sample Transmission Flowchart

27.3.3 Reception

When the CPU sets the EDRRR.RR bit while the receive function is enabled, the EDMAC reads the descriptor following the previously used descriptor from the receive descriptor list (or the descriptor indicated by the receive descriptor start address register (RDLAR) at the initial start time) then enters the receive standby state. Upon receiving the frame for this LSI while the RD0.RACT bit is set to 1 (active), the EDMAC transfers the frame to the receive buffer specified by RD2. If the data length of a received frame is longer than the buffer length specified by RD1, the EDMAC performs a write-back operation to the descriptor (with RD0.RFP set to 10b or 00b) when the buffer becomes full, then reads the next descriptor. The EDMAC then continues to transfer data to the receive buffer specified by the new RD2. When frame reception is completed, or if frame reception is aborted because of a certain kind of error, the EDMAC performs write-back to the relevant descriptor (with RD0.RFP set to 11b or 01b), and then ends the receive processing. The EDMAC then reads the next descriptor and enters the receive standby state again.

To receive frames continuously, the RNC bit must be set to 1 in the receive method control register (RMCR). The initial value is 0.

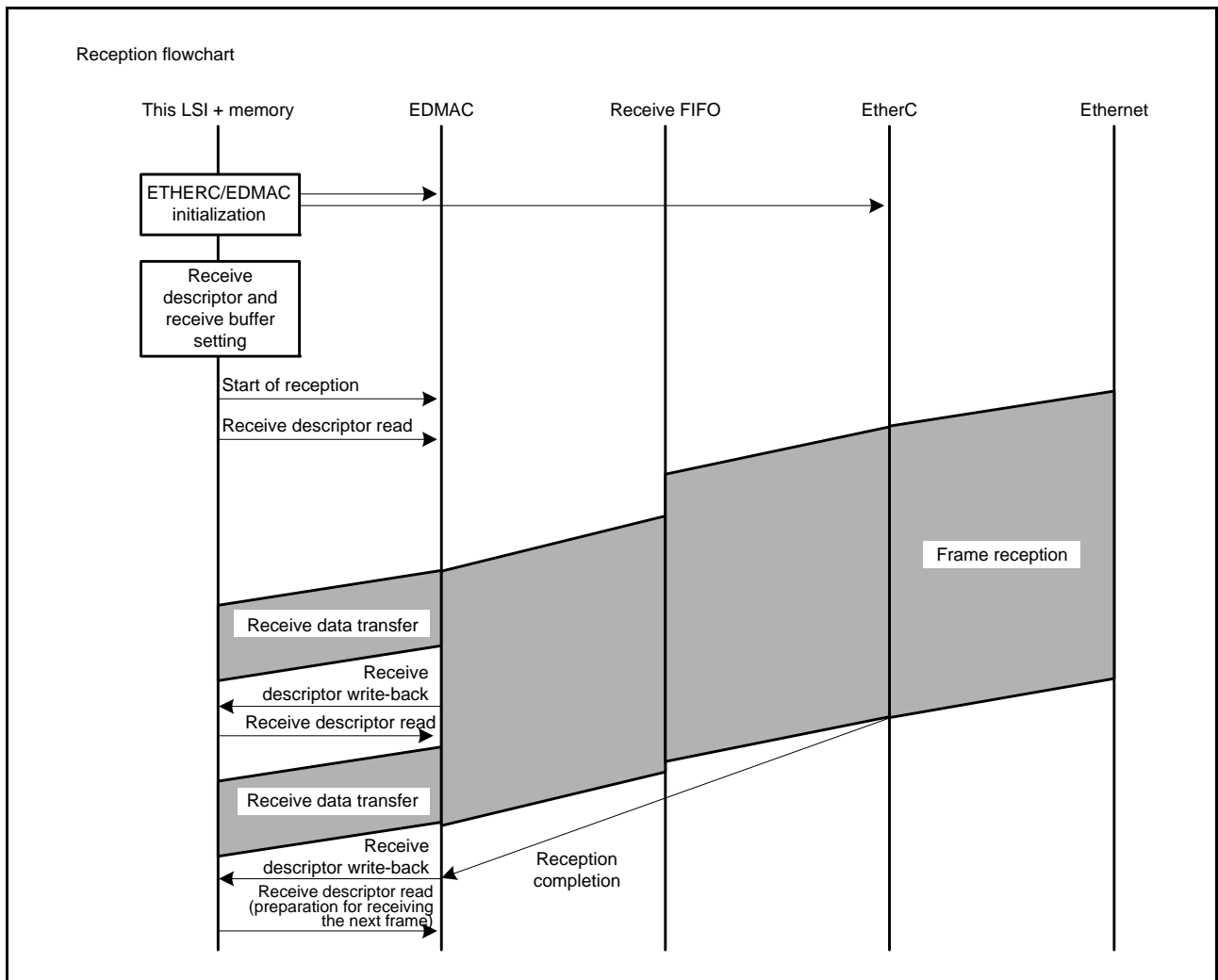


Figure 27.5 Sample Reception Flowchart

27.3.4 Transmit/Receive Processing of Multi-Buffer Frame

27.3.4.1 Multi-Buffer Frame Transmit Processing

If an error occurs during multi-buffer frame transmission, the processing shown in Figure 27.6 is carried out by the EDMAC.

In the figure, for the inactive transmit descriptors (TD0.TACT bit = 0), buffer data has already been transmitted successfully, and for the active transmit descriptors (TD0.TACT bit = 1), buffer data has not been transmitted. If a frame transmit error occurs in the first active descriptor (TD0.TACT bit = 1), transmission is halted immediately and the TD0.TACT bit is cleared to 0. The next descriptor is then read, and the position within the transmit frame is determined on the basis of the TD0.TFP[1:0] bits (continuing [00b] or end [01b]). In the case of a continuing descriptor, only the TD0.TACT bit is cleared to 0 and the next descriptor is read immediately. If the descriptor is the end descriptor, not only is the TD0.TACT bit cleared to 0, but write-back is also performed to the TD0.TFE and TD0.TFS bits at the same time. Data in the buffer is not transmitted between the occurrence of an error and write-back to the end descriptor. If error interrupts are enabled in the ETHERC/EDMAC status interrupt permission register (EESIPR), an interrupt is generated immediately after the write-back to the end descriptor.

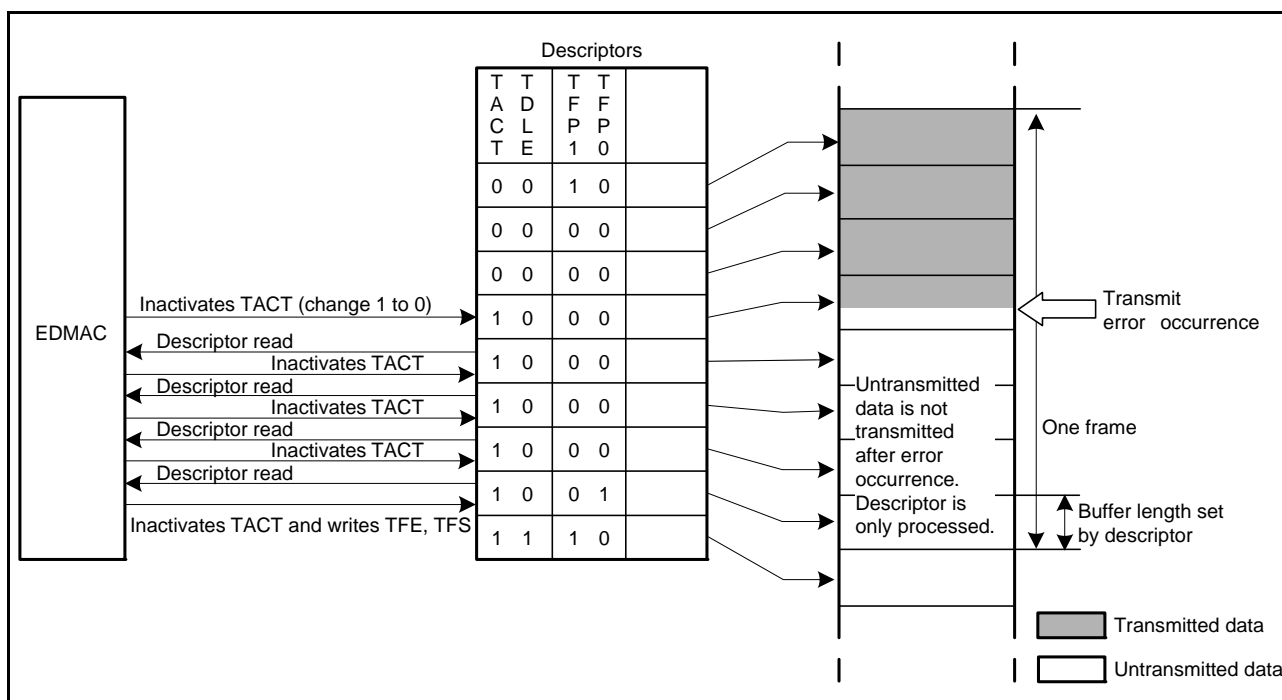


Figure 27.6 EDMAC Operation after Transmit Error

27.3.4.2 Receive Processing in Case of Multi-Buffer Frame

If an error occurs during multi-buffer frame reception, the EDMAC performs the processing shown in Figure 27.7.

In the figure, for the inactive receive descriptors (RD0.RACT bit = 0), the buffers have received data successfully, and for the active receive descriptors (RD0.RACT bit = 1), the buffers have not received data. If a frame receive error occurs in the first active descriptor (RD0.RACT bit = 1) in the figure, the status is written back to the descriptor.

If error interrupts are enabled in EESIPR, an interrupt is generated immediately after the write-back. If there is a new frame receive request, reception is continued from the next buffer after the one in which the error occurred.

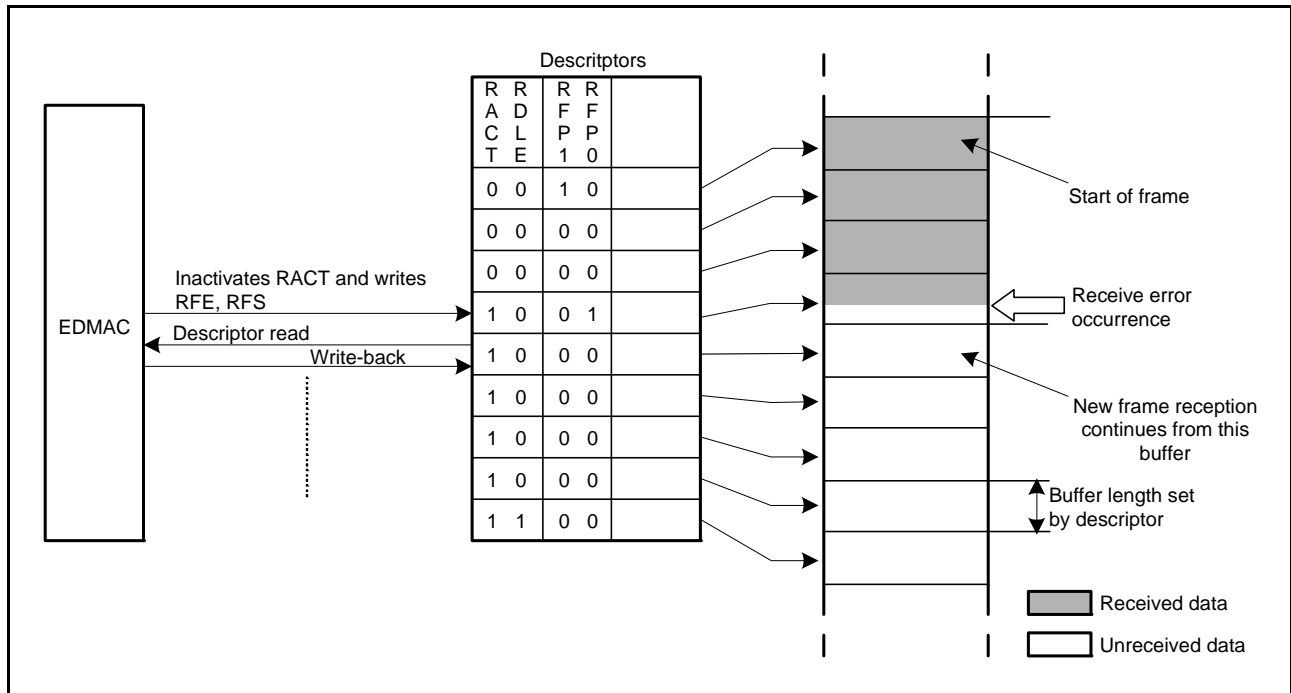


Figure 27.7 EDMAC Operation after Receive Error

28. USB 2.0 Host/Function Module (USB)

28.1 Overview

The RX62N/RX621 Group provides two ports of USB2.0 host/function module (USB).

The USB is a USB controller which provides capabilities as a USB host controller and a USB function controller. The USB supports full-speed transfer defined by the USB (universal serial bus) Specifications 2.0 when used as the host controller, and supports full-speed transfer when used as the function controller. The USB has a USB transceiver and supports all of the transfer types defined by the USB Specifications.

The USB has buffer memory for data transfer, providing a maximum of ten pipes. Any endpoint numbers can be assigned to PIPE1 to PIPE9, based on the peripheral devices or user system for communication.

Table 28.1 shows the specifications of the USB.

Table 28.1 Specifications of USB

Item	Specifications
Features	<ul style="list-style-type: none"> • USB Device Controller (UDC) and transceiver for USB2.0 are incorporated. • Two ports are provided. • The USB host controller and USB function controller are incorporated (can be switched by software). • Self-power mode or bus-power mode can be selected. • OTG (ON-The-Go) is supported. <p>(1) Features of the USB host controller</p> <ul style="list-style-type: none"> • Full-speed transfer (12 Mbps) is supported • Communications with multiple peripheral devices connected via a single HUB • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers <p>(2) Features of the USB function controller</p> <ul style="list-style-type: none"> • Full-speed transfer (12 Mbps) is supported • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF interpolation function
Communication data transfer type	<ul style="list-style-type: none"> • Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer
Internal bus interface	<ul style="list-style-type: none"> • Connected to internal peripheral bus 3
Pipe configuration	<ul style="list-style-type: none"> • Buffer memory for USB communications is provided. • Up to ten pipes can be selected (including the default control pipe). • Endpoint numbers can be assigned flexibly to PIPE1 to PIPE9. • Transfer conditions that can be set for each pipe: <ul style="list-style-type: none"> PIPE0:Control transfer only (default control pipe: DPC) Buffer size: 8, 16, 32, or 64 bytes (single buffer) PIPE1 and PIPE2:Bulk transfer or isochronous transfer Buffer size: 8, 16, 32, or 64 bytes for bulk transfer or 1 to 256 bytes for isochronous transfer (double buffer can be specified) PIPE3 to PIPE5:Bulk transfer only Buffer size: 8, 16, 32, or 64 bytes (double buffer can be specified) PIPE6 to PIPE9:Interrupt transfer only Buffer size: 1 to 64 bytes (single buffer)
Others	<ul style="list-style-type: none"> • Reception ending function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO (n = 0 or 1) port has been read (DCLRM) • NAK setting function for response PID generated by end of transfer (SHTNAK)

Figure 28.1 shows a block diagram of the USB module.

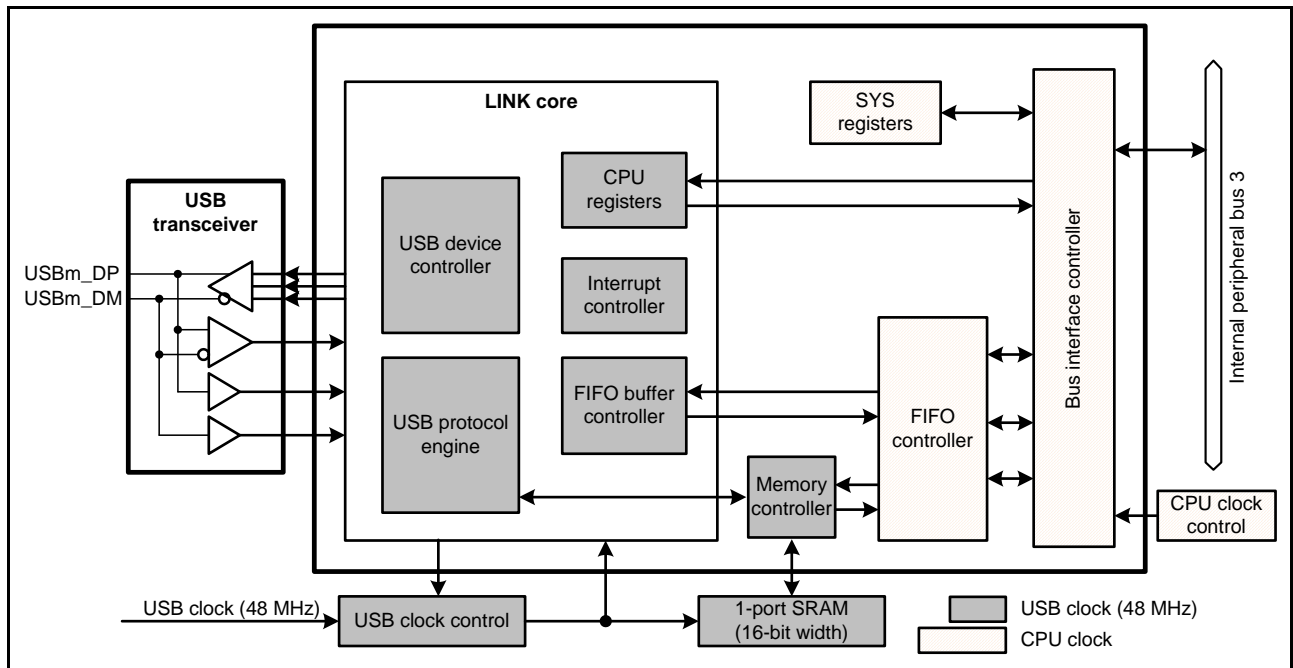


Figure 28.1 Block Diagram of USB

Table 28.2 shows the input/output pins of the USB.

Table 28.2 USB Pin Configuration

Port	Pin Name	I/O	Function
USB0	USB0_DP	I/O	D+ I/O pin of the port 0 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 port 0 USB on-chip transceiver This pin should be connected to the D- pin of the USB bus.
	USB0_VBUS	Input	Port 0 USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. Whether VBUS is connected or disconnected can be detected during operation as a function controller.
	USB0_EXICEN	Output	Low-power control signal for port 0 external power supply (OTG) chip
	USB0_VBUSEN	Output	VBUS (5 V) supply enable signal for port 0 external power supply chip
	USB0_OVRCURA USB0_OVRCURB	Input	Port 0 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	Mini-AB connector ID input signal should be connected to this pin when port 0 operates in OTG mode.
	USB0_DPUPE	Output	1.5-k Ω pull-up resistor control signal* for USB D+ signal when port 0 operates as a function controller
	USB0_DPRPD USB0_DRPD	Output	15-k Ω pull-down resistor control signal* for USB D+ and USB D- signals when port 0 operates as a host controller
	USB1	USB1_DP	I/O
USB1_DM		I/O	D- I/O pin of the port 1 USB on-chip transceiver This pin should be connected to the D- pin of the USB bus.
USB1_VBUS		Input	Port 1 USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. Whether VBUS is connected or disconnected can be detected during operation as a function controller.
USB1_EXICEN		Output	Low-power control signal for port 1 external power supply (OTG) chip
USB1_VBUSEN		Output	VBUS (5 V) supply enable signal for port 1 external power supply chip
USB1_OVRCURA USB1_OVRCURB		Input	Port 1 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.
USB1_ID		Input	Mini-AB connector ID input signal should be connected to this pin when port 1 operates in OTG mode.
USB1_DPUPE		Output	1.5-k Ω pull-up resistor control signal for USB D+ signal when port 1 operates as a function controller
USB1_DPRPD USB1_DRPD		Output	15-k Ω pull-down resistor control signal for USB D+ and USB D- signals when port 1 operates as a host controller
Common		VCC_USB	Input
	VSS_USB	Input	USB ground pin

28.2 Register Descriptions

Table 28.3 lists the registers of the USB.

Table 28.3 Registers of USB (1 / 4)

Port	Register Name	Symbol	Value after Reset	Address	Access Size
USB0	System configuration control register	SYSCFG	0000h	000A 0000h	16
	System configuration status register 0	SYSSTS0	0000h	000A 0004h	16
	Device state control register 0	DVSTCTR0	0000h	000A 0008h	16
	CFIFO port register	CFIFO	0000h	000A 0014h	8, 16
	D0FIFO port register	D0FIFO	0000h	000A 0018h	8, 16
	D1FIFO port register	D1FIFO	0000h	000A 001Ch	8, 16
	CFIFO port select register	CFIFOSEL	0000h	000A 0020h	16
	CFIFO port control register	CFIFOCTR	0000h	000A 0022h	16
	D0FIFO port select register	D0FIFOSEL	0000h	000A 0028h	16
	D0FIFO port control register	D0FIFOCTR	0000h	000A 002Ah	16
	D1FIFO port select register	D1FIFOSEL	0000h	000A 002Ch	16
	D1FIFO port control register	D1FIFOCTR	0000h	000A 002Eh	16
	Interrupt enable register 0	INTENB0	0000h	000A 0030h	16
	Interrupt enable register 1	INTENB1	0000h	000A 0032h	16
	BRDY interrupt enable register	BRDYENB	0000h	000A 0036h	16
	NRDY interrupt enable register	NRDYENB	0000h	000A 0038h	16
	BEMP interrupt enable register	BEMPENB	0000h	000A 003Ah	16
	SOF output configuration register	SOFCFG	0000h	000A 003Ch	16
	Interrupt status register 0	INTSTS0	0000h	000A 0040h	16
	Interrupt status register 1	INTSTS1	0000h	000A 0042h	16
	BRDY interrupt status register	BRDYSTS	0000h	000A 0046	16
	NRDY interrupt status register	NRDYSTS	0000h	000A 0048h	16
	BEMP interrupt status register	BEMPSTS	0000h	000A 004Ah	16
	Frame number register	FRMNUM	0000h	000A 004Ch	16
	Device state change register	DVCHGR	0000h	000A 004Eh	16
	USB address register	USBADDR	0000h	000A 0050h	16
	USB request type register	USBREQ	0000h	000A 0054h	16
	USB request value register	USBVAL	0000h	000A 0056h	16
	USB request index register	USBINDX	0000h	000A 0058h	16
	USB request length register	USBLENG	0000h	000A 005Ah	16
	DCP configuration register	DCPCFG	0000h	000A 005Ch	16
	DCP maximum packet size register	DCPMAXP	0040h *1	000A 005Eh	16
	DCP control register	DCPCTR	0040h *1	000A 0060h	16
	Pipe window select register	PIPESEL	0000h	000A 0064h	16
	Pipe configuration register	PIPECFG	0000h	000A 0068h	16
	Pipe maximum packet size register	PIPEMAXP	0000h/0040h *1,*3	000A 006Ch	16
	Pipe cycle control register	PIPEPERI	0000h	000A 006Eh	16

Table 28.3 Registers of USB (2 / 4)

Port	Register Name	Symbol	Value after Reset	Address	Access Size
USB0	Pipe 1 control register	PIPE1CTR	0000h	000A 0070h	16
	Pipe 2 control register	PIPE2CTR	0000h	000A 0072h	16
	Pipe 3 control register	PIPE3CTR	0000h	000A 0074h	16
	Pipe 4 control register	PIPE4CTR	0000h	000A 0076h	16
	Pipe 5 control register	PIPE5CTR	0000h	000A 0078h	16
	Pipe 6 control register	PIPE6CTR	0000h	000A 007Ah	16
	Pipe 7 control register	PIPE7CTR	0000h	000A 007Ch	16
	Pipe 8 control register	PIPE8CTR	0000h	000A 007Eh	16
	Pipe 9 control register	PIPE9CTR	0000h	000A 0080h	16
	Pipe 1 transaction counter enable register	PIPE1TRE	0000h	000A 0090h	16
	Pipe transaction counter register	PIPE1TRN	0000h	000A 0092h	16
	Pipe 2 transaction counter enable register	PIPE2TRE	0000h	000A 0094h	16
	Pipe 2 transaction counter register	PIPE2TRN	0000h	000A 0096h	16
	Pipe 3 transaction counter enable register	PIPE3TRE	0000h	000A 0098h	16
	Pipe 3 transaction counter register	PIPE3TRN	0000h	000A 009Ah	16
	Pipe 4 transaction counter enable register	PIPE4TRE	0000h	000A 009Ch	16
	Pipe 4 transaction counter register	PIPE4TRN	0000h	000A 009Eh	16
	Pipe 5 transaction counter enable register	PIPE5TRE	0000h	000A 00A0h	16
	Pipe 5 transaction counter register	PIPE5TRN	0000h	000A 00A2h	16
	Device address 0 configuration register	DEVADD0	0000h	000A 00D0h	16
	Device address 1 configuration register	DEVADD1	0000h	000A 00D2h	16
	Device address 2 configuration register	DEVADD2	0000h	000A 00D4h	16
	Device address 3 configuration register	DEVADD3	0000h	000A 00D6h	16
	Device address 4 configuration register	DEVADD4	0000h	000A 00D8h	16
Device address 5 configuration register	DEVADD5	0000h	000A 00DAh	16	
USB1	System configuration control register	SYSCFG	0000h	000A 0200h	16
	System configuration status register 0	SYSSTS0	0000h	000A 0204h	16
	Device state control register 0	DVSTCTR0	0000h	000A 0208h	16
	CFIFO port register	CFIFO	0000h	000A 0214h	8, 16
	D0FIFO port register	D0FIFO	0000h	000A 0218h	8, 16
	D1FIFO port register	D1FIFO	0000h	000A 021Ch	8, 16
	CFIFO port select register	CFIFOSEL	0000h	000A 0220h	16
	CFIFO port control register	CFIFOCTR	0000h	000A 0222h	16
	D0FIFO port select register	D0FIFOSEL	0000h	000A 0228h	16
	D0FIFO port control register	D0FIFOCTR	0000h	000A 022Ah	16
	D1FIFO port select register	D1FIFOSEL	0000h	000A 022Ch	16
	D1FIFO port control register	D1FIFOCTR	0000h	000A 022Eh	16
	Interrupt enable register 0	INTENB0	0000h	000A 0230h	16
	Interrupt enable register 1	INTENB1	0000h	000A 0232h	16
	BRDY interrupt enable register	BRDYENB	0000h	000A 0236h	16
	NRDY interrupt enable register	NRDYENB	0000h	000A 0238h	16
BEMP interrupt enable register	BEMPENB	0000h	000A 023Ah	16	

Table 28.3 Registers of USB (3 / 4)

Port	Register Name	Symbol	Value after Reset	Address	Access Size
USB1	SOF output configuration register	SOFCFG	0000h	000A 023Ch	16
	Interrupt status register 0	INTSTS0	0000h	000A 0240h	16
	Interrupt status register 1	INTSTS1	0000h	000A 0242h	16
	BRDY interrupt status register	BRDYSTS	0000h	000A 0246	16
	NRDY interrupt status register	NRDYSTS	0000h	000A 0248h	16
	BEMP interrupt status register	BEMPSTS	0000h	000A 024Ah	16
	Frame number register	FRMNUM	0000h	000A 024Ch	16
	Device state change register	DVCHGR	0000h	000A 024Eh	16
	USB address register	USBADDR	0000h	000A 0250h	16
	USB request type register	USBREQ	0000h	000A 0254h	16
	USB request value register	USBVAL	0000h	000A 0256h	16
	USB request index register	USBINDX	0000h	000A 0258h	16
	USB request length register	USBLENG	0000h	000A 025Ah	16
	DCP configuration register	DCPCFG	0000h	000A 025Ch	16
	DCP maximum packet size register	DCPMAXP	0040h*2	000A 025Eh	16
	DCP control register	DCPCTR	0040h*2	000A 0260h	16
	Pipe window select register	PIPESEL	0000h	000A 0264h	16
	Pipe configuration register	PIPECFG	0000h	000A 0268h	16
	Pipe maximum packet size register	PIPEMAXP	0000h/0040h*2.*3	000A 026Ch	16
	Pipe cycle control register	PIPEPERI	0000h	000A 026Eh	16
	Pipe 1 control register	PIPE1CTR	0000h	000A 0270h	16
	Pipe 2 control register	PIPE2CTR	0000h	000A 0272h	16
	Pipe 3 control register	PIPE3CTR	0000h	000A 0274h	16
	Pipe 4 control register	PIPE4CTR	0000h	000A 0276h	16
	Pipe 5 control register	PIPE5CTR	0000h	000A 0278h	16
	Pipe 6 control register	PIPE6CTR	0000h	000A 027Ah	16
	Pipe 7 control register	PIPE7CTR	0000h	000A 027Ch	16
	Pipe 8 control register	PIPE8CTR	0000h	000A 027Eh	16
	Pipe 9 control register	PIPE9CTR	0000h	000A 0280h	16
	Pipe 1 transaction counter enable register	PIPE1TRE	0000h	000A 0290h	16
	Pipe 1 transaction counter register	PIPE1TRN	0000h	000A 0292h	16
	Pipe 2 transaction counter enable register	PIPE2TRE	0000h	000A 0294h	16
	Pipe 2 transaction counter register	PIPE2TRN	0000h	000A 0296h	16
	Pipe 3 transaction counter enable register	PIPE3TRE	0000h	000A 0298h	16
	Pipe 3 transaction counter register	PIPE3TRN	0000h	000A 029Ah	16
	Pipe 4 transaction counter enable register	PIPE4TRE	0000h	000A 029Ch	16
	Pipe 4 transaction counter register	PIPE4TRN	0000h	000A 029Eh	16
	Pipe 5 transaction counter enable register	PIPE5TRE	0000h	000A 02A0h	16
	Pipe 5 transaction counter register	PIPE5TRN	0000h	000A 02A2h	16
	Device address 0 configuration register	DEVADD0	0000h	000A 02D0h	16
Device address 1 configuration register	DEVADD1	0000h	000A 02D2h	16	
Device address 2 configuration register	DEVADD2	0000h	000A 02D4h	16	

Table 28.3 Registers of USB (4 / 4)

Port	Register Name	Symbol	Value after Reset	Address	Access Size
USB1	Device address 3 configuration register	DEVADD3	0000h	000A 02D6h	16
	Device address 4 configuration register	DEVADD4	0000h	000A 02D8h	16
	Device address 5 configuration register	DEVADD5	0000h	000A 02DAh	16
Common	Deep standby USB transceiver control/pin monitor register	DPUSR0R	xxxx 0000h	000A 0400h	32
	Deep standby USB suspend/resume interrupt register	DPUSR1R	0000 0000h	000A 0404h	32

Note 1. This value is when the usage of USB is enabled (the MSTPCRB.MSTPB19 bit is 0) and the USB0.SYSCFG.SCKE bit is 1.

Note 2. This value is when the usage of USB is enabled (the MSTPCRB.MSTPB18 bit is 0) and the USB1.SYSCFG.SCKE bit is 1.

Note 3. The initial value of this register differs according to the setting of the PIPESEL.PIPESEL[3:0] bits. The initial value is 0000h when the pipe is not selected and 0040h when selected.

28.2.1 System Configuration Control Register (SYSCFG)

Addresses: USB0.SYSCFG 000A 0000h, USB1.SYSCFG 000A 0200h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKE	—	—	—	DCFM	DRPD	DPRPU	—	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	USBE	USB Module Operation Enable	0: USB module operation is disabled. 1: USB module operation is enabled.	R/W
b3 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	DPRPU	D+ Line Resistor Control	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
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 function is selected. 1: Host controller function is selected.	R/W
b9 to b7	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b10	SCKE	USB Module Clock Enable	0: Stops supplying the clock signal to the USB module. 1: Enables supplying the clock signal to the USB module.	R/W
b15 to b11	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

SYSCFG selects the host controller function or function controller function, controls the USBm_DP and USBm_DM pins, and enables operation of the USB module. (m = 0, 1)

USBE Bit (USB Module Operation Enable)

The USBE bit enables or disables operation of the USB module.

Modifying the USBE bit from 1 to 0 initializes some register bits as listed in Table 28.4 and Table 28.5.

This bit should be modified while the SYSCFG.SCKE bit is 1.

When the host controller function is selected, this bit should be set to 1 after setting the SYSCFG.DRPD bit to 1, eliminating LNST bit chattering, and checking that the USB bus state has been settled.

DPRPU Bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line when the function controller function is selected.

When the DPRPU bit is set to 1 while the function controller function is selected, the USB module asserts the USBm_DPUPE pin to notify the USB host of connection.

Setting the DPRPU bit to 1 when the function controller function is selected allows the USB module to assert the USBm_DPUPE pin, thus notifying the USB host of connection. Modifying the DPRPU bit from 1 to 0 allows the USB module to negate the USBm_DPUPE pin, thus notifying the USB host of disconnection.

This bit should be set to 1 if the function controller function is selected, and should be set to 0 if the host controller function is selected.

DRPD Bit (D+/D- Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D- lines by the USBm_DRPDP and USBm_DRPD pins when the host controller function is selected.

This bit should be set to 1 if the host controller function is selected, and should be set to 0 if the function controller function is selected.

DCFM Bit (Controller Function Select)

The DCFM bit selects the function of the USB module.

This bit should be modified while both the SYSCFG.DPRPU and SYSCFG.DRPD bits are 0.

SCKE Bit (USB Module Clock Enable)

The SCKE bit stops or enables supplying 48-MHz clock signals to the USB module.

When this bit is 0, only SYSCFG, DMA0PCFG, and DMA1PCFG can be read from and written to; the other registers in the USB module cannot be read from or written to.

Table 28.4 Registers Initialized by Writing SYSCFG.USBE = 0 (When Function Controller Function is Selected)

Register	Symbol	Remarks
SYSSTS0	LNST	The value is retained when the host controller function is selected.
DVSTCTR0	RHST	
INTSTS0	DVSQ	The value is retained when the host controller function is selected.
USBADDR	USBADDR	The value is retained when the host controller function is selected.
USBREQ	BREQUEST, BMREQUESTTYPE	The value is retained when the host controller function is selected.
USBVAL	WVALUE	The value is retained when the host controller function is selected.
USBINDX	WINDEX	The value is retained when the host controller function is selected.
USBLENG	WLENGTH	The value is retained when the host controller function is selected.

Table 28.5 Registers Initialized by Writing SYSCFG.USBE = 0 (When Host Controller Function is Selected)

Register	Symbol	Remarks
DVSTCTR0	RHST	
FRMNUM	FRNM	The value is retained when the function controller function is selected.

28.2.2 System Configuration Status Register 0 (SYSSTS0)

Addresses: USB0.SYSSTS0 000A 0004h, USB1.SYSSTS0 000A 0204h

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*	0*	0	0	0	0	0	0	0	0	0	0	0*	0	0

Note: * Depends on the USBm_OVRCURA/USBm_OVRCURB and USBm_ID pin status.

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor	Indicates the status of the USB data bus lines (D+ and D-) as shown in Table 28.6.	R
b2	IDMON	External ID0 Input Pin Monitor	Indicates the status of the USBm_ID pin.	R
b5 to b3	—	Reserved	These bits are always read as 0 and cannot be modified.	R
b6	HTACT	USB Host Sequencer Status Monitor	0: Host sequencer of the USB module is completely stopped. 1: Host sequencer of the USB module is not completely stopped.	R
b13 to b7	—	Reserved	These bits are always read as 0 and cannot be modified.	R
b15, b14	OVCMON [1:0]	External USBm_OVRCURA/ USBm_OVRCURB Input Pin Monitor	The OVCMON[1] bit indicates the status of the USBm_OVRCURA pin. The OVCMON[0] bit indicates the status of the USBm_OVRCURB pin.	R

[Legend] m = 0, 1

SYSSTS0 monitors the line status (D+ and D- lines) of the USB data bus.

LNST[1:0] Bits (USB Data Line Status Monitor)

The LNST[1:0] bits should be read after the connection processing (SYSCFG.DPRPU = 1 is set) when the function controller function is selected; whereas after enabling pull-down of the lines (SYSCFG.DRPD = 1 is set) when the host controller function is selected.

Table 28.6 USB Data Bus Line Status

LNST[1]	LNST[0]	Status
0	0	SE0
0	1	J-state
1	0	K-state
1	1	SE1

HTACT Bit (USB Host Sequencer Status Monitor)

The HTACT bit is 0 when the host sequencer of the USB module is completely stopped.

Make sure the HTACT bit is 0 when stopping the clock supply to the USB module.

OVCMON[1:0] Bits (External USBm_OVRCURA/USBm_OVRCURB Input Pin Monitor)

The OCVMON[1:0] bits indicate the status of overcurrent from an external power-supply chip.

28.2.3 Device State Control Register 0 (DVSTCTR0)

Addresses: USB0.DVSTCTR0 000A 0008h, USB1.DVSTCTR0 000A 0208h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	—	RHST[2:0]	
Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status	<ul style="list-style-type: none"> When the host controller function is selected <ul style="list-style-type: none"> b2 b1 b0 0 0 0: Communication speed not determined (powered state or no connection) 1 x x: USB bus reset in progress 0 0 1: Low-speed connection*1 0 1 0: Full-speed connection [Legend] x: Don't care <ul style="list-style-type: none"> When the function controller function is selected <ul style="list-style-type: none"> b2 b1 b0 0 0 0: Communication speed not determined 1 0 0: USB bus reset in progress 0 1 0: Full-speed connection 	R
b3	—	Reserved	This bit is always read as 0. The write value should always 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 *2
b9	VBUSEN	USBm_VBUSEN Output Pin Control	The VBUSEN bit value is output as the status of the external USBm_VBUSEN pin without change.	R/W
b10	EXICEN	USBm_EXICEN Output Pin Control	The EXICEN bit value is output as the status of the external USBm_EXICEN pin without change.	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	This bit is used when switching from device B to device A while in OTG (On-The-Go) 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 is set.	R/W
b15 to b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. The USB controller does not support communication with a low-speed device. When this value is read, abnormal connection processing should be executed in a higher application.

Note 2. Only 1 can be written.

DVSTCTR0 controls and confirms the state of the USB data bus.

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 function is selected, the RHST[2:0] bits indicate 100b after software has written 1 to the USBRST bit.

The USB module fixes the value of the RHST[2:0] bits when software writes 0 to the USBRST bit and the USB module completes SE0 driving.

When the function controller function is selected, a DVST interrupt is generated as soon as the USB module detects the USB bus reset and then the RHST[2:0] bits are fixed to 010b.

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 function is selected.

With this bit set to 1, the USB module 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 software has written 1 to the UACT bit.

With this bit set to 0, the USB module enters the idle state after outputting SOF packets.

The USB module 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 function is selected.

RESUME Bit (Resume Output)

The RESUME bit controls the resume signal output when the host controller function is selected.

Setting the RESUME bit to 1 allows the USB module to drive the port to the K-state and output the resume signal.

The USB module continues outputting K-state while RESUME = 1 (until software sets the RESUME bit to 0). The RESUME bit should be 1 (= resume period) for the time defined by the USB Specifications 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 function is selected.

USBRST Bit (USB Bus Reset Output)

The USBRST bit controls the USB bus reset signal output when the host controller function is selected.

When the host controller function is selected, setting this bit to 1 allows the USB module to drive SE0 of the USB port to reset the USB bus.

The USB module continues outputting SE0 while USBRST = 1 (until software sets the USBRST bit to 0). The USBRST bit should be 1 (= USB bus reset period) for the time defined by the USB Specifications 2.0.

Writing 1 to this bit during communication (UACT = 1) or during the resume processing (RESUME = 1) prevents the USB module 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 function 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 function is selected.

With this bit set to 1, on detecting the remote wakeup signal, the USB module detects the resume signal (K-state for 2.5 μ 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 module ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port.

While the PWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SCKE should be set to 1).

This bit should be set to 0 if the function controller function 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 function is selected.

The USB module controls the output time of a remote wakeup signal. When this bit is set to 1, the USB module clears this bit to 0 after outputting the 10-ms K-state.

According to the USB Specifications, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USB module writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms.

Do not write 1 to this bit, unless the device state is in the suspended state (INTSTS0.DVSQ bit = 1xx) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while SYSCFG.SCKE = 1).

This bit should be set to 0 if the host controller function 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 (On-The-Go) 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 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 at FW to terminate the HNP processing when connection to the host (pull-up on the target side) or timeout of the HNP processing is detected.

28.2.4 CFIFO Port Register (CFIFO)
 D0FIFO Port Register (D0FIFO)
 D1FIFO Port Register (D1FIFO)

Addresses: USB0.CFIFO 000A 0014h, USB0.D0FIFO 000A 0018h, USB0.D1FIFO 000A 001Ch
 USB1.CFIFO 000A 0214h, USB1.D0FIFO 000A 0218h, USB1.D1FIFO 000A 021Ch



Bit	Symbol	Bit Name	Description	R/W
b15 to b8	L[7:0]	FIFO Port	The valid bits in a FIFO port register depend on the settings of the corresponding MBW and BIGEND bits as shown in Table 28.7 and Table 28.8	R/W
b7 to b0	H[7:0]			

CFIFO, D0FIFO, and D1FIFO are port registers that are used to read data from the FIFO buffer memory and write data to the FIFO buffer memory.

There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO ports. Each FIFO port is configured of a FIFO port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).

Each FIFO port has the following features.

- The FIFO buffer for DCP (control transfer) should be accessed through the CFIFO port.
- Accessing the FIFO buffer using DMA transfer should be performed through the D0FIFO or D1FIFO port.
- The D1FIFO and D0FIFO ports can be accessed also by the CPU.
- When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed (when the DMA transfer function is used, etc.).
- Registers configuring a FIFO port do not affect other FIFO ports.
- The same pipe should not be assigned to two or more FIFO ports.
- There are two FIFO buffer states: the access right is on the CPU side and it is on the SIE side. When the FIFO buffer access right is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

L[7:0]/H[7:0] (FIFO Port)

Accessing the FIFOPORT bits allow 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 control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in a FIFO port register depend on the settings of the corresponding MBW and BIGEND bits as shown in Table 28.7 and Table 28.8.

In 8-bit access, this bit should be accessed in bytes. The BIGEND bit setting is disabled.

In 16-bit access, this bit should be accessed in words. If a total number of data is odd, the last data should be accessed in bytes.

In both 8-bit access and 16-bit access, the FIFO register start address should be accessed regardless of settings.

Table 28.7 Endian Operation in 16-Bit Access

CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

Table 28.8 Endian Operation in 8-Bit Access

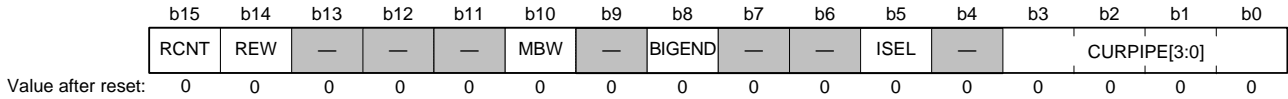
CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0
x (Setting is invalid)	Access prohibited*	N + 0 data

Note: * Accessing an access-prohibited area to read data is not allowed.

28.2.5 CFIFO Port Select Register (CFIFOSEL)
 D0FIFO Port Select Register (D0FIFOSEL)
 D1FIFO Port Select Register (D1FIFOSEL)

- CFIFOSEL

Addresses: USB0.CFIFOSEL 000A 0020h, USB1.CFIFOSEL 000A 0220h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE[3:0]	CFIFO Port Access Pipe Specification	b3 b2 b1 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Other than above: Setting prohibited	R/W
b4	—	Reserved	This bit is always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W
b8	BIGEND	CFIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W*
b15	RCNT	Read Count Mode	0: When all of the receive data has been read from the CFIFO, the DTLN bit value is cleared. (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 bit is decremented each time the receive data is read from the CFIFO.	R/W

Note : * Only 0 can be read.

CFIFOSEL assigns the pipe to the FIFO port, and controls access to the corresponding FIFO port.

The same pipe should not be specified by the CURPIPE bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL. When the CURPIPE bits in D0FIFOSEL and D1FIFOSEL are cleared to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA transfer is enabled.

CURPIPE[3:0] Bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the CFIFO port.

After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.

Do not set the same pipe number to the CURPIPE 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 bits simultaneously.

BIGEND Bit (CFIFO Port Endian Control)

The BIGEND bit specifies the byte endian for the CFIFO port.

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 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 bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1.

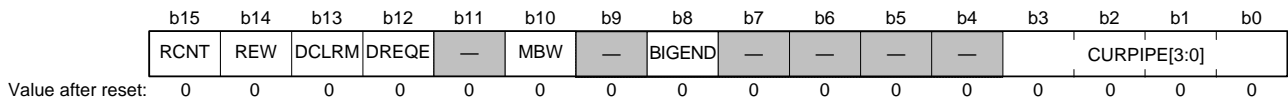
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT Bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the DTLN[8:0] bits in CFIFOCTR.

• D0FIFOSEL, D1FIFOSEL

Addresses: USB0.D0FIFOSEL 000A 0028h, USB0.D1FIFOSEL 000A 002Ch
 USB1.D0FIFOSEL 000A 0228h, USB1.D1FIFOSEL 000A 022Ch



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE[3:0]	FIFO Port Access Pipe Specification	b3 b2 b1 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Other than above: Setting prohibited	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W
b12	DREQE	DMA Transfer Request Enable	0: DMA transfer request is disabled. 1: DMA transfer request is enabled.	R/W
b13	DCLRM	Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read	0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W*
b15	RCNT	Read Count Mode	0: When all of the receive data has been read from the DnFIFO, the DTLN bit value is cleared. (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 bit is decremented each time the receive data is read from the DnFIFO. (n = 0, 1)	R/W

Note : * Only 0 can be read.

D0FIFOSEL and D1FIFOSEL assign the pipe to the FIFO port, and control access to the corresponding FIFO port. The same pipe should not be specified by the CURPIPE bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL. When the CURPIPE bits in D0FIFOSEL and D1FIFOSEL are cleared to 0000b, no pipe is selected. The pipe number should not be changed while the DMA transfer is enabled.

CURPIPE[3:0] Bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the D0FIFO port or D1FIFO port.

After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.

Do not set the same pipe number to the CURPIPE 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.

BIGEND Bit (FIFO Port Endian Control)

The BIGEND bit specifies the byte endian for the D0FIFO port or D1FIFO port.

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 and MBW bits simultaneously.

When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

DREQE Bit (DMA Transfer Request Enable)

The DREQE bit enables or disables the DMA transfer request to be issued.

Before setting the DREQE bit to 1 to enable the DMA transfer request to be issued, set the CURPIPE[3:0] bits.

When modifying the setting of the CURPIPE[3:0] bits, set this bit to 0 first.

DCLRM Bit (Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read)

The DCLRM bit enables or disables the buffer memory to be cleared automatically after data has been read out using the selected pipe.

With this bit set to 1, the USB module sets the BCLR bit to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while the PIPECFG.BFRE bit is 1.

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 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.

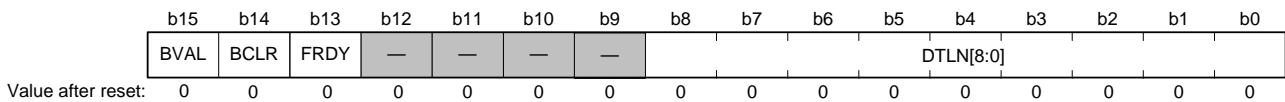
RCNT Bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the DTLN[8:0] bits in DnFIFOCTR.

When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

28.2.6 CFIFO Port Control Register (CFIFOCTR) D0FIFO Port Control Register (D0FIFOCTR) D1FIFO Port Control Register (D1FIFOCTR)

Addresses: USB0.CFIFOCTR 000A 0022h, USB0.D0FIFOCTR 000A 002Ah, USB0.D1FIFOCTR 000A 002Eh
USB1.CFIFOCTR 000A 0222h, USB1.D0FIFOCTR 000A 022Ah, USB1.D1FIFOCTR 000A 022Eh



Bit	Symbol	Bit Name	Description	R/W
b8 to b0	DTLN[8:0]	Receive Data Length	Indicates the length of the receive data. These bits indicate different values depending on the setting of the RCNT bit in the port select register. For details, refer to the description on the DTLN[8:0] bits shown below.	R
b12 to b9	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b13	FRDY	FIFO Port Ready	0: FIFO port access is disabled. 1: FIFO port access is enabled.	R
b14	BCLR	CPU Buffer Clear	0: Invalid 1: Clears the buffer memory on the CPU side.	R/W*1
b15	BVAL	Buffer Memory Valid Flag	0: Invalid 1: Writing ended	R/W*2

Note 1. Only 0 can be read and 1 can be written.

Note 2. Only 1 can be written.

CFIFOCTR, D0FIFOCTR, and D1FIFOCTR determine whether or not writing to the buffer memory has been finished, the buffer accessed from the CPU has been cleared, and the FIFO port is accessible. The FIFO port control registers are used for the corresponding FIFO ports.

DTLN[8:0] Bits (Receive Data Length)

The DTLN[8:0] bits indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the DnFIFOSEL.RCNT (n = 0, 1) bit value as described below.

- RCNT = 0
The USB module sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU (DTC or DMACA) has read all the received data from a single FIFO buffer plane.
While PIPECFG.BFRE = 1, these bits retain the length of the receive data until the BCLR bit is set to 1 even after all the data has been read.
- RCNT = 1
The USB module decrements the value indicated by these bits each time data is read from the FIFO buffer. (The value is decremented by one when the MBW bit is 0, and by two when the MBW bit is 1.)

The USB module sets 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 module 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 (DTC or DMACA).

In the following cases, the USB module sets the FRDY bit to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
- A short packet is received and the data is completely read while PIPECFG.BFRE = 1.

BCLR Bit (CPU Buffer Clear)

The BCLR bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB module clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the selected pipe is the DCP, setting the BCLR bit to 1 allows the USB module to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the PID bits for the DCP control register to NAK before setting the BCLR bit to 1.

When the selected pipe is in the transmitting direction, if 1 is written to the BVAL and BCLR bits simultaneously, the USB module clears the data that has been written before it, enabling transmission of a zero-length packet.

When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while the FRDY bit in the FIFO port control register is 1 (set by the USB module).

BVAL Bit (Buffer Memory Valid Flag)

The BVAL bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE bits (selected pipe).

When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB module switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

- To transmit a short packet, set the BVAL bit to 1 after data has been written.
- To transmit a zero-length packet, set the BVAL bit to 1 before data is written to the FIFO buffer.

When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB module sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

Writing 1 to the BVAL bit should be done while the FRDY bit is 1 (set by the USB module).

When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.

28.2.7 Interrupt Enable Register 0 (INTENB0)

Addresses: USB0.INTENB0 000A 0030h, USB1.INTENB0 000A 0230h

	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 always read as 0. The write value should always be 0.	R/W
b8	BRDYE	Buffer Ready Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10	BEMPE	Buffer Empty Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Enable*	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DVSE	Device State Transition Interrupt Enable*	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	SOFE	Frame Number Update Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b14	RSME	Resume Interrupt Enable*	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	VBSE	VBUS Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note : * The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller function is selected; do not set these bits to 1 to enable the corresponding interrupt output when the host controller function is selected.

INTENB0 specifies the various interrupt masks. On detecting the interrupt corresponding to the bit in INTENB0 to which software has set 1, the USB module generates the USB interrupt.

The USB module sets 1 to each status bit in INTSTS0 when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in INTENB0 (regardless of whether the interrupt output is enabled or disabled).

While the status bit in INTSTS0 corresponding to the interrupt source indicates 1, the USB module generates the USB interrupt when software modifies the corresponding interrupt enable bit in INTENB0 from 0 to 1.

BRDYE Bit (Buffer Ready Interrupt Enable)

Enables or disables the USB interrupt output when the BRDY interrupt is detected.

NRDYE Bit (Buffer Not Ready Response Interrupt Enable)

Enables or disables the USB interrupt output when the NRDY interrupt is detected.

BEMPE Bit (Buffer Empty Interrupt Enable)

Enables or disables the USB interrupt output when the BEMP interrupt is detected.

CTRE Bit (Control Transfer Stage Transition Interrupt Enable)

Enables or disables the USB interrupt output when the CTRT interrupt is detected.

DVSE Bit (Device State Transition Interrupt Enable)

Enables or disables the USB interrupt output when the DVST interrupt is detected.

SOFSE Bit (Frame Number Update Interrupt Enable)

Enables or disables the USB interrupt output when the SOFR interrupt is detected.

RSME Bit (Resume Interrupt Enable)

Enables or disables the USB interrupt output when the RESM interrupt is detected.

VBSE Bit (VBUS Interrupt Enable)

Enables or disables the USB interrupt output when the VBINT interrupt is detected.

28.2.8 Interrupt Enable Register 1 (INTENB1)

Addresses: USB0.INTENB1 000A 0032h, USB1.INTENB1 000A 0232h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	—
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	—	Reserved	These bits are always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W
b11	ATTCHE	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 always read as 0. The write value should always 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 function is selected; do not set these bits to 1 to enable the corresponding interrupt output when the function controller function is selected.

INTENB1 specifies the various interrupt masks when the host controller function is selected. INTENB1 also specifies the interrupt mask for the setup transaction.

On detecting the interrupt corresponding to the bit in INTENB1 to which software has set 1, the USB module generates the USB interrupt.

The USB module 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 module generates the USB interrupt when software modifies the corresponding interrupt enable bit in INTENB1 from 0 to 1.

If operation as a USB-function controller has been selected, do not make any of the interrupt-enabling settings in INTENB1.

SACKE Bit (Setup Transaction Normal Response Interrupt Enable)

Enables or disables the USB interrupt output when the SACK interrupt is detected.

SIGNE Bit (Setup Transaction Error Interrupt Enable)

Enables or disables the USB interrupt output when the SIGN interrupt is detected.

EOFERRE Bit (EOF Error Detection Interrupt Enable)

Enables or disables the USB interrupt output when the EOFERR interrupt is detected.

ATTCHE Bit (Connection Detection Interrupt Enable)

Enables or disables the USB interrupt output when the ATTCH interrupt is detected.

DTCHE Bit (Disconnection Detection Interrupt Enable)

Enables or disables the USB interrupt output when the DTCH interrupt is detected.

BCHGE Bit (USB Bus Change Interrupt Enable)

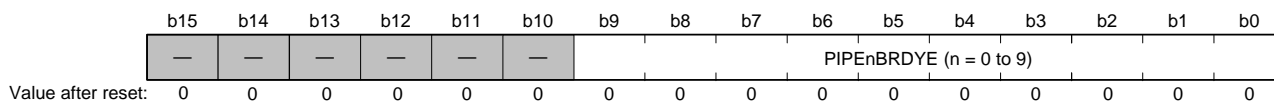
Enables or disables the USB interrupt output when the BCHG interrupt is detected.

OVRCRE Bit (Overcurrent Input Change Interrupt Enable)

Enables or disables the USB interrupt output when the OVRCCR interrupt is detected.

28.2.9 BRDY Interrupt Enable Register (BRDYENB)

Addresses: USB0.BRDYENB 000A 0036h, USB1.BRDYENB 000A 0236h



Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0 BRDYE	BRDY Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1 BRDYE	BRDY Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2 BRDYE	BRDY Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3 BRDYE	BRDY Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4 BRDYE	BRDY Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5 BRDYE	BRDY Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6 BRDYE	BRDY Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7 BRDYE	BRDY Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8 BRDYE	BRDY Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9 BRDYE	BRDY Interrupt Enable for PIPE9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

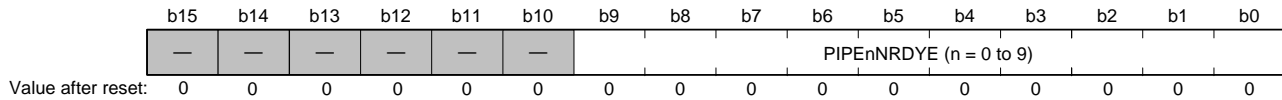
BRDYENB enables or disables the BRDY bit in INTSTS0 to be set to 1 when the BRDY interrupt is detected for each pipe.

On detecting the BRDY interrupt for the pipe corresponding to the bit in BRDYENB to which software has set 1, the USB module sets 1 to the corresponding PIPEBRDY bit in BRDYSTS and the BRDY bit in INTSTS0, and generates the BRDY interrupt.

While at least one PIPEBRDY bit in BRDYSTS indicates 1, the USB module generates the BRDY interrupt when software modifies the corresponding interrupt enable bit in BRDYENB from 0 to 1.

28.2.10 NRDY Interrupt Enable Register (NRDYENB)

Addresses: USB0.NRDYENB 000A 0038h, USB1.NRDYENB 000A 0238h



Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0 NRDYE	NRDY Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1 NRDYE	NRDY Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2 NRDYE	NRDY Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3 NRDYE	NRDY Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4 NRDYE	NRDY Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5 NRDYE	NRDY Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6 NRDYE	NRDY Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7 NRDYE	NRDY Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8 NRDYE	NRDY Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9 NRDYE	NRDY Interrupt Enable for PIPE9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

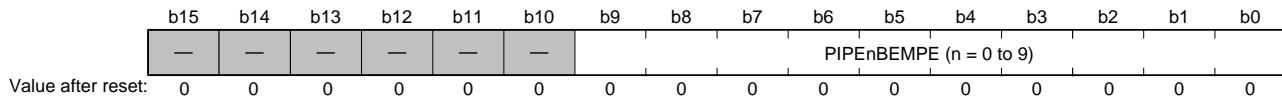
NRDYENB enables or disables the NRDY bit in INTSTS0 to be set to 1 when the NRDY interrupt is detected for each pipe.

On detecting the NRDY interrupt for the pipe corresponding to the bit in NRDYENB to which software has set 1, the USB module sets 1 to the corresponding PIPENRDY bit in NRDYSTS and the NRDY bit in INTSTS0, and generates the NRDY interrupt.

While at least one PIPENRDY bit in NRDYSTS indicates 1, the USB module generates the NRDY interrupt when software modifies the corresponding interrupt enable bit in NRDYENB from 0 to 1.

28.2.11 BEMP Interrupt Enable Register (BEMPENB)

Addresses: USB0.BEMPENB 000A 003Ah, USB1.BEMPENB 000A 023Ah



Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0 BEMPE	BEMP Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1 BEMPE	BEMP Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2 BEMPE	BEMP Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3 BEMPE	BEMP Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4 BEMPE	BEMP Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5 BEMPE	BEMP Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6 BEMPE	BEMP Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7 BEMPE	BEMP Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8 BEMPE	BEMP Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9 BEMPE	BEMP Interrupt Enable for PIPE9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

BEMPENB enables or disables the BEMP bit in INTSTS0 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 software has set 1, the USB module sets 1 to the corresponding PIPEBEMP bit in BEMPSTS and the BEMP bit in INTSTS0, and generates the BEMP interrupt.

While at least one PIPEBEMP bit in BEMPSTS indicates 1, the USB module generates the BEMP interrupt when software modifies the corresponding interrupt enable bit in BEMPENB from 0 to 1.

28.2.12 SOF Output Configuration Register (SOFCFG)

Addresses: USB0.SOFCFG 000A 003Ch, USB1.SOFCFG 000A 023Ch

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	EDGESTS	—	—	—	—
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	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	EDGESTS* 1	Edge Interrupt Output Status Monitor	Indicates 1 when the edge interrupt output signal is in the middle of the edge processing.	R
b5	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b6	BRDYM*2	BRDY Interrupt Status Clear Timing for each Pipe	0: Software clears the status. 1: The USB module clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.	R/W
b7	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select	0: For non-low-speed communication 1: Setting prohibited	R/W
b15 to b9	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. This bit is unnecessary in a system which does not use the edge interrupt output signal.

Note 2. Make sure to set the BRDYM bit to 0.

SOFCFG specifies the transaction-enabled time and BRDY interrupt status clear timing.

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.

Note: Make sure the EDGESTS bit is 0 when stopping the clock supply to the USB module.

BRDYM Bit (BRDY Interrupt Status Clear Timing for each Pipe)

The BRDYM bit specifies the timing for clearing the BRDY interrupt status for each pipe.

Make sure to set the BRDYM bit to 0.

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 module issues tokens in a frame via the port.

The TRNENSEL bit is valid only when the host controller function is selected.

This bit should be set to 0 if the function controller function is selected.

28.2.13 Interrupt Status Register 0 (INTSTS0)

Addresses: USB0.INTSTS0 000A 0040h, USB1.INTSTS0 000A 0240h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
VBINT	RESM	SOFR	DVST	CTRT	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 ³ 0 0 0/1 ² 0 0 0 0															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage	b2 b1 b0 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error 1 1 1: Setting prohibited	R
b3	VALID	USB Request Reception	0: Not detected 1: Setup packet reception	R/W*4
b6 to b4	DVSQ[2:0]	Device State	b6 b5 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspended state [Legend] x: Don't care	R
b7	VBSTS	VBUS Input Status	0: USBm_VBUS pin is low. 1: USBm_VBUS pin is high.	R
b8	BRDY	Buffer Ready Interrupt Status	0: BRDY interrupts are not generated. 1: BRDY interrupts are generated.	R
b9	NRDY	Buffer Not Ready Interrupt Status	0: NRDY interrupts are not generated. 1: NRDY interrupts are generated.	R
b10	BEMP	Buffer Empty Interrupt Status	0: BEMP interrupts are not generated. 1: BEMP interrupts are generated.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status*6	0: Control transfer stage transition interrupts are not generated. 1: Control transfer stage transition interrupts are generated.	R/W*4
b12	DVST	Device State Transition Interrupt Status*6	0: Device state transition interrupts are not generated. 1: Device state transition interrupts are generated.	R/W*4
b13	SOFR	Frame Number Refresh Interrupt Status	0: SOF interrupts are not generated. 1: SOF interrupts are generated. (1) When the host controller function is selected The USB module sets the SOFR bit to 1 on updating the frame number when software has set the UACT bit to 1. (A SOFR interrupt is detected every 1 ms.) (2) When the function controller function is selected The USB module sets the SOFR bit to 1 on updating the frame number. (A SOFR interrupt is detected every 1 ms.) The USB module can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.	R/W*4
b14	RESM	Resume Interrupt Status*5*6	0: Resume interrupts are not generated. 1: Resume interrupts are generated.	R/W*4
b15	VBINT	VBUS Interrupt Status*5	0: VBUS interrupts are not generated. 1: VBUS interrupts are generated.	R/W*4

- Note 1. This bit is initialized to 0b by a power-on reset and 1b by a USB bus reset.
- Note 2. These bits are initialized to 000b by a power-on reset and 001b by a USB bus reset.
- Note 3. This bit is initialized to 1 when the level of the USBm_VBUS pin input is high and 1 when low.
- Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bit, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.
- Note 5. A change in the status indicated by the VBINT and RESM bits can be detected even while the clock supply is stopped (SCKE = 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.
- Note 6. A change in the status of the RESM, DVST, and CTRT bits occur only when the host controller function is selected; disable the corresponding interrupt enable bits (set to 0) when the function controller function is selected.

INTSTS0 indicates the status of the various interrupts detected.

The DVSQ[2:0] bits are initialized by a USB bus reset.

CTSQ[2:0] Bits (Control Transfer Stage)

When the host controller function is selected, the read value is invalid.

VALID Bit (USB Request Reception)

When the host controller function is selected, the read value is invalid.

DVSQ[2:0] Bits (Device State)

When the host controller function is selected, the read value is invalid.

BRDY Bit (Buffer Ready Interrupt Status)

Indicates the BRDY interrupt status.

The USB module sets the BRDY bit to 1 when at least one PIPEBRDY bit in BRDYSTS is set to 1 among the PIPEBRDY bits corresponding to the PIPEBRDYE bits in BRDYENB to which 1 has been set (when the USB module detects the BRDY interrupt status in at least one pipe among the pipes for which software enables the BRDY interrupt output).

For the conditions for PIPEBRDY status assertion, refer to section 28.3.3.1, BRDY Interrupt.

The USB module clears the BRDY bit to 0 when software writes 0 to all the PIPEBRDY bits corresponding to the PIPEBRDYE bits to which 1 has been set.

The BRDY bit cannot be cleared to 0 even if software writes 0 to this bit.

NRDY Bit (Buffer Not Ready Interrupt Status)

The USB module sets the NRDY bit to 1 when at least one PIPENRDY bit in NRDYSTS is set to 1 among the PIPENRDY bits corresponding to the PIPENRDYE bits in NRDYENB to which 1 has been set (when the USB module 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 PIPENRDY status assertion, refer to section 28.3.3.2, NRDY Interrupt.

The USB module clears the NRDY bit to 0 when software writes 0 to all the PIPENRDY bits corresponding to the PIPENRDYE bits to which 1 has been set.

The NRDY bit cannot be cleared to 0 even if software writes 0 to this bit.

BEMP Bit (Buffer Empty Interrupt Status)

The USB module sets the BEMP bit to 1 when at least one PIPEBEMP bit in BEMPSTS is set to 1 among the PIPEBEMP bits corresponding to the PIPEBEMPE bits in BEMPENB to which 1 has been set (when the USB module detects the BEMP interrupt status in at least one pipe among the pipes for which software enables the BEMP interrupt output).

For the conditions for PIPEBEMP status assertion, refer to section 28.3.3.3, BEMP Interrupt.

The USB module clears the BEMP bit to 0 when software writes 0 to all the PIPEBEMP bits corresponding to the PIPEBEMPE bits to which 1 has been set.

The BEMP bit cannot be cleared to 0 even if software writes 0 to this bit.

CTRT Bit (Control Transfer Stage Transition Interrupt Status)

When the function controller function is selected, the USB module updates the CTSQ value 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 module detects the next control transfer stage transition.

When the host controller function is selected, the read value is invalid.

DVST Bit (Device State Transition Interrupt Status)

When the function controller function is selected, the USB module updates the DVSQ 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 module detects the next device state transition.

When the host controller function is selected, the read value is invalid.

SOFR Bit (Frame Number Refresh Interrupt Status)

(1) When the host controller function is selected

The USB module sets the SOFR bit to 1 on updating the frame number when software has set the UACT bit in DVSTCTR0 to 1. (A frame number refresh interrupt is detected every 1 ms.)

(2) When the function controller function is selected

The USB module sets the SOFR bit to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.)

The USB module can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.

RESM Bit (Resume Interrupt Status)

When the function controller function is selected, the USB module sets the RESM bit to 1 on detecting the falling edge of the signal on the USBm_DP pin in the suspended state (DVSQ = 1xx).

When the host controller function is selected, the read value is invalid.

VBINT Bit (VBUS Interrupt Status)

The USB module sets the VBINT bit to 1 on detecting a level change (high to low or low to high) in the USBm_VBUS pin input value. The USB module sets the VBSTS bit to indicate the USBm_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.

28.2.14 Interrupt Status Register 1 (INTSTS1)

Addresses: USB0.INTSTS1 000A 0042h, USB1.INTSTS1 000A 0242h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OVRCCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	—
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	—	Reserved	These bits are always read as 0. The write value should always 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 always read as 0. The write value should always 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 always read as 0. The write value should always 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	OVRCCR	Overcurrent Input Change Interrupt Status*2	0: OVRCCR interrupts are not generated. 1: OVRCCR 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 OVRCCR or BCHG bit can be detected even while the clock supply is stopped (while SCKE = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after enabling the clock supply.

No interrupts other than those indicated by the BCHG and OVRCCR bits can be detected while the clock supply is stopped (while SCKE = 0).

INTSTS1 is used to confirm the status of each interrupt when the host controller function is selected.

The various status change interrupts indicated by the bits in INTSTS1 should be enabled only when the host controller function is selected.

SACK Bit (Setup Transaction Normal Response Interrupt Status)

Indicates the status of the setup transaction normal response interrupt when the host controller function is selected.

The USB module detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by the USB module, and sets the SACK bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the SACK interrupt.

When the function controller function 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 function is selected.

The USB module 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 software has set the corresponding interrupt enable bit to 1, the USB module generates the SIGN interrupt.

Specifically, the USB module detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions.

- Timeout is detected by the USB module 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 function 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 function is selected.

The USB module detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by the USB Specifications 2.0, and sets the EOFERR bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the EOFERR interrupt.

After detecting the EOFERR interrupt, the USB module controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried for the USB port and perform re-enumeration of the USB port.

- Modifies the 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 function is selected, the read value is invalid.

ATTCH Bit (ATTCH Interrupt Status)

Indicates the status of the ATTCH interrupt when the host controller function is selected.

The USB module detects the ATTCH interrupt on detecting J-state or K-state of the full-speed signal level for 2.5 μ s, and sets the ATTCH bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt.

Specifically, the USB module 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 μ s.
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 μ s.

When the function controller function 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 function is selected.

The USB module detects the DTCH interrupt on detecting USB bus disconnection, and sets the DTCH bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt.

The USB module detects bus disconnection based on the USB Specifications 2.0.

After detecting the DTCH interrupt, the USB module controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried out for the USB port and make a transition to the wait state for bus connection to the USB port (wait state for ATTCH interrupt generation).

- Modifies the 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 function 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 module detects the BCHG interrupt when a change in the full-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 software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt.

The USB module sets the LNST bits in SYSSTS0 to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST 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 function is selected, the read value is invalid.

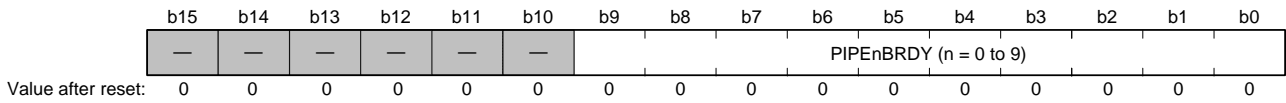
OVRCCR Bit (Overcurrent Input Change Interrupt Status)

Indicates the status of the USBm_OVRCURA and USBm_OVRCURB input pin change interrupt.

The USB module 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 USBm_OVRCURA and USBm_OVRCURB pins, and sets the OVRCCR bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt.

28.2.15 BRDY Interrupt Status Register (BRDYSTS)

Addresses: USB0.BRDYSTS 000A 0046h, USB1.BRDYSTS 000A 0246h



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 always read as 0. The write value should always be 0.	R/W

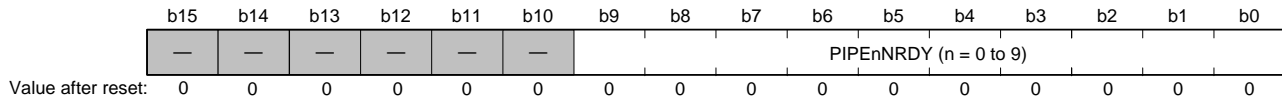
Note 1. When the BRDYM bit is set to 0, to clear the status indicated by the bits in BRDYSTS, write 0 only to the bits to be cleared; write 1 to the other bits.

Note 2. When the BRDYM bit is set to 0, clearing BRDY Interrupts should be done before accessing the FIFO.

BRDYSTS indicates the BRDY interrupt status for each pipe.

28.2.16 NRDY Interrupt Status Register (NRDYSTS)

Addresses: USB0.NRDYSTS 000A 0048h, USB1.NRDYSTS 000A 0248h



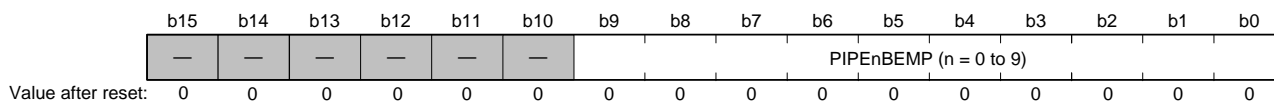
Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0NRDY	NRDY Interrupt Status for PIPE0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b1	PIPE1NRDY	NRDY Interrupt Status for PIPE1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b2	PIPE2NRDY	NRDY Interrupt Status for PIPE2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b3	PIPE3NRDY	NRDY Interrupt Status for PIPE3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b4	PIPE4NRDY	NRDY Interrupt Status for PIPE4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b5	PIPE5NRDY	NRDY Interrupt Status for PIPE5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b6	PIPE6NRDY	NRDY Interrupt Status for PIPE6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b7	PIPE7NRDY	NRDY Interrupt Status for PIPE7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b8	PIPE8NRDY	NRDY Interrupt Status for PIPE8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b9	PIPE9NRDY	NRDY Interrupt Status for PIPE9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b15 to b10	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note : * 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.

NRDYSTS indicates the NRDY interrupt status for each pipe.

28.2.17 BEMP Interrupt Status Register (BEMPSTS)

Addresses: USB0.BEMPSTS 000A 004Ah, USB1.BEMPSTS 000A 024Ah



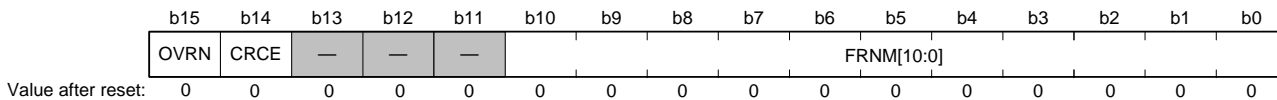
Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BEMP	BEMP Interrupt Status for PIPE0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b1	PIPE1BEMP	BEMP Interrupt Status for PIPE1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b2	PIPE2BEMP	BEMP Interrupt Status for PIPE2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b3	PIPE3BEMP	BEMP Interrupt Status for PIPE3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b4	PIPE4BEMP	BEMP Interrupt Status for PIPE4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b5	PIPE5BEMP	BEMP Interrupt Status for PIPE5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b6	PIPE6BEMP	BEMP Interrupt Status for PIPE6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b7	PIPE7BEMP	BEMP Interrupt Status for PIPE7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b8	PIPE8BEMP	BEMP Interrupt Status for PIPE8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b9	PIPE9BEMP	BEMP Interrupt Status for PIPE9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*
b15 to b10	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note : * 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.

BEMPSTS indicates the BEMP interrupt status for each pipe.

28.2.18 Frame Number Register (FRMNUM)

Addresses: USB0.FRNUM 000A 004Ch, USB1.FRNUM 000A 024Ch



Bit	Symbol	Bit Name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number	The USB module sets the FRNM[10:0] bits to indicate the latest frame number, which is updated every time an SOF packet is issued or received (every 1 ms). Repeat reading the FRNM[10:0] bits until the same value is read twice.	R
b13 to b11	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b14	CRCE	Receive Data Error	0: No error 1: An error occurred	R/W*
b15	OVRN	Overrun/Underrun Detection Status	0: No error 1: An error occurred	R/W*

Note : * Only 0 can be written.

FRMNUM determines the source of isochronous error notification and indicates the frame number.

CRCE Bit (Receive Data Error)

Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer.

Software can clear the CRCE bit to 0 by writing 0 to the CRCE bit. Here, 1 should be written to the other bits in FRMNUM.

(1) When the host controller function is selected

On detecting a CRC error, the USB module generates the internal NRDY interrupt request.

(2) When the function controller function is selected

On detecting a CRC error, the USB module does not generate the internal NRDY interrupt request.

OVRN Bit (Overrun/Underrun Detection Status)

Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer.

Software can clear the OVRN bit to 0 by writing 0 to the OVRN bit. Here, 1 should be written to the other bits in FRMNUM.

(1) When the host controller function is selected

The USB module sets the OVRN bit to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the 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.

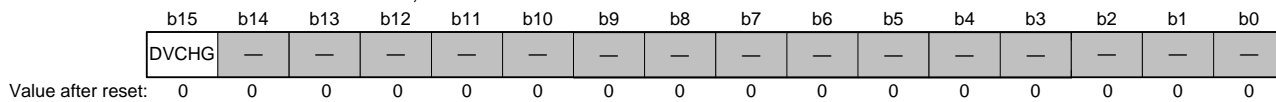
(2) When the function controller function is selected

The USB module sets the OVRN bit to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty.

28.2.19 Device State Change Register (DVCHGR)

Addresses: USB0.DVCHGR 000A 004Eh, USB1.DVCHGR 000A 024Eh

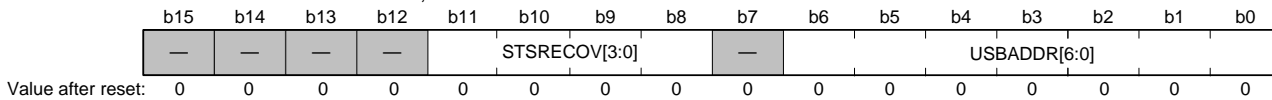


Bit	Symbol	Bit Name	Description	R/W
b14 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b15	DVCHG	Device State Change	0: Disables the writing to the STSRECOV and USBADDR bits. 1: Enables the writing to the STSRECOV and USBADDR bits.	R/W

DVCHGR enables/disables the writing to the USBADDR.STSRECOV[3:0] and USBADDR.USBADDR[6:0] bits. For details, section 9.5.4.5, Canceling Deep Software Standby Mode by USB Suspend/Resume Interrupt.

28.2.20 USB Address Register (USBADDR)

Addresses: USB0.USBADDR 000A 0050h, USB1.USBADDR 000A 0250h



Bit	Symbol	Bit Name	Description	R/W
b6 to b0	USBADDR[6:0]	USB Address	When the function controller function is selected, these bits indicate the USB address assigned by the host when the SET_ADDRESS request is successfully processed.	R/W
b7	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b11 to b8	STSRECOV[3:0]	Status Recovery	b11b8 <ul style="list-style-type: none"> • Recovery when the function controller function is selected 1001: Return to the full-speed state (RHST[2:0] = 010), DVST = 001 (Default state) 1010: Return to the full-speed state (RHST[2:0] = 010), DVST = 010 (Address state) 1011: Return to the full-speed state (RHST[2:0] = 010), DVST = 010 (Configured state) Other than above: Setting prohibited • Recovery when the host controller function is selected 1000: Return to the full-speed state (RHST[2:0] = 010) Other than above: Setting prohibited 	R/W
b15 to b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

USBADDR indicates the USB address.

USBADDR is used when the internal sequencer resumes from the USB power shut-off. For details, see section 9.5.4.5, Canceling Deep Software Standby Mode by USB Suspend/Resume Interrupt.

USBADDR[6:0] Bits (USB Address)

On detecting the USB bus reset, the USB module sets the USBADDR[6:0] bits to 00h.

The writing to these bits is enabled while the DVCHGR.DVCHG bits are set to 1. On returning from the USB power shut-off, the operation can resume to the USB address before the shut-off by the software.

When the host controller function is selected, the USBADDR[6:0] bits are invalid.

The USBADDR[6:0] bits are initialized by a USB bus reset detection.

STSRECOV[3:0] Bits (Status Recovery)

These bits are used to resume the state of the internal sequencer on returning from the USB power shut-off.

The writing to the STSRECOV[3:0] bits is enabled while the DVCHGR.DVCHG bit is set to 1.

28.2.21 USB Request Type Register (USBREQ)

Addresses: USB0.USBREQ 000A 0054h, USB1.USBREQ 000A 0254h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	These bits store the USB request bmRequestType value. <ul style="list-style-type: none"> When the host controller function is selected The USB request data value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while the SUREQ bit is 1. When the function controller function is selected These bits indicate the USB request data value received during the setup transaction. Writing to these bits is invalid. 	R/W*
b15 to b8	BREQUEST[7:0]	Request	These bits store the USB request bRequest value. <ul style="list-style-type: none"> When the host controller function is selected The USB request data value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while the SUREQ bit is 1. When the function controller function is selected These bits indicate the USB request data value received during the setup transaction. Writing to these bits is invalid. 	R/W*

Note : * When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

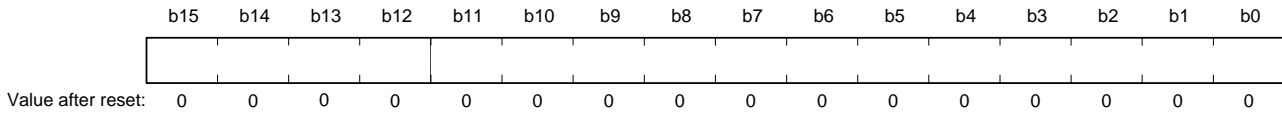
USBREQ stores setup requests for control transfers.

When the function controller function is selected, the values of bRequest and bmRequestType that have been received are stored. When the host controller function is selected, the values of bRequest and bmRequestType to be transmitted are set.

USBREQ is initialized by a USB bus reset.

28.2.22 USB Request Value Register (USBVAL)

Addresses: USB0.USBVAL 000A 0056h, USB1.USBVAL 000A 0256h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	These bits store the USB request wValue value. <ul style="list-style-type: none"> When the host controller function is selected The USB request wValue value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while the SUREQ bit is 1. When the function controller function is selected These bits indicate the USB request wValue value received during the setup transaction. Writing to these bits is invalid. 	R/W*

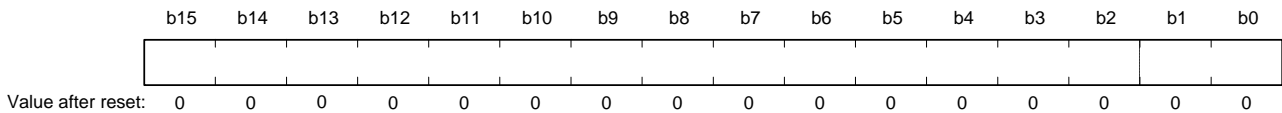
Note : * When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

When the function controller function is selected, the value of wValue that has been received is stored in USBVAL. When the host controller function is selected, the value of wValue to be transmitted is set.

USBVAL is initialized by a USB bus reset.

28.2.23 USB Request Index Register (USBINDX)

Addresses: USB0.USBINDX 000A 0058h, USB1.USBINDX 000A 0258h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	These bits store the USB request wIndex value. <ul style="list-style-type: none"> When the host controller function is selected The USB request wIndex value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while the SUREQ bit in DCPCTR is 1. When the function controller function is selected These bits indicate the USB request wIndex value received during the setup transaction. Writing to these bits is invalid. 	R/W*

Note : * When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

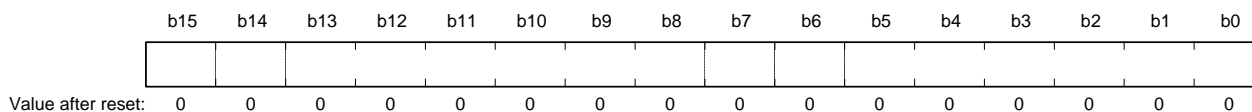
USBINDX stores setup requests for control transfers.

When the function controller function is selected, the value of wIndex that has been received is stored. When the host controller function is selected, the value of wIndex to be transmitted is set.

USBINDX is initialized by a USB bus reset.

28.2.24 USB Request Length Register (USBLENG)

Addresses: USB0.USBLENG 000A 005Ah, USB1.USBLENG 000A 025Ah



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	These bits store the USB request wLength value. <ul style="list-style-type: none"> When the host controller function is selected The USB request wLength value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while the SUREQ bit in DCPCTR is 1. When the function controller function is selected These bits indicate the USB request wLength value received during the setup transaction. Writing to these bits is invalid. 	R/W*

Note : * When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

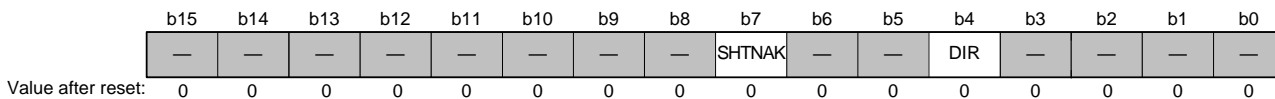
USBLENG stores setup requests for control transfers.

When the function controller function is selected, the value of wLength that has been received is stored. When the host controller function is selected, the value of wLength to be transmitted is set.

USBLENG is initialized by a USB bus reset.

28.2.25 DCP Configuration Register (DCPCFG)

Addresses: USB0.DCPCFG 000A 005Ch, USB1.DCPCFG 000A 025Ch



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	DIR	Transfer Direction*	0: Data receiving direction 1: Data transmitting direction	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*	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 always read as 0. The write value should always be 0.	R/W

Note : * Modify these bits while PID is NAK. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that the PBUSY bit is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

DCPCFG specifies the data transfer direction for the default control pipe (DCP).

DIR Bit (Transfer Direction)

When the host controller function is selected, the DIR bit sets the transfer direction of the data stage and status stage.

When the function controller function 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.

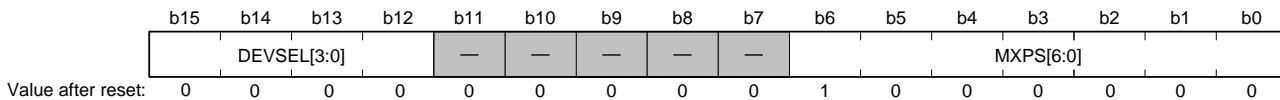
The SHTNAK bit is valid when the selected pipe in the receiving direction.

When the SHTNAK bit is set to 1, the USB module modifies the PID bits for the DCP to NAK on determining the end of the transfer. The USB module determines that the transfer has ended on the following condition.

- A short packet (including a zero-length packet) is successfully received.

28.2.26 DCP Maximum Packet Size Register (DCPMAXP)

Addresses: USB0.DCPMAXP 000A 005Eh, USB1.DCPMAXP 000A 025Eh



Bit	Symbol	Bit Name	Description	R/W																																			
b6 to b0	MXPS[6:0]	Maximum Packet Size*2	These bits specify the maximum data payload (maximum packet size) for the DCP. The initial value is 40h (64 bytes). These bits should be set to the value based on the USB Specifications. While MXPS = 0, do not write to the FIFO buffer or do not set PID to BUF.	R/W																																			
b11 to b7	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W																																			
b15 to b12	DEVSEL[3:0]	Device Select*1	<table border="0"> <tr> <td>b15</td> <td>b14</td> <td>b13</td> <td>b12</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0: USB address 0000</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1: USB address 0001</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0: USB address 0010</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1: USB address 0011</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0: USB address 0100</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>1: USB address 0101</td> </tr> </table> Other than above: Setting prohibited	b15	b14	b13	b12		0	0	0	0	0: USB address 0000	0	0	0	1	1: USB address 0001	0	0	1	0	0: USB address 0010	0	0	1	1	1: USB address 0011	0	1	0	0	0: USB address 0100	0	1	0	1	1: USB address 0101	R/W
b15	b14	b13	b12																																				
0	0	0	0	0: USB address 0000																																			
0	0	0	1	1: USB address 0001																																			
0	0	1	0	0: USB address 0010																																			
0	0	1	1	1: USB address 0011																																			
0	1	0	0	0: USB address 0100																																			
0	1	0	1	1: USB address 0101																																			

Note 1. Modify the DEVSEL bits while PID is NAK and the SUREQ bit is 0. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that the PBUSY bit is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 2. Modify the MXPS[6:0] bits while PID is NAK. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that the PBUSY bit is 0. However, if the PID bits have been modified to NAK by the USB module, 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 bits, clear the buffer by setting the BCLR bit to 1.

DCPMAXP specifies the maximum packet size for the DCP.

DEVSEL[3:0] Bits (Device Select)

When the host controller function is selected, these bits specify the USB 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 0010, the DEVADD2 register should be set.

When the function controller function is selected, the DEVSEL[3:0] bits should be set to 0000b.

28.2.27 DCP Control Register (DCPCTR)

Addresses: USB0.DCPCTR 000A 0060h, USB1.DCPCTR 000A 0260h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	SUREQ	—	—	SUREQCLR	—	—	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 always read as 0. The write value should always be 0.	R/W
b5	PBUSY	Pipe Busy	0: DCP is not used for the transaction. 1: DCP is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Monitor	0: DATA0 1: DATA1	R
b7	SQSET	Toggle Bit Set*3	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Toggle Bit Clear*3	0: Invalid 1: Specifies DATA0.	R/W*1
b10, b9	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b11	SUREQCLR	SUREQ Bit Clear	This bit is always read as 0, even when 1 is written to. Writing 0 to this bit is invalid.	R/W*1
b13, b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b14	SUREQ	Setup Token Transmission	0: Invalid 1: Transmits the setup packet.	R/W*2
b15	BSTS	Buffer Status	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. This bit is always read as 0. Only 1 can be written.

Note 2. Only 1 can be written.

Note 3. Write 1 to the SQSET and SQCLR bits while PID is NAK. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that the PBUSY bit is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

DCPCTR is used to confirm the buffer memory status, change and confirm the data PID sequence bits, and set the response PID for the DCP.

The CCPL and PID[1:0] bits in DCPCTR are initialized by a USB bus reset.

PID[1:0] Bits (Response PID)

The PID[1:0] bits control the response type of the USB module during control transfer.

(1) When the host controller function 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 UACT is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, the USB module executes the OUT transaction.
- When the receiving direction is set
Check that the FIFO buffer is empty (or empty the buffer) while UACT is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, the USB module executes the IN transaction.

The USB module modifies the setting of the PID[1:0] bits as follows.

- The USB module sets PID to STALL (11) on receiving the data of a size exceeding the maximum packet size when software has set the PID[1:0] bits to BUF.
- The USB module sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times.
- The USB module also sets PID to STALL (11) on receiving the STALL handshake.

(2) When the function controller function is selected

The USB module modifies the setting of the PID[1:0] bits as follows.

- The USB module modifies the PID[1:0] bits to NAK on receiving the setup packet. Here, the USB module sets VALID to 1. Software cannot modify the setting of the PID[1:0] bits until software sets VALID to 0.
- The USB module sets PID to STALL (11) on receiving the data of a size exceeding the maximum packet size when software has set the PID[1:0] bits to BUF.
- The USB module sets PID to STALL (1x) on detecting the control transfer sequence error.
- The USB module sets PID to NAK on detecting the USB bus reset.

The USB module does not reference to the setting of the PID[1:0] bits while the SET_ADDRESS request is processed (auto processing).

CCPL Bit (Control Transfer End Enable)

When the function controller function is selected, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.

When software sets the CCPL bit to 1 while the corresponding PID bits are set to BUF, the USB module completes the control transfer stage.

Specifically, during control read transfer, the USB module transmits the ACK handshake in response to the OUT transaction from the USB host, and transmits the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, the USB module operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of the CCPL bit.

The USB module modifies the CCPL bit from 1 to 0 on receiving a new setup packet.

Software cannot write 1 to the CCPL bit while the VALID bit is 1.

When the host controller function 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 bits from BUF to NAK.

The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.

Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible.

For details, refer to section 28.3.4.1, Pipe Control Register Switching Procedures.

SQMON Bit (Sequence Toggle Bit Monitor)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer.

The USB module allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.

When the function controller function is selected, the USB module sets the SQMON bit to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet.

When the function controller function is selected, the USB module does not reference the SQMON bit during the IN/OUT transaction of the status stage, and does not allow the SQMON bit to toggle upon normal completion.

SQSET Bit (Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR Bit (Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The SQCLR bit always indicates 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR Bit (SUREQ Bit Clear)

When the host controller function is selected, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The SUREQCLR bit always 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 module 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 UACT bit is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.

When the function controller function is selected, be sure to write 0 to the SUREQCLR bit.

SUREQ Bit (Setup Token Transmission)

The USB module transmits the setup packet by setting the SUREQ bit to 1 when the host controller function is selected. After completing the setup transaction process, the USB module generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0.

The USB module also clears the SUREQ bit to 0 when software sets the SUREQCLR bit to 1.

Before setting the SUREQ bit to 1, set the DEVSEL 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 bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not modify the DEVSEL bits, USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is completed (SUREQ = 1).

Write 1 to the SUREQ bit only when transmitting the setup token; for other purposes, write 0.

When the function controller function 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 ISEL bit setting as follows.

- When ISEL = 0, the BSTS bit indicates whether the received data can be read from the buffer.
- When ISEL = 1, the BSTS bit indicates whether the data to be transmitted can be written to the buffer.

28.2.28 Pipe Window Select Register (PIPESEL)

Addresses: USB0.PIPESEL 000A 0064h, USB1.PIPESEL 000A 0264h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PIPESEL[3:0]	Pipe Window Select	b3 b2 b1 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 Other than above: Setting prohibited	R/W
b15 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PIPESEL is a register to specify the pipe number.

PIPE1 to PIPE 9 should be set using PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN.

After selecting the pipe using PIPESEL, functions of the pipe should be set using PIPECFG, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set regardless of the pipe selection in PIPESEL.

PIPESEL[3:0] Bits (Pipe Window Select)

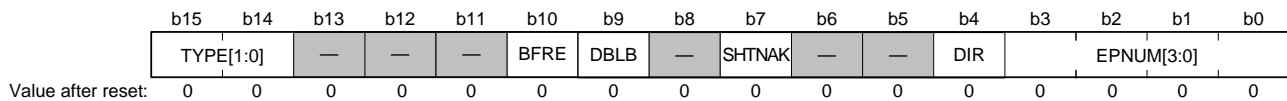
The PIPESEL[3:0] bits select the pipe number corresponding to PIPECFG, PIPEMAXP, and PIPEPERI which data are written to or read from.

Selecting a pipe number through the PIPESEL[3:0] bits allows writing to and reading from PIPECFG, PIPEMAXP, and PIPEPERI which correspond to the selected pipe number.

When PIPESEL[3:0] = 0000, 0 is read from all of the bits in PIPECFG, PIPEMAXP, PIPEPERI, and PIPEnCTR. Writing to these bits is invalid.

28.2.29 Pipe Configuration Register (PIPECFG)

Addresses: USB0.PIPECFG 000A 0068h, USB1.PIPECFG 000A 0268h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number*1	These bits specify the endpoint number for the selected pipe. Setting 0000 means unused pipe.	R/W
b4	DIR	Transfer Direction*2*3	0: Receiving direction 1: Transmitting direction	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer	R/W
b8	—	Reserved	This bit is always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W
b15, b14	TYPE[1:0]	Transfer Type*1	b15 b14 • PIPE1 and PIPE2 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Isochronous transfer • PIPE3 to PIPE5 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Setting prohibited • PIPE6 to PIPE9 0 0: Pipe not used 0 1: Setting prohibited 1 0: Interrupt transfer 1 1: Setting prohibited	R/W

Note 1. Modify the TYPE, SHTNAK, and EPNUM bits while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID bits have been modified to NAK by the USB module, 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 bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID bits have been modified to NAK by the USB module, checking the 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 ACLRM bit continuously through software to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE 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 unused pipe.

Do not make the settings such that the combination of the settings of the DIR and EPNUM bits should be the same for two or more pipes (EPNUM = 0000 can be set for all of the pipes).

DIR Bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.

When software has set the DIR bit to 0, the USB module uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB module uses the selected pipe in the transmitting direction.

SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction.

The SHTNAK bit is valid when the selected pipe is PIPE1 to PIPE5 in the receiving direction.

When software has set the SHTNAK bit to 1 for the selected pipe in the receiving direction, the USB module modifies the PID bits corresponding to the selected pipe to NAK on determining the end of the transfer. The USB module determines that the transfer has ended on any of the following conditions.

A short packet (including a zero-length packet) is successfully received.

The transaction counter is used and the number of packets specified by the counter are successfully received.

DBLB Bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe.

The DBLB bit is valid when PIPE1 to PIPE5 are selected.

BFRE Bit (BRDY Interrupt Operation Specification)

Specifies the BRDY interrupt generation timing from the USB module to the CPU with respect to the selected pipe.

When software has set the BFRE bit to 1 and the selected pipe is in the receiving direction, the USB module detects the transfer completion and generates the BRDY interrupt on having read the relevant packet.

When the BRDY interrupt is generated with the above conditions, software needs to write 1 to the BCLR bit. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.

When software has set the BFRE bit to 1 and the selected pipe is in the transmitting direction, the USB module does not generate the BRDY interrupt.

For details, refer to section 28.3.3.1, BRDY Interrupt.

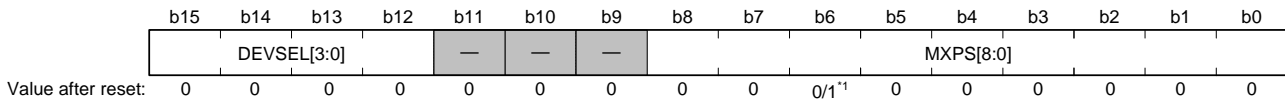
TYPE[1:0] Bits (Transfer Type)

The TYPE[1:0] bits select the transfer type for the pipe selected by the PIPESEL bits (selected pipe).

Before setting PID to BUF for the selected pipe (before starting USB communication using the selected pipe), be sure to set the TYPE[1:0] bits to a value other than 00b.

28.2.30 Pipe Maximum Packet Size Register (PIPEMAXP)

Addresses: USB0.PIPEMAXP 000A 006Ch, USB1.PIPEMAXP 000A 026Ch



Bit	Symbol	Bit Name	Description	R/W
b8 to b0	MXPS[8:0]	Maximum Packet Size*3	<ul style="list-style-type: none"> • PIPE1 and PIPE2 1 byte (001h) to 256 bytes (100h) • PIPE3 to PIPE5 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h) (Bits [8:7] and [2:0] are not provided.) • PIPE6 to PIPE9 1 byte (001h) to 64 bytes (040h) (Bits [8:7] are not provided.) 	R/W
b11 to b9	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*2	b3 b2 b1 b0 0 0 0 0: USB address 0000 0 0 0 1: USB address 0001 0 0 1 0: USB address 0010 0 0 1 1: USB address 0011 0 1 0 0: USB address 0100 0 1 0 1: USB address 0101 Other than above: Setting prohibited	R/W

- Note 1. The initial value of the MXPS[8:0] bits is 0000h when no pipe is selected with the PIPESEL bits in PIPESEL and 0040h when a pipe is selected.
- Note 2. Modify the DEVSEL[3:0] bits while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.
- Note 3. Modify the MXPS[8:0] bits while PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID bits have been modified to NAK by the USB module, 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)

Specifies the maximum data payload (maximum packet size) for the selected pipe.

These bits should be set to the appropriate value for each transfer type based on the USB Specifications. While MXPS = 0, do not write to the FIFO buffer or set PID to BUF.

DEVSEL[3:0] Bits (Device Select)

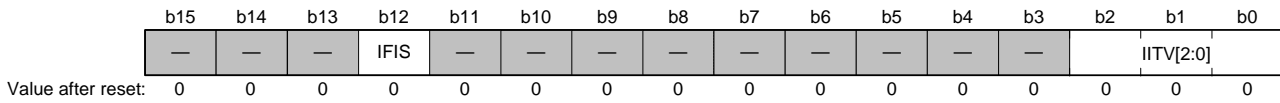
When the host controller function is selected, these bits specify the USB address of the peripheral device which is the communication target.

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 0010, the DEVADD2 register should be set.

When the function controller function is selected, the DEVSEL[3:0] bits should be set to 0000b.

28.2.31 Pipe Cycle Control Register (PIPEPERI)

Addresses: USB0.PIPEPERI 000A 006Eh, USB1.PIPEPERI 000A 026Eh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	IITV[2:0]	Interval Error Detection Interval	Specifies the interval error detection timing for the selected pipe in terms of frames, which is expressed as n-th power of 2. As described later, the detailed functions are different in host controller mode and in function controller mode. 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 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 000 for PIPE3 to PIPE5.	R/W
b11 to b3	—	Reserved	These bits are always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W

Note: Modify the IITV bits while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID bits have been modified to NAK by the USB module, 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.

IFIS Bit (Isochronous IN Buffer Flush)

Specifies whether to flush the buffer when the pipe selected by the PIPESEL bits (selected pipe) is used for isochronous IN transfers.

When the function controller function is selected and the selected pipe is for isochronous IN transfers, the USB module automatically clears the FIFO buffer when the USB module fails to receive the IN token from the USB host within the interval set by the IITV bits in terms of frames.

In double buffer mode (DBLB = 1), the USB module only clears the data in the plane used earlier.

The USB module clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USB module has expected to receive the IN token. Even if the SOF packet is damaged, the USB module also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation function.

When the host controller function is selected, set the IFIS bit to 0.

When the selected pipe is not for isochronous transfer, set the IFIS bit to 0.

28.2.32 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9)

- PIPEnCTR (n = 1 to 5)

Addresses: USB0.PIPE1CTR 000A 0070h, USB0.PIPE2CTR 000A 0072h, USB0.PIPE3CTR 000A 0074h,
 USB0.PIPE4CTR 000A 0076h, USB0.PIPE5CTR 000A 0078h, USB1.PIPE1CTR 000A 0270h,
 USB1.PIPE2CTR 000A 0272h, USB01.PIPE3CTR 000A 0274h, USB1.PIPE4CTR 000A 0276h,
 USB1.PIPE5CTR 000A 0278h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BSTS	INBUFM	—	—	—	ATREPM	ACLARM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b5	PBUSY	Pipe Busy	0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction.	R
b6	SQMON	Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Toggle Bit Clear*2	0: Invalid 1: Specifies DATA0.	R/W*1
b9	ACLARM	Auto Buffer Clear Mode*3	0: Disabled 1: Enabled (all buffers are initialized)	R/W
b10	ATREPM	Auto Response Mode*2	0: Auto response is disabled. 1: Auto response is enabled.	R/W
b13 to b11	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b14	INBUFM	Transmit Buffer Monitor	0: There is no data to be transmitted in the buffer memory. 1: There is data to be transmitted in the buffer memory.	R
b15	BSTS	Buffer Status	0: Buffer access by the CPU is disabled. 1: Buffer access by the CPU is enabled.	R

Note 1. Only 0 can be read and 1 can be written.

Note 2. Modify the ATREPM bit or write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that the PBUSY bit in DCPCTR is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit in DCPCTR through software is not necessary.

Note 3. Modify the ACLARM bit while PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that the PBUSY bit in DCPCTR is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit in DCPCTR through software is not necessary.

PIPEnCTR is used to confirm the buffer memory status for the corresponding pipe, change and confirm the data PID sequence bit, determine whether auto response mode is set, determine whether auto buffer clear mode is set, and set a response PID for PIPE1 to PIPE9. PIPEnCTR can be set regardless of the pipe selection in PIPESEL.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.

The default setting of the PID[1:0] bits is NAK. Modify the setting of the PID[1:0] bits to BUF to use the relevant pipe for USB transfer. Table 28.9 and Table 28.10 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB module depending on the PID 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 in DCPCTR 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 module, checking the PBUSY bit through software is not necessary.

The USB module modifies the setting of the PID[1:0] bits in the following cases.

- The USB module sets PID to NAK on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and software has set the SHTNAK bit for the selected pipe to 1.
- The USB module sets PID to STALL (11) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB module sets PID to NAK on detecting a USB bus reset when the function controller function is selected.
- The USB module sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times when the host controller function is selected.
- The USB module sets PID to STALL (11) on receiving the STALL handshake when the host controller function is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00) to STALL, set 10.
- To make a transition from BUF (01) to STALL, set 11.
- To make a transition from STALL (11) to NAK, set 10 and then 00.
- To make a transition from STALL to BUF, set 00 (NAK) and then 01 (BUF).

PBUSY Bit (Pipe Busy)

The PBUSY bit indicates whether the relevant pipe is being currently used or not for the transaction.

The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.

Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible.

For details, refer to section 28.3.4.1, Pipe Control Register Switching Procedures.

SQMON Bit (Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.

When the relevant pipe is not for the isochronous transfer, the USB module allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.

SQSET Bit (Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.

Setting the SQSET bit to 1 through software allows the USB module to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQSET bit to 0.

SQCLR Bit (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 module to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always 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 28.11 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 function 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 module responds to the token from the USB host as described below.

(1) When the relevant pipe is for bulk IN transfer (TYPE = 01 and DIR = 1)

When ATREPM = 1 and PID = BUF, the USB module transmits a zero-length packet in response to the IN token.

The USB module updates (allows toggling of) the sequence toggle bit (DATA-PID) each time the USB module receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.).

In this case, the USB module does not generate the BRDY or BEMP interrupt.

(2) When the relevant pipe is for bulk OUT transfer (TYPE = 01 and DIR = 0)

When ATREPM = 1 and PID = BUF, the USB module returns NAK in response to the OUT token and generates the NRDY interrupt.

For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode.

When the relevant pipe is for isochronous transfer, be sure to set the ATREPM bit to 0.

When the host controller function 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 (DIR = 1), the USB module sets the INBUFM bit to 1 when CPU (DTC or DMACA) completes writing data to at least one FIFO buffer plane.

The USB module sets the INBUFM bit to 0 when the USB module completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (DBLB = 1), the USB module sets the INBUFM bit to 0 when the USB module completes transmitting the data from the two FIFO buffer planes before CPU (DTC or DMACA) 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 (DIR = 0).

BSTS Bit (Buffer Status)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS bit depends on the settings of the DIR, BFRE, and DCLRM bits as shown in Table 28.12.

Table 28.9 Operation of USB Module depending on PID Bit Setting (When Host Controller Function is Selected)

PID Bit	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB Module
00 (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01 (BUF)	Bulk or interrupt	Operation does not depend on the setting.	Issues tokens while the 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 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.
10 (STALL) or 11 (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.

Table 28.10 Operation of USB Module depending on PID Bit Setting (When Function Controller Function is Selected)

PID Bit	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB Module
00 (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host. For the operation when the ATREPM bit is 1, refer to the description of the ATREPM bit.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.
01 (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Bulk or interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
	Isochronous	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Discards data if not ready.
	Isochronous	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Transmits the zero-length packet if not ready.
10 (STALL) or 11 (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 28.11 Information Cleared by USB Module by Setting ACLRM = 1

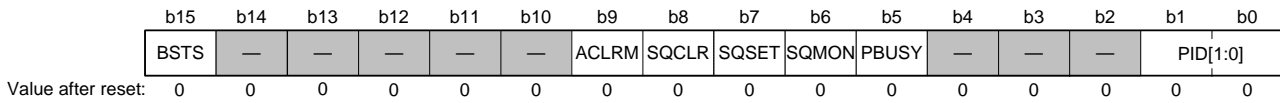
No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the relevant pipe (both FIFO buffer planes are cleared when double buffer mode is selected)	when the pipe is to be initialized
2	The interval count value when the relevant pipe is for isochronous transfer	When the interval count value is to be reset
3	Internal flags concerning the BFRE bit	When the BFRE setting is modified
4	FIFO buffer toggle control	When the DBLB setting is modified
5	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

Table 28.12 Operation of BSTS Bit

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Bit Function	
0	0	0	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.	
		1	Setting prohibited	
	1	0	1: The received data can be read from the FIFO buffer. 0: Software has set the BCLR bit to 1 after the received data has been completely read from the FIFO buffer.	
		1	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.	
	1	0	0	1: The transmit data can be written to the FIFO buffer. 0: The transmit data has been completely written to the FIFO buffer.
			1	Setting prohibited
1		0	Setting prohibited	
		1	Setting prohibited	

- PIPE_nCTR (n = 6 to 9)

Addresses: USB0.PIPE6CTR 000A 007Ah, USB0.PIPE7CTR 000A 007Ch, USB0.PIPE8CTR 000A 007Eh,
 USB0.PIPE9CTR 000A 0080h, USB1.PIPE6CTR 000A 027Ah, USB1.PIPE7CTR 000A 027Ch,
 USB1.PIPE8CTR 000A 027Eh, USB1.PIPE9CTR 000A 0280h



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 always read as 0. The write value should always be 0.	R/W
b5	PBUSY	Pipe Busy	0: The relevant pipe is not used at the USB bus. 1: The relevant pipe is used at the USB bus.	R
b6	SQMON	Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Toggle Bit Clear*2	0: Invalid 1: Specifies DATA0.	R/W*1
b9	ACLRM	Auto Buffer Clear Mode*2*3	0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled (all buffers are initialized)	R/W
b14 to b10	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b15	BSTS	Buffer Status	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read and 1 can be written.

Note 2. Write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 3. Modify the ACLRM bit while PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.

The default setting of the PID[1:0] bits is NAK. Modify the setting of the PID[1:0] bits to BUF to use the relevant pipe for USB transfer. Table 28.9 and Table 28.10 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB module depending on the PID 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 bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

The USB module modifies the setting of the PID[1:0] bits in the following cases.

- The USB module sets PID to STALL (11) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB module sets PID to NAK on detecting a USB bus reset when the function controller function is selected.
- The USB module sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times when the host controller function is selected.
- The USB module sets PID to STALL (11) on receiving the STALL handshake when the host controller function is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00) to STALL, set 10.
- To make a transition from BUF (01) to STALL, set 11.
- To make a transition from STALL (11) to NAK, set 10 and then 00.
- To make a transition from STALL to BUF, set 00 (NAK) and then 01 (BUF).

PBUSY Bit (Pipe Busy)

The PBUSY bit indicates whether the relevant pipe is being currently used or not for the transaction.

The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.

Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible.

SQMON Bit (Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.

The USB module allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.

SQSET Bit (Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.

Setting the SQSET bit through software allows the USB module to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQSET bit to 0.

SQCLR Bit (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 module to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always 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 28.13 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.

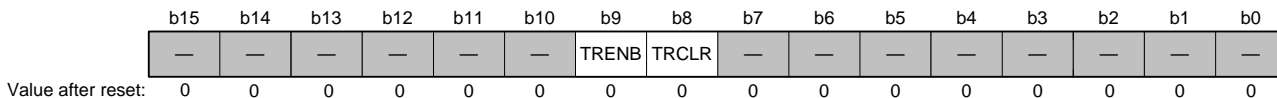
The meaning of the BSTS bit depends on the settings of the DIR, BFRE, and DCLRM bits as shown in Table 28.12.

Table 28.13 Information Cleared by USB Module by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the selected pipe	
2	The interval count value when the selected pipe is for interrupt transfer and the host controller function is selected	When the interval count value is to be reset
3	Internal flags concerning the BFRE bit	When the BFRE setting is modified
4	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

28.2.33 PIPE_n Transaction Counter Enable Registers (PIPE_nTRE) (n = 1 to 5)

Addresses: USB0.PIPE1TRE 000A 0090h, USB0.PIPE2TRE 000A 0094h, USB0.PIPE3TRE 000A 0098h, USB0.PIPE4TRE 000A 009Ch, USB0.PIPE5TRE 000A 00A0h, USB1.PIPE1TRE 000A 0290h, USB1.PIPE2TRE 000A 0294h, USB1.PIPE3TRE 000A 0298h, USB1.PIPE4TRE 000A 029Ch, USB1.PIPE5TRE 000A 02A0h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W

Note: Modify each bit in PIPE_nTRE while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

PIPE_nTRE enables or disables the transaction counter corresponding to PIPE1 to PIPE5, and clears the transaction counter.

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 TRNCNT bits through software allows the USB module to control hardware as described below on having received the number of packets equal to the setting of the TRNCNT bits.

- While the SHTNAK bit is 1, the USB module modifies the PID bits to NAK for the corresponding pipe on having received the number of packets equal to the setting of the TRNCNT bits.
- While the BFRE bit is 1, the USB module asserts the BRDY interrupt on having received the number of packets equal to the setting of the TRNCNT bits and then reading out the last received data.

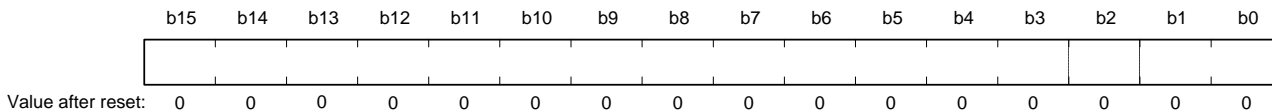
For the pipe in the transmitting direction, set the TRENB bit to 0.

When the transaction counter is not used, set the TRENB bit to 0.

When the transaction counter is used, set the TRNCNT 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.

28.2.34 PIPEn Transaction Counter Registers (PIPEnTRN) (n = 1 to 5)

Addresses: USB0.PIPE1TRN 000A 0092h, USB0.PIPE2TRN 000A 0096h, USB0.PIPE3TRN 000A 009Ah, USB0.PIPE4TRN 000A 009Eh, USB0.PIPE5TRN 000A 00A2h, USB1.PIPE1TRN 000A 0292h, USB1.PIPE2TRN 000A 0296h, USB1.PIPE3TRN 000A 029Ah, USB1.PIPE4TRN 000A 029Eh, USB1.PIPE5TRN 000A 02A2h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	<ul style="list-style-type: none"> When written to Specifies the number of transactions to be transferred through DMA. When read from Indicates the specified number of transactions if the TRENb bit is 0. Indicates the number of currently counted transactions if the TRENb bit is 1. 	R/W

PIPEnTRN is a transaction counter corresponding to PIPE1 to PIPE5.

PIPEnTRN retains the setting by a USB bus reset.

TRNCNT[15:0] Bits (Transaction Counter)

The USB module increments the value of the TRNCNT bits by one when all of the following conditions are satisfied on receiving the packet.

- TRENb = 1
- (TRNCNT set value ≠ current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the MXPS bits.

The USB module clears the value of the TRNCNT bits to 0 when any of the following conditions are satisfied.

(1) All of the following conditions are satisfied.

- TRENb = 1
- (TRNCNT set value = current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the MXPS bits.

(2) All of the following conditions are satisfied.

- TRENb = 1
- The USB module has received a short packet.

(3) All of the following conditions are satisfied.

- TRENb = 1
- Software has set the TRCLR bit to 1.

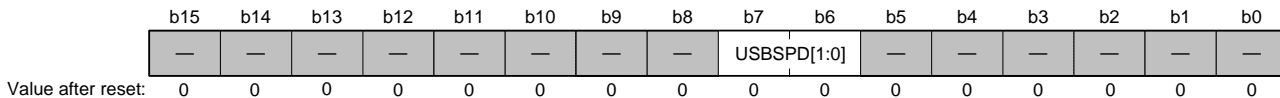
For the pipe in the transmitting direction, set the TRNCNT bits to 0.

When the transaction counter is not used, set the TRNCNT bits to 0.

Setting the number of transactions to be transferred to the TRNCNT bits is only enabled when the TRENb bit in PIPEnTRE register is 0. To modify the number of transactions to be transferred, set the TRCLR bit in the PIPEnTRE register to 1 (to clear the current counter value) before setting the TRENb bit to 1.

28.2.35 Device Address n Configuration Registers (DEVADDn) (n = 0 to 5)

Addresses: USB0.DEVADD0 000A 00D0h, USB0.DEVADD1 000A 00D2h, USB0.DEVADD2 000A 00D4h,
 USB0.DEVADD3 000A 00D6h, USB0.DEVADD4 000A 00D8h, USB0.DEVADD5 000A 00DAh,
 USB1.DEVADD0 000A 02D0h, USB1.DEVADD1 000A 02D2h, USB1.DEVADD2 000A 02D4h,
 USB1.DEVADD3 000A 02D6h, USB1.DEVADD4 000A 02D8h, USB1.DEVADD5 000A 02DAh



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are always read as 0. The write value should always 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: Setting prohibited 1 0: Full speed 1 1: Setting prohibited	R/W
b15 to b8	—	Reserved	These bits are always read as 0. The write value should always 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 function 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 bits.
- The PID bits are set to BUF for the selected pipe or the selected pipe is the DCP with the SUREQ bit set to 1.

USBSPD[1:0] Bits (Transfer Speed of Communication Target Device)

Specifies the USB transfer speed of the communication target peripheral device.

Set these bits to 10b when a full-speed device is connected via the HUB.

When the host controller function is selected, the USB module refers to the setting of the USBSPD[1:0] bits to generate packets.

When the function controller function is selected, set these bits to 00b.

28.2.36 Deep Standby USB Transceiver Control/Pin Monitor Register (DPUSR0R)

Address: 000A 0400h

	b31	b30	b29	b28	b27	b26	b25	b24
	DVBSTS1	—	DOVCB1	DOVCA1	—	—	DM1	DP1
Value after reset:	x	0	x	x	0	0	x	x
	b23	b22	b21	b20	b19	b18	b17	b16
	DVBSTS0	—	DOVCB0	DOVCA0	—	—	DM0	DP0
Value after reset:	x	0	x	x	0	0	x	x
	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	FIXPHY1	—	—	—	SRPC1
Value after reset:	0	0	0	0	0	0	0	0
	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	FIXPHY0	—	—	—	SRPC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SRPC0	USB0 Single End Receiver Control	0: DP and DM inputs are disabled. 1: DP and DM inputs are enabled.	R/W
b3 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	FIXPHY0	USB0 Transceiver Output Fix	0: The outputs are fixed during normal mode or on return from deep software standby mode. 1: The outputs are fixed on making a transition to deep software standby mode.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b8	SRPC1	USB1 Single End Receiver Control	0: DP and DM inputs are disabled. 1: DP and DM inputs are enabled.	R/W
b11 to b9	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b12	FIXPHY1	USB1 Transceiver Output Fix	0: The outputs are fixed during normal mode or on return from deep software standby mode. 1: The outputs are fixed on making a transition to deep software standby mode.	R/W
b15 to b13	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b16	DP0	USB0 DP Input	DP input signal on the USB port 0 side is indicated.	R
b17	DM0	USB0 DM Input	DM input signal on the USB port 0 side is indicated.	R
b19, b18	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b20	DOVCA0	USB0 OVRCURA Input	OVRCURA input signal on the USB port 0 side is indicated.	R
b21	DOVCB0	USB0 OVRCURB Input	OVRCURB input signal on the USB port 0 side is indicated.	R
b22	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b23	DVBSTS0	USB0 VBUS Input	VBUS input signal on the USB port 0 side is indicated.	R

Bit	Symbol	Bit Name	Description	R/W
b24	DP1	USB1 DP Input	DP input signal on the USB port 1 side is indicated.	R
b25	DM1	USB1 DM Input	DM input signal on the USB port 1 side is indicated.	R
b27, b26	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b28	DOVCA1	USB1 OVRCURA Input	OVRCURA input signal on the USB port 1 side is indicated.	R
b29	DOVCB1	USB1 OVRCURB Input	OVRCURB input signal on the USB port 1 side is indicated.	R
b30	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b31	DVBSTS1	USB1 VBUS Input	VBUS input signal on the USB port 1 side is indicated.	R

DPUSR0R controls data of the USB internal transceivers and monitors the USB pins.

SRPCn Bit (USBn Single End Receiver Control) (n = 0 or 1)

The SRPCn bit controls the DP and DM inputs of the transceiver on the USB port n side.

This bit is valid only when the FIXPHYn bit is 1. .

FIXPHYn Bit (USBn PHY Output Fix) (n = 0 or 1)

The FIXPHYn bit holds the outputs to the transceiver on the USB port n side disabled.

28.2.37 Deep Standby USB Suspend/Resume Interrupt Register (DPUSR1R)

Address: 000A 0404h

	b31	b30	b29	b28	b27	b26	b25	b24
	DVBINT1	—	DOVRCRB1	DOVRCRA1	—	—	DMINT1	DPINT1
Value after reset:	0	0	0	0	0	0	0	0
	b23	b22	b21	b20	b19	b18	b17	b16
	DVBINT0	—	DOVRCRB0	DOVRCRA0	—	—	DMINT0	DPINT0
Value after reset:	0	0	0	0	0	0	0	0
	b15	b14	b13	b12	b11	b10	b9	b8
	DVBSE1	—	DOVRCRBE1	DOVRCRAE1	—	—	DMINTE1	DPINTE1
Value after reset:	0	0	0	0	0	0	0	0
	b7	b6	b5	b4	b3	b2	b1	b0
	DVBSE0	—	DOVRCRBE0	DOVRCRAE0	—	—	DMINTE0	DPINTE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DPINTE0	USB0 DP Interrupt Enable/Clear	0: Recovery from deep software standby mode by the USB port 0 DP input is disabled. 1: Recovery from deep software standby mode by the USB port 0 DP input is enabled.	R/W
b1	DMINTE0	USB0 DM Interrupt Enable/Clear	0: Recovery from deep software standby mode by the USB port 0 DM input is disabled. 1: Recovery from deep software standby mode by the USB port 0 DM input is enabled.	R/W
b3, b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	DOVRCRAE0	USB0 OVRCURA Interrupt Enable/Clear	0: Recovery from deep software standby mode by the USB port 0 OVRCURA input is disabled. 1: Recovery from deep software standby mode by the USB port 0 OVRCURA input is enabled.	R/W
b5	DOVRCRBE0	USB0 OVRCURB Interrupt Enable/Clear	0: Recovery from deep software standby mode by the USB port 0 OVRCURB input is disabled. 1: Recovery from deep software standby mode by the USB port 0 OVRCURB input is enabled.	R/W
b6	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b7	DVBSE0	USB0 VBUS Interrupt Enable/Clear	0: Recovery from deep software standby mode by the USB port 0 VBUS input is disabled. 1: Recovery from deep software standby mode by the USB port 0 VBUS input is enabled.	R/W
b8	DPINTE1	USB1 DP Interrupt Enable/Clear	0: Recovery from deep software standby mode by the USB port 1 DP input is disabled. 1: Recovery from deep software standby mode by the USB port 1 DP input is enabled.	R/W
b9	DMINTE1	USB1 DM Interrupt Enable/Clear	0: Recovery from deep software standby mode by the USB port 1 DM input is disabled. 1: Recovery from deep software standby mode by the USB port 1 DM input is enabled.	R/W

Bit	Symbol	Bit Name	Description	R/W
b11, b10	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b12	DOVRCRAE1	USB1 OVRCURA Interrupt Enable/Clear	0: Recovery from deep software standby mode by the USB port 1 OVRCURA input is disabled. 1: Recovery from deep software standby mode by the USB port 1 OVRCURA input is enabled.	R/W
b13	DOVRCRBE1	USB1 OVRCURB Interrupt Enable/Clear	0: Recovery from deep software standby mode by the USB port 1 OVRCURB input is disabled. 1: Recovery from deep software standby mode by the USB port 1 OVRCURB input is enabled.	R/W
b14	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b15	DVBSE1	USB1 VBUS Interrupt Enable/Clear	0: Recovery from deep software standby mode by the USB port 1 VBUS input is disabled. 1: Recovery from deep software standby mode by the USB port 1 VBUS input is enabled.	R/W
b16	DPINT0	USB0 DP Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode by the USB port 0 DP input. 1: The system has returned from deep software standby mode by the USB port 0 DP input.	R
b17	DMINT0	USB0 DM Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode by the USB port 0 DM input. 1: The system has returned from deep software standby mode by the USB port 0 DM input.	R
b19, b18	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b20	OVRCURAIINT0	USB0 OVRCURA Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode by the USB port 0 OVRCURA input. 1: The system has returned from deep software standby mode by the USB port 0 OVRCURA input.	R
b21	OVRCURBIINT0	USB0 OVRCURB Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode by the USB port 0 OVRCURB input. 1: The system has returned from deep software standby mode by the USB port 0 OVRCURB input.	R
b22	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b23	DVBINT0	USB0 VBUS Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode by the USB port 0 VBUS input. 1: The system has returned from deep software standby mode by the USB port 0 VBUS input.	R
b24	DPINT1	USB1 DP Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode by the USB port 1 DP input. 1: The system has returned from deep software standby mode by USB port 1 DP input.	R
b25	DMINT1	USB1 DM Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode by the USB port 1 DM input. 1: The system has returned from deep software standby mode by the USB port 1 DM input.	R
b27, b26	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b28	DOVRCRA1	USB1 OVRCURA Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode by the USB port 1 OVRCURA input. 1: The system has returned from deep software standby mode by the USB port 1 OVRCURA input.	R
b29	DOVRCRB1	USB1 OVRCURB Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode by the USB port 1 OVRCURB input. 1: The system has returned from deep software standby mode by the USB port 1 OVRCURB input.	R

Bit	Symbol	Bit Name	Description	R/W
b30	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b31	DVBINT1	USB1 VBUS Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode by the USB port 1 VBUS input. 1: The system has returned from deep software standby mode by the USB port 1 VBUS input.	R

DPUSR1R is used to control a recovery from deep software standby mode by a USB suspend/resume interrupt. DPUSR1R enables or disables recovery sources and indicates or clears the sources.

DPINTEn Bit (USBn DP Interrupt Enable/Clear) (n = 0 or 1)

The DPINTEn bit enables or disables a recovery from deep software standby mode by the DP input in the USB port n side. When 0 is written to this bit with the DPINTn bit set to 1, DPINTn is cleared to 0.

DMINTEn Bit (USBn DM Interrupt Enable/Clear) (n = 0 or 1)

The DMINTEn bit enables or disables a recovery from deep software standby mode by the DM input in the USB port n side. When 0 is written to this bit with the DMINTn bit set to 1, DMINTn is cleared to 0.

DOVRCRAEn Bit (USBn OVRCURA Interrupt Enable/Clear) (n = 0 or 1)

The DOVRCRAEn bit enables or disables a recovery from deep software standby mode by the OVRCURA input in the USB port n side. When 0 is written to this bit with the DOVRCRA flag set to 1, DOVRCRA flag is cleared to 0.

DOVRCRBen Bit (USBn OVRCURB Interrupt Enable/Clear) (n = 0 or 1)

The OVRCURB_INTEn bit enables or disables a recovery from deep software standby mode by the OVRCURB input in the USB port n side. When 0 is written to this bit with the DOVRCRB flag set to 1, DOVRCRB flag is cleared to 0.

DVBSEn Bit (USBn VBUS Interrupt Enable/Clear) (n = 0 or 1)

The VBUSTA_INTEn bit enables or disables a recovery from deep software standby mode by the VBUS input in the USB port n side. When 0 is written to this bit with the DVBINTn flag set to 1, DVBINT flag is cleared to 0.

DPINTn Flag (USBn DP Interrupt Source Recovery Flag) (n = 0 or 1)

The DPINTn flag indicates that the system has returned from deep software standby mode by the DP input in the USB port n side. A recovery from deep software standby mode by USBn DP input is enabled only when DPINTEn bit is 1.

This flag is cleared to 0 when 0 is written to DPINTEn bit with this flag set to 1.

DMINTn Flag (USBn DM Interrupt Source Recovery Flag) (n = 0 or 1)

The DMINTn flag indicates that the system has returned from deep software standby mode by the DM input in the USB port n side. Returning from deep software standby mode by USBn DM input is enabled only when DMINTEn bit is 1.

This flag is cleared to 0 when 0 is written to DMINTEn bit with this flag set to 1.

DOVCRAn Flag (USBn OVRCURA Interrupt Source Recovery Flag) (n = 0 or 1)

The DOVCRAn flag indicates that the system has returned from deep software standby mode by the OVRCURA input in the USB port n side. A recovery from deep software standby mode by USBn OVRCURA input is enabled only when DOVRCRAEn bit is 1.

This flag is cleared to 0 when 0 is written to OVRCURA_INTEn with this bit set to 1.

DOVCRBn Flag (USBn OVRCURB Interrupt Source Recovery Flag) (n = 0 or 1)

The DOVCRBn flag indicates that the system has returned from deep software standby mode by the OVRCURB input in the USB port n side. A recovery from deep software standby mode by USBn OVRCURB input is enabled only when DOVCRBEn bit is 1.

This flag is cleared to 0 when 0 is written to DOVCRBEn bit with this bit set to 1.

DVBINT Flag (USBn VBUS Interrupt Source Recovery Flag) (n = 0 or 1)

The DVBINT flag indicates that the system has returned from deep software standby mode by the VBUS input in the USB port n side. A recovery from deep software standby mode by USBn VBUS input is enabled only when DVBINTEn bit is 1.

This flag is cleared to 0 when 0 is written to DVBINTEn bit with this bit set to 1.

28.3 Operation

28.3.1 System Control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

28.3.1.1 Starting Operation

Setting the SYSCFG.USBE bit to 1 after starting the clock supply to the USB module (SYSCFG.SCKE = 1) enables and starts USB module operation.

28.3.1.2 Controller Function Selection

For the USB module, the host controller function or function controller function can be selected using the DCFM bit in SYSCFG. The DCFM bit should be modified in the initial settings immediately after a power-on reset or in the D+ pull-up-disabled (DPRPU = 0) and D+ and D- pull-down-disabled (DRPD = 0) state.

28.3.1.3 Example of USB External Connection Circuit

Figure 28.2 shows an example of OTG connection of the USB connector (USB0) in the self-powered state.

The USB module 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 function is selected and the DPRPU bit in SYSCFG is cleared 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.

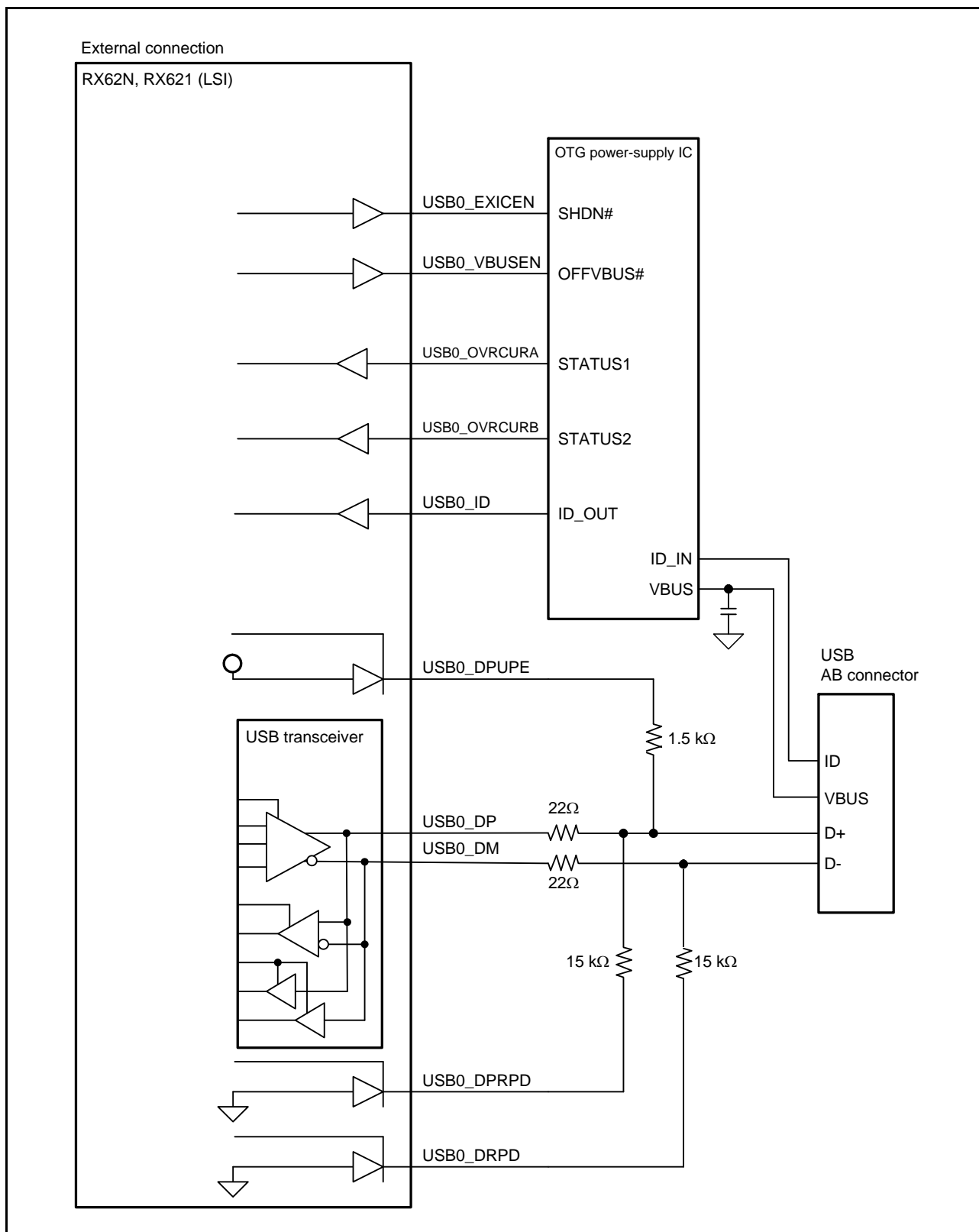


Figure 28.2 Sample OTG Connection of USB Connector (USB0) in Self-Powered State

Figure 28.3 shows an example of functional connection sample of the USB connector (USB0) in the self-powered state.

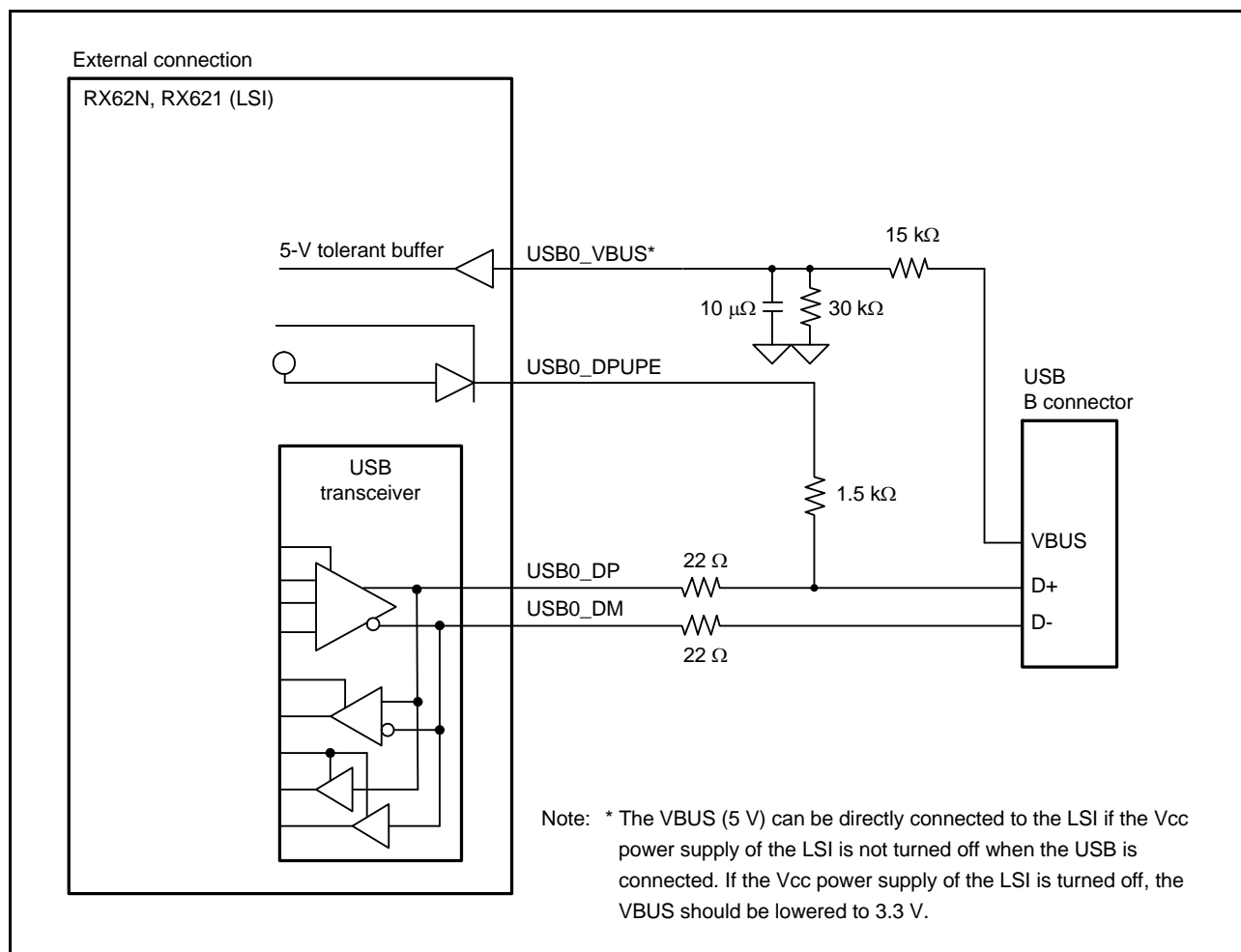


Figure 28.3 Functional Connection Sample of USB Connector (USB0) in Self-Powered State

Figure 28.4 shows an example of host connection of the USB connector (USB0).

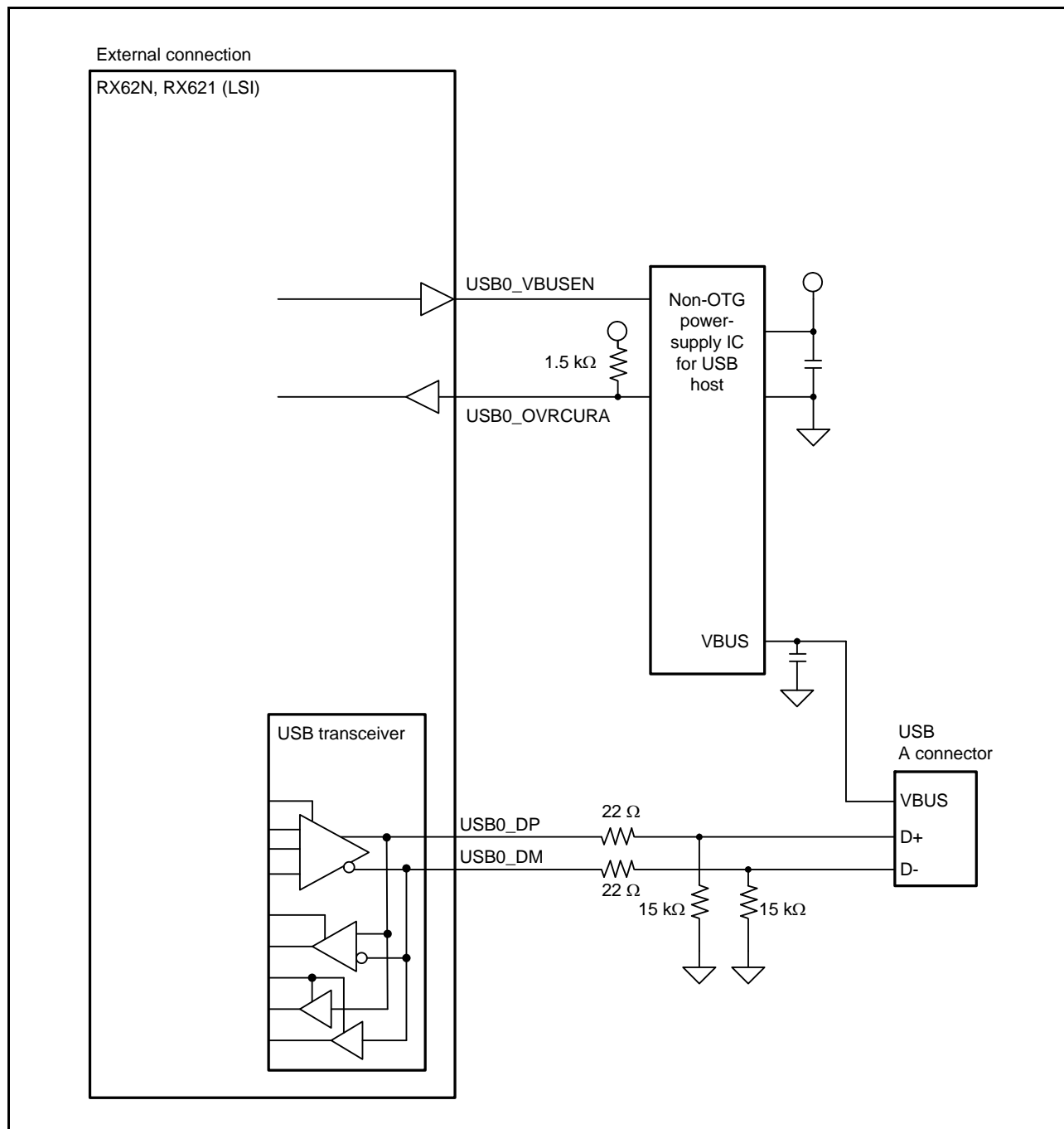


Figure 28.4 Sample Host Connection of USB Connector (USB0)

Figure 28.5 shows an example of functional connection sample of the USB connector (USB0) in the bus-powered state.

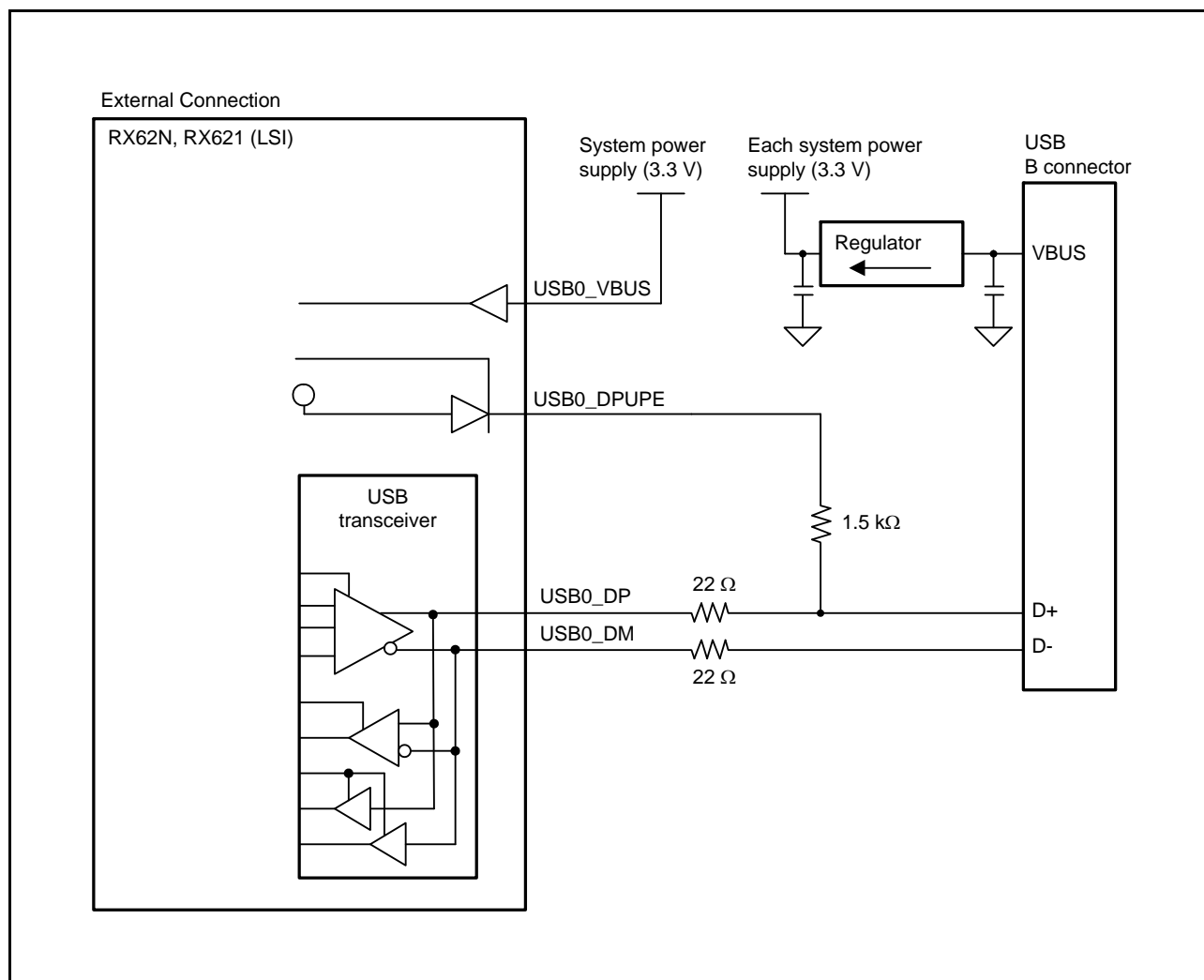


Figure 28.5 Functional Connection Sample of USB Connector (USB0) in Bus-Powered State

The examples of external circuits described in this section are only simplified circuits, and their operations are not guaranteed under any system.

28.3.1.4 Canceling Deep Software Standby Mode by a USB Suspend/Resume Interrupt

Deep software standby mode can be canceled by a USB suspend/resume interrupt.

A USB suspend/resume interrupt is detected at the USB resume detecting unit. The USB resume detecting unit controls and monitors the I/O pins for USB0 and USB1 to detect USB suspend/resume interrupts.

Figure 28.6 shows a schematic diagram of connection between the USB resume detecting unit and the I/O pins for USB0 and USB1.

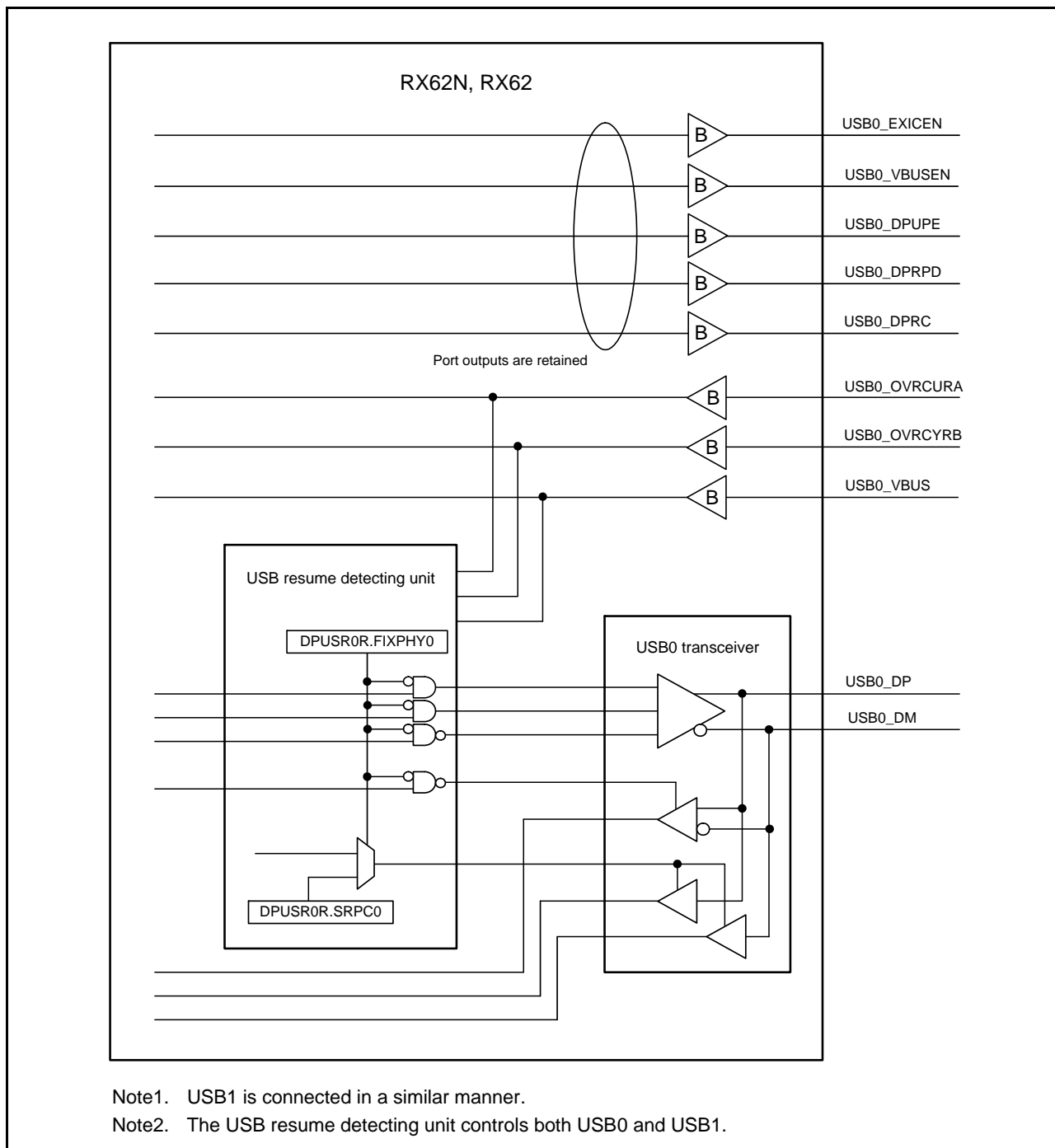


Figure 28.6 Overview of Connection between USB Resume Detecting Unit and USB0/USB1 I/O Pins

Table 28.14 shows the USB suspend/resume interrupt sources and their corresponding I/O pins.

Table 28.14 USB Suspend/Resume Interrupt Sources and Corresponding I/O Pins

USB Operating Mode	Source	Pin Name
Function/OTG	Resume	USB0_DP / USB1_DP
Host controller function is selected or in OTG mode	Connection/Disconnection	USB0_DP/USB0_DM USB1_DP/USB1_DM
Function controller function is selected	Connection/Disconnection	USB0_VBUS/USB1_VBUS
Host controller function is selected	Overcurrent detection	USB0_OVRCURA/USB1_OVRCURA
OTG mode	Overcurrent detection	USB0_OVRCURA/USB0_OVRCURB USB1_OVRCURA/USB1_OVRCURB

When canceling deep software standby mode using a USB suspend/resume interrupt, set the DPSBYCR.IOKEEP bit to have the outputs of the I/O ports retained.

Figure 28.7 shows a flowchart for setting the USB when entering deep software standby mode while the host controller function or function controller function is selected. Figure 28.8 shows a flowchart for setting the USB when canceling deep software standby mode while the host controller function is selected. Figure 28.9 shows a flowchart for setting the USB when canceling deep software standby mode while the function controller function is selected.

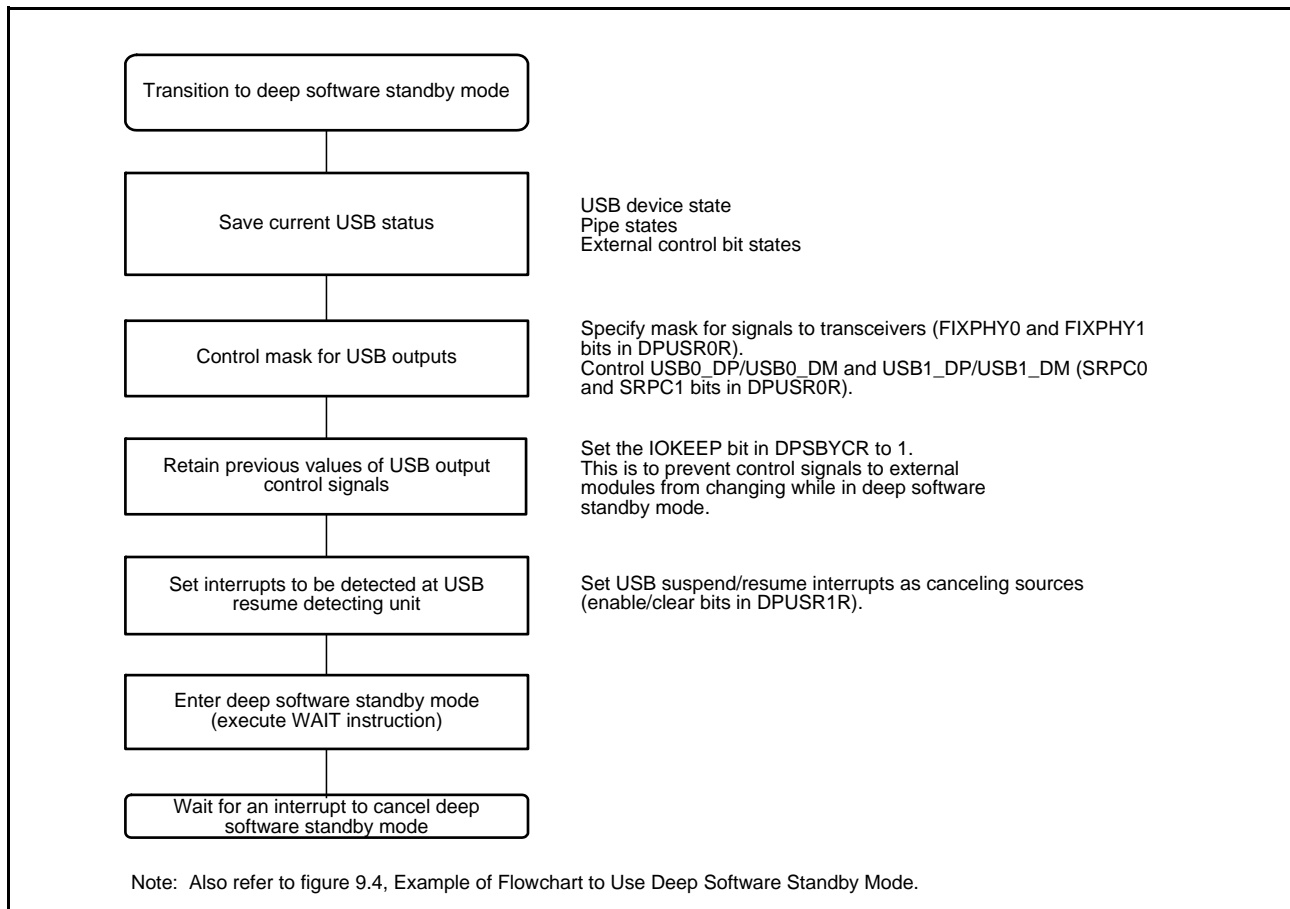


Figure 28.7 USB Setting Flowchart for Transition to Deep Software Standby Mode as USB Host/Function Controller

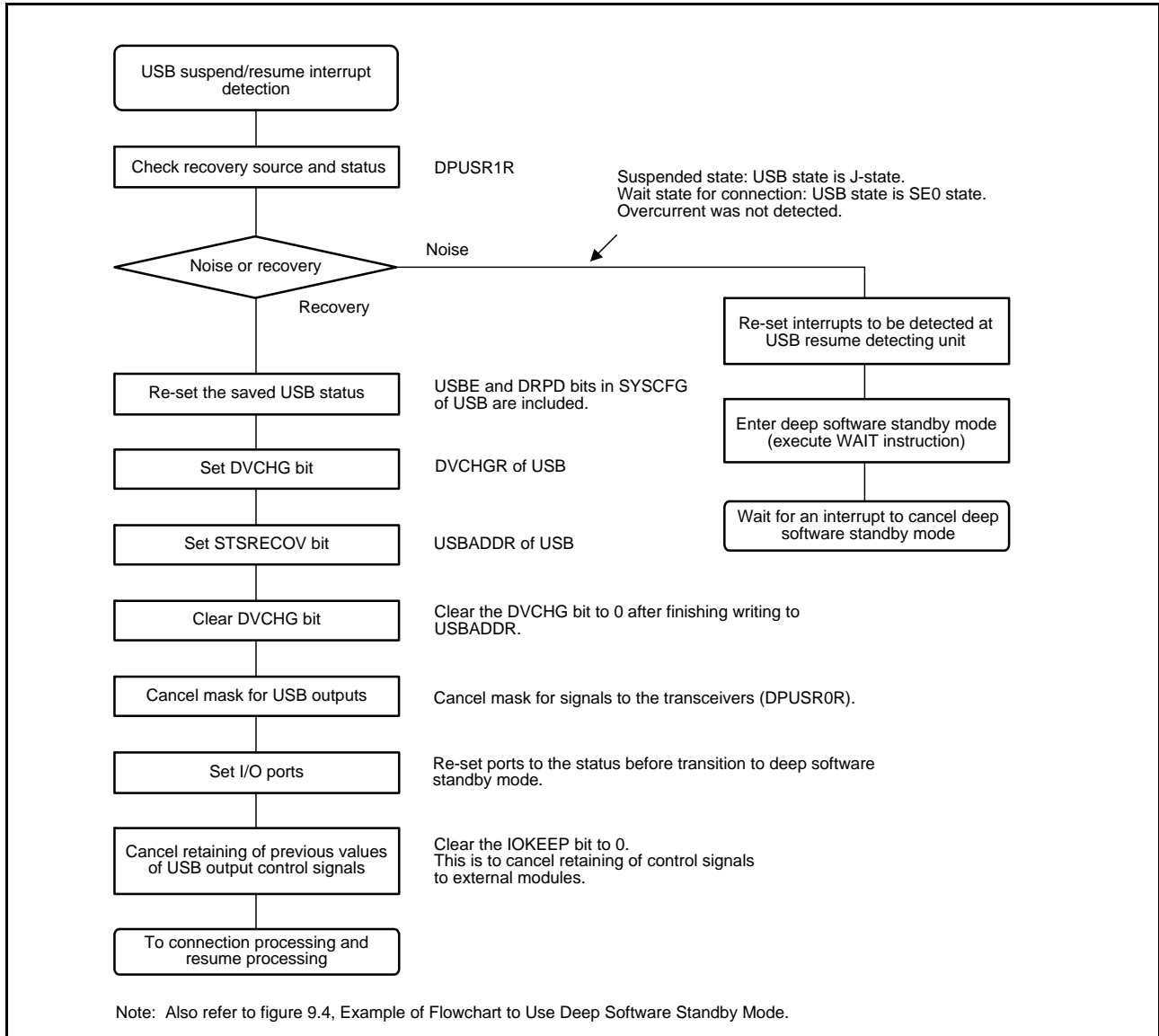


Figure 28.8 USB Setting Flowchart for Canceling Deep Software Standby Mode as USB Host Controller

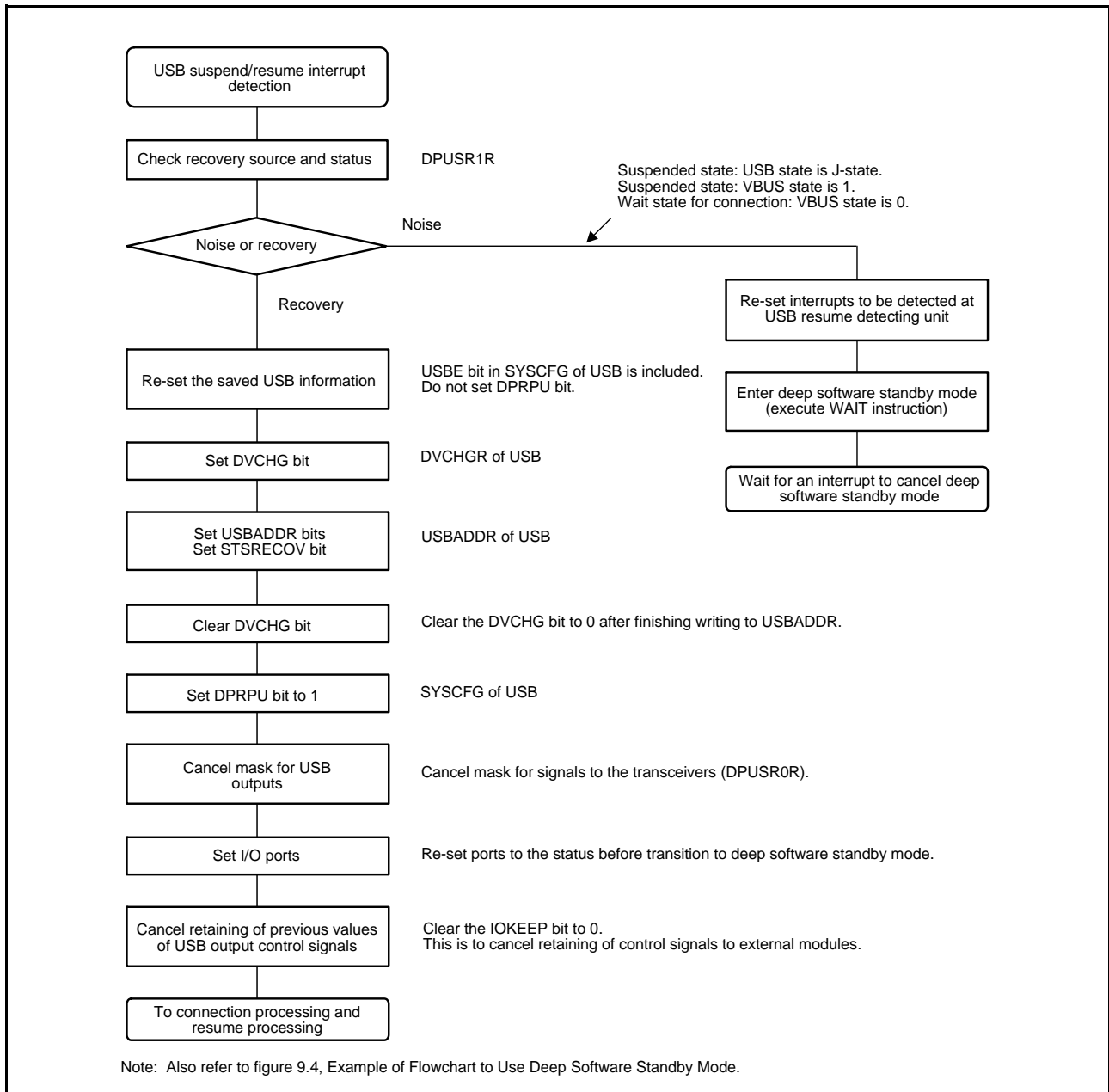


Figure 28.9 USB Setting Flowchart for Canceling Deep Software Standby Mode as USB Function Controller

28.3.2 Interrupt Sources

Table 28.15 lists the interrupt sources in the USB module.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, the USB issues a USB interrupt request to the interrupt controller (ICU) and an USB interrupt will be generated.

Table 28.15 Interrupt Sources (1 / 2)

Bit to be Set	Name	Interrupt Source	Function That Generates the Interrupt	Status Flag
VBINT	VBUS interrupt	<ul style="list-style-type: none"> When a change in the state of the USBm_VBUS input pin has been detected (low to high or high to low) 	Host/function*1	VBSTS
RESM	Resume interrupt	<ul style="list-style-type: none"> 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) 	Function	—
SOFR	Frame number update interrupt	<p>[Host controller function is selected]</p> <ul style="list-style-type: none"> When an SOF packet with a different frame number has been transmitted <p>[Function controller function is selected]</p> <ul style="list-style-type: none"> When an SOF packet with a different frame number has been received 	Host/function	—
DVST	Device state transition interrupt	<ul style="list-style-type: none"> When a device state transition has been detected (any of the following conditions) A USB bus reset detected Suspend state detected SET_ADDRESS request received SET_CONFIGURATION request received 	Function	DVSQ
CTRT	Control transfer stage transition interrupt	<ul style="list-style-type: none"> When a stage transition has been detected in control transfer (any of the following conditions) Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed A control transfer sequence error occurred 	Function	CTSQ
BEMP	Buffer empty interrupt	<ul style="list-style-type: none"> When transmission of all data in the buffer memory has been completed and the buffer has become empty When a packet larger than the maximum packet size has been received 	Host/function	BEMPSTS. PIPEBEMP
NRDY	Buffer not ready interrupt	<p>[Host controller function is selected]</p> <ul style="list-style-type: none"> When STALL has been received from the peripheral device for the issued token 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) When an overrun/underrun occurred during isochronous transfer <p>[Function controller function is selected]</p> <ul style="list-style-type: none"> When NAK has been returned for an IN or OUT token while the PID bit = BUF When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer When an overrun/underrun occurred during data reception in isochronous transfer 	Host/function	NRDYSTS. PIPENRDY
BRDY	Buffer ready interrupt	<ul style="list-style-type: none"> When the buffer has become ready (reading or writing is enabled) 	Host/function	BRDYSTS. PIPEBRDY

Table 28.15 Interrupt Sources (2 / 2)

Bit to be Set	Name	Interrupt Source	Function That Generates the Interrupt	Status Flag
OCRCR	Overcurrent input change interrupt	• When a change in the state of the USBm_OVRCURA or USBm_OVRCURB input pin has been detected (low to high or high to low)	Host	OVCMON
BCHG	Bus change interrupt	• When a change of USB bus state has been detected	Host/function	SYSSTS0 LNST
DTCH	Disconnection detection during full-speed operation	• When disconnection of a peripheral device has been detected in full-speed operation	Host	DCSTCTR0 RHST
ATTCH	Device connection detection	• When J-state or K-state is detected on the USB port for 2.5 μ s. Used for checking whether a peripheral device is connected.	Host	—
EOFERR	EOF error detection	• When an EOF error of a peripheral device has been detected	Host	—
SACK	Normal setup operation	• When the normal response (ACK) for the setup transaction has been received	Host	—
SIGN	Setup error	• When a setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	—

Note: All bits without register name are in INTSTS0.

Note 1. Though this interrupt can be generated while the host function is selected, it is not usually used with the host function.

Figure 28.10 shows the circuits related to the interrupts in the USBm.

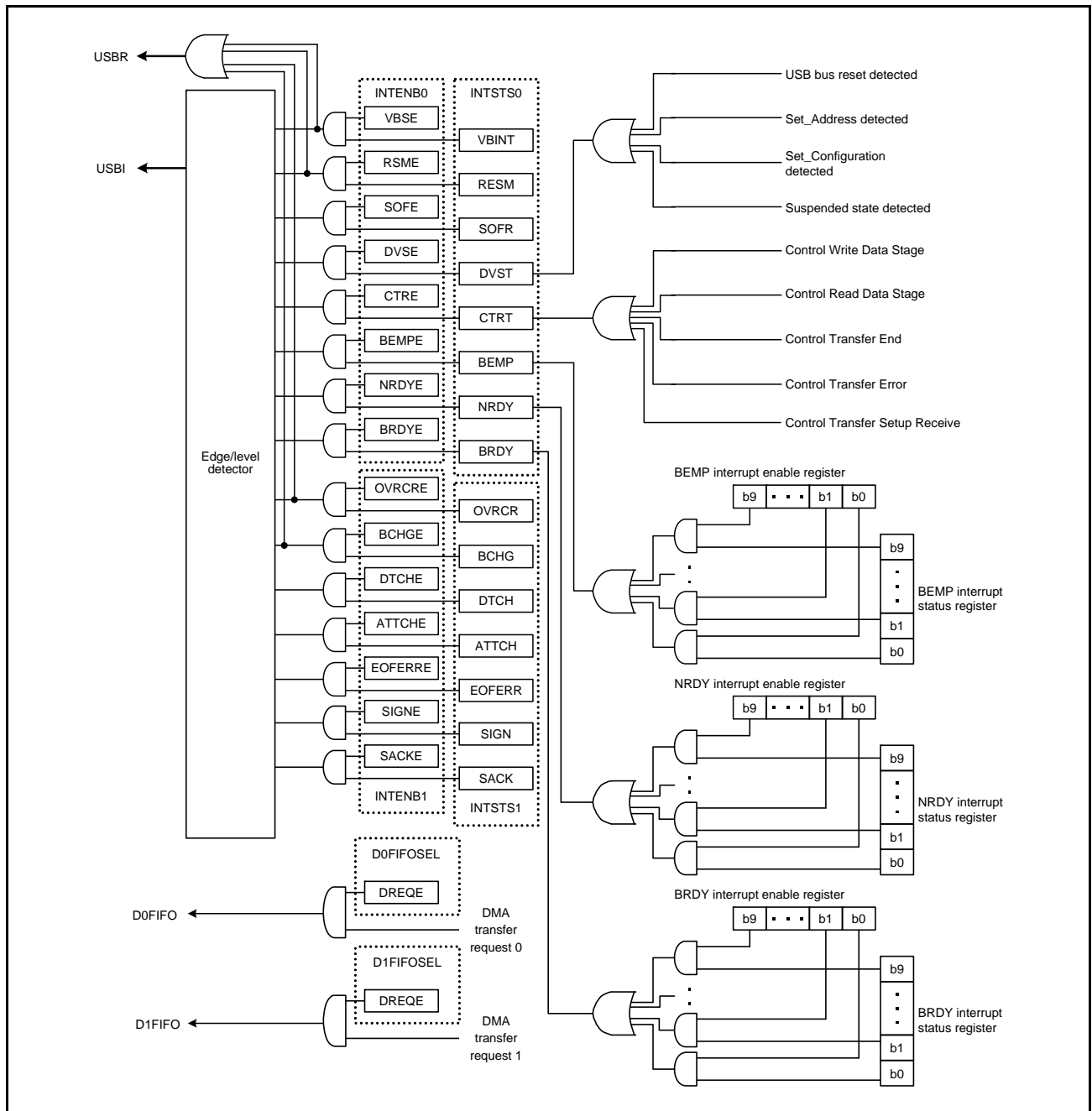


Figure 28.10 Circuits Related to Interrupts

Table 28.16 shows the interrupts generated in the USB_m (m = 0 or 1).

Table 28.16 USB_m Interrupts

Interrupt Name	Interrupt Flag	DTC Activation	DMACA Activation	Priority
D0FIFO	DMA transfer request 0	Possible	Possible	High
D1FIFO	DMA transfer request 1	Possible	Possible	↑
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, and setup error	Not possible	Not possible	
USBR*1	VBUS interrupt, resume interrupt, overcurrent input change interrupt, and bus change interrupt	Not possible	Not possible	

Note 1. Release from all-module clock-stop mode or from software standby mode is possible.

28.3.3 Interrupt Descriptions

28.3.3.1 BRDY Interrupt

The BRDY interrupt is generated when either of the host controller function or function controller function is selected. The following shows the conditions under which the USB module sets 1 to a corresponding bit in BRDYSTS. Under this condition, the USB module generates a BRDY interrupt if software has set 1 to the PIPEBRDYE bit in BRDYENB that corresponds to the pipe and 1 to the BRDYE bit in INTENB0.

The conditions for generating and clearing the BRDY interrupt depend on the settings of the BRDYM bit and BFRE bit for each pipe as described below.

(1) When BRDYM = 0 and BFRE = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, the USB module generates an internal BRDY interrupt request trigger and sets 1 to the PIPEBRDY bit corresponding to the pertinent pipe.

For the pipe in the transmitting direction:

- When software changes the DIR bit from 0 to 1.
- When packet transmission is completed using the pertinent pipe while write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the 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 ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP (that is, during data transmission for control transfers).

For the pipe in the receiving direction:

- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0). No request trigger is generated for the transaction in which DATA-PID disagreement has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode. No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

When the function controller function 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 cleared to 0 by writing 0 to the corresponding PIPEBRDY bit in BRDYSTS through software. In this case, 1s should be written to the PIPEBRDY bits for the other pipes.

Be sure to clear the BRDY status before accessing the FIFO buffer.

(2) When BRDYM = 0 and BFRE = 1

With these settings, the USB module generates a BRDY interrupt on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the bit in BRDYSTS corresponding to the pertinent pipe.

On any of the following conditions, the USB module determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the transaction counter register (TRNCNT bits) is used and the number of packets specified by the TRNCNT bits are completely received.

When the pertinent data is completely read out after any of the above conditions has been satisfied, the USB module determines that all data for a single transfer has been completely read out.

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 out upon passing the zero-length packet data to the CPU. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding FIFOCTR register through software.

With these settings, the USB module does not detect a BRDY interrupt for the pipe in the transmitting direction.

The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY bit through software. In this case, 1s should be written to the PIPEBRDY bits for the other pipes.

In this mode, the BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the ACLRM bit.

Figure 28.11 shows the timing of BRDY interrupt generation.

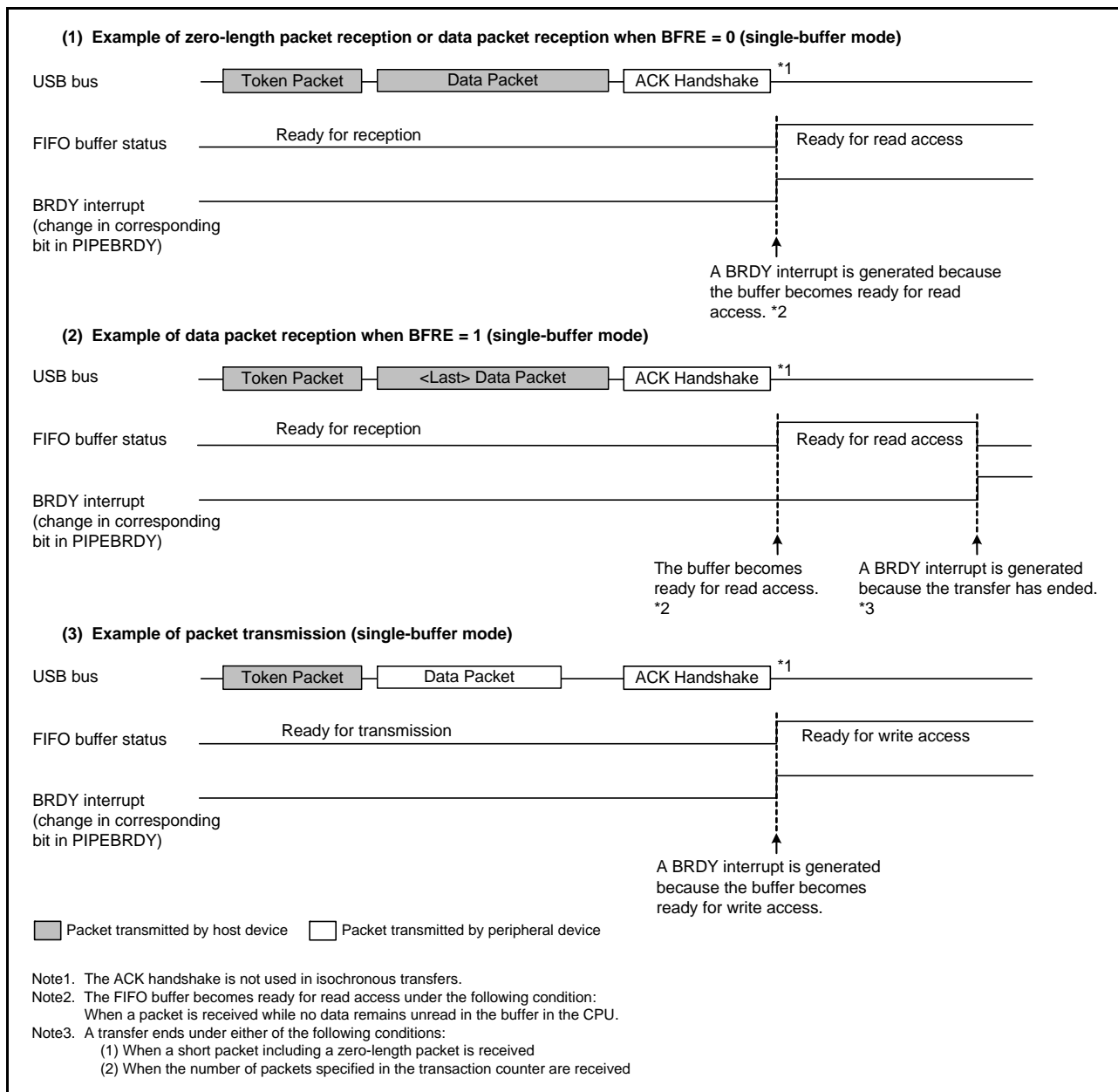


Figure 28.11 Timing of BRDY Interrupt Generation

The condition that USB module clears the BRDY bit in INTSTS0 depends on the SOFCFG.BRDYM bit setting. Table 28.17 shows the condition for clearing the BRDY bit.

Table 28.17 Condition for Clearing BRDY Bit

BRDYM	Condition for Clearing BRDY Bit
0	The USB module clears the BRDY bit in INTSTS0 when software has cleared all bits in BRDYSTS.
1	The USB module clears the BRDY bit in INTSTS0 when the BSTS bits for all piles have become 0.

28.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, the USB module sets the corresponding PIPENRDY bit in NRDYSTS to 1. If the corresponding bit in NRDYENB has been set to 1 by software, the USB module sets the NRDY bit in INTSTS0 to 1 and generates a USB interrupt.

The following describes the conditions on which the USB module generates the internal NRDY interrupt request for a given pipe.

Note that the internal NRDY interrupt request is not generated during setup transaction execution when the host controller function is selected. During setup transactions when the host controller function 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 function is selected.

(1) When Host Controller Function is Selected

- For the pipe in the transmitting direction:

On any of the following conditions, the USB module 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 module transmits a zero-length packet following the OUT token and sets the corresponding PIPENRDY bit and the 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 module sets the corresponding PIPENRDY bit to 1 and modifies the setting of the PID 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 module sets the corresponding PIPENRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to STALL (11b).

- 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 module discards the received data for the IN token and sets the PIPENRDY 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 module also sets the 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 module (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 module sets the PIPENRDY bit corresponding to the pipe to 1 and modifies the setting of the PID 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 module sets the PIPENRDY bit corresponding to the pipe to 1. (The setting of the PID 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 module sets the PIPENRDY bit corresponding to the pipe and the CRCE bit to 1.
- When the STALL handshake is received.
In this case, the USB module sets the PIPENRDY bit corresponding to the pipe to 1 and modifies the setting of the PID bits of the corresponding pipe to STALL.

(2) When Function Controller Function is Selected

- For the pipe in the transmitting direction:
 - When an IN token is received while there is no data to be transmitted in the FIFO buffer.
In this case, the USB module generates a NRDY interrupt request at the reception of the IN token and sets the PIPENRDY bit to 1.
For the pipe for the isochronous transfers in which an interrupt is generated, the USB module transmits a zero-length packet and sets the OVRN bit to 1.
- For the pipe in the receiving direction:
 - When an OUT token is received while there is no space available in the FIFO buffer.
For the pipe for the isochronous transfers in which an interrupt is generated, the USB module generates a NRDY interrupt request at the reception of the OUT token and sets the PIPENRDY bit to 1 and OVRN bit to 1.
For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, the USB module generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPENRDY bit to 1.
However, during re-transmission (due to DATA-PID disagreement), the NRDY interrupt request is not generated.
In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.
- For the pipe for isochronous transfers, when a token is not received successfully within an interval frame.
In this case, the USB module generates a NRDY interrupt request when SOF is received, and sets the PIPENRDY bit to 1.

Figure 28.12 shows the timing of NRDY interrupt generation when the function controller function is selected.

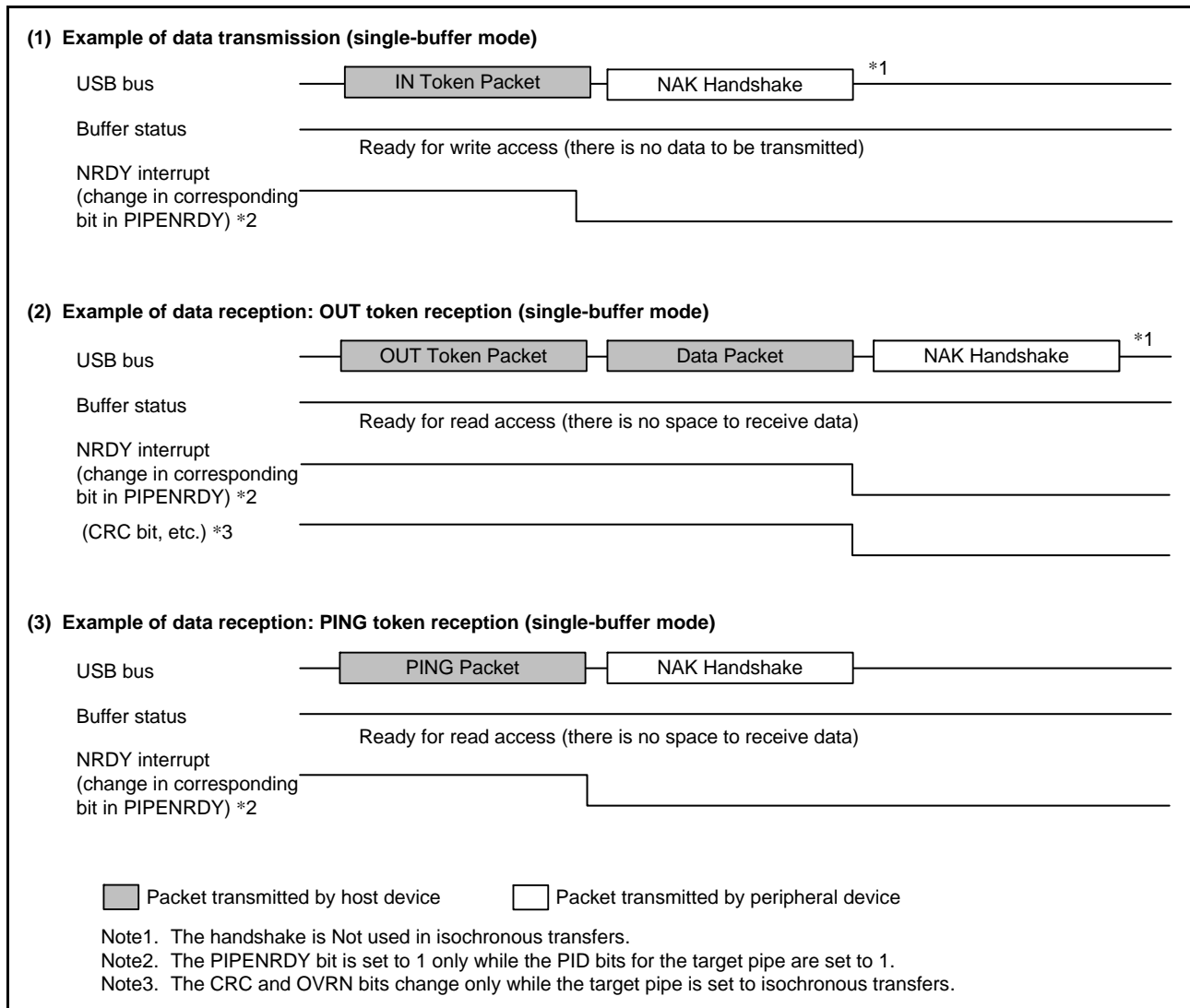


Figure 28.12 Timing of NRDY Interrupt Generation (When Function Controller Function is Selected)

28.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by software, the USB module sets the corresponding PIPEBEMP bit in BEMPSTS to 1. If the corresponding bit in BEMPENB has been set to 1 by software, the USB module sets the BEMP bit in INTSTS0 to 1 and generates a USB interrupt.

The following describes the conditions on which the USB module generates an internal BEMP interrupt request.

- **For the pipe in the transmitting direction:**

- When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).

In single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP. However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When CPU (DTC or DMACA) 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 ACLRM or BCLR bit to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage while the function controller function is selected.

- **For the pipe in the receiving direction:**

In this case, the USB module generates a BEMP interrupt request, sets the corresponding BEMPSTS.PIPEnBEMP bit to 1, discards the received data, and modifies the setting of the PID bits of the corresponding pipe to STALL (11b).

Here, the USB module 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 28.13 shows the timing of BEMP interrupt generation when the function controller function is selected.

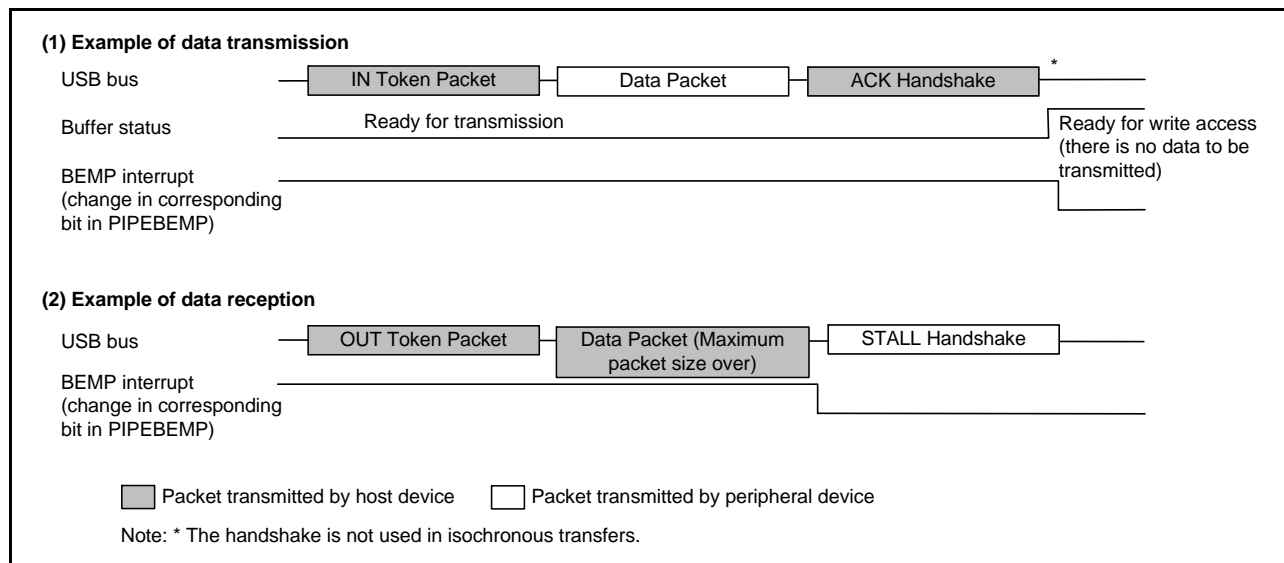


Figure 28.13 Timing of BEMP Interrupt Generation (When Function Controller Function is Selected)

28.3.3.4 Device State Transition Interrupt

Figure 28.14 is a diagram of device state transitions in the USB module. The USB module controls device state and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using INTENB0. The device state to which a transition was made can be confirmed using the DVSQ bits in INTSTS0. When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

Device state can be controlled only when the function controller function is selected. The device state transition interrupts can also be generated only when the function controller function is selected.

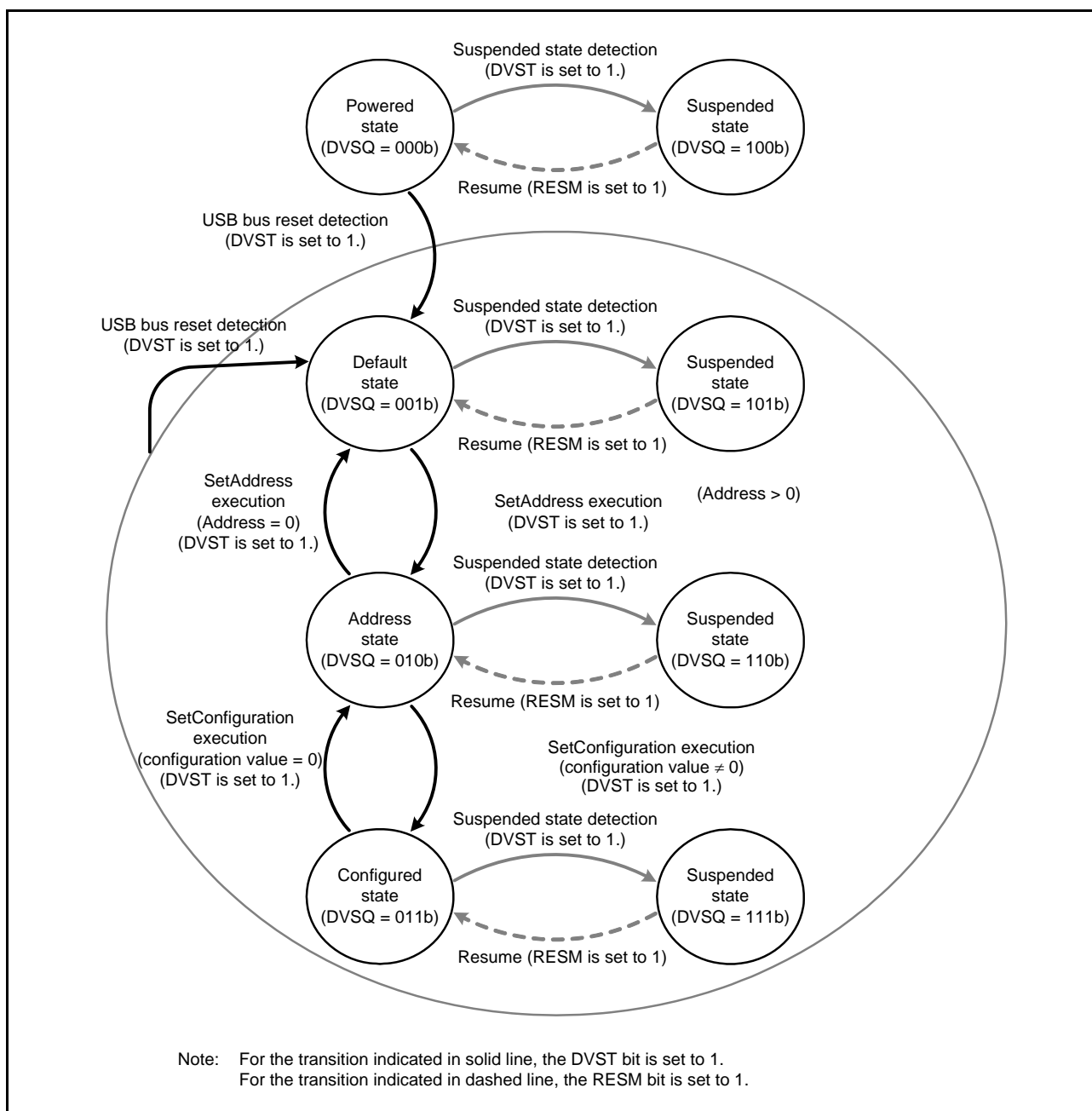


Figure 28.14 Device State Transitions

28.3.3.5 Control Transfer Stage Transition Interrupt

Figure 28.15 is a diagram of control transfer stage transitions in the USB module. The USB module controls the control transfer sequence and generates control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually using INTENB0. The transfer stage to which a transition was made can be confirmed using the CTSQ bits in INTSTS0.

Control transfer stage transition interrupts are generated only when the function controller function is selected.

The control transfer sequence errors are listed below. If an error occurs, the PID bits in DCPCTR are set to 1xb (STALL response).

During control read transfer:

- An OUT token is received while no data has been transferred for the IN token at the data stage.
- An IN token is received at the status stage.
- A data packet with DATAPID = DATA0 is received at the status stage.

During control write transfer:

- An IN token is received while no ACK response has been returned for the OUT token at the data stage.
- A data packet with DATAPID = DATA0 is received for the first data packet at the data stage.
- An OUT token is received at the status stage

During no-data control transfers:

- An OUT token is received at the status stage.

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (SERR = 1), the CTSQ = 110 value is retained until CTRT = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ = 110 is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB module retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

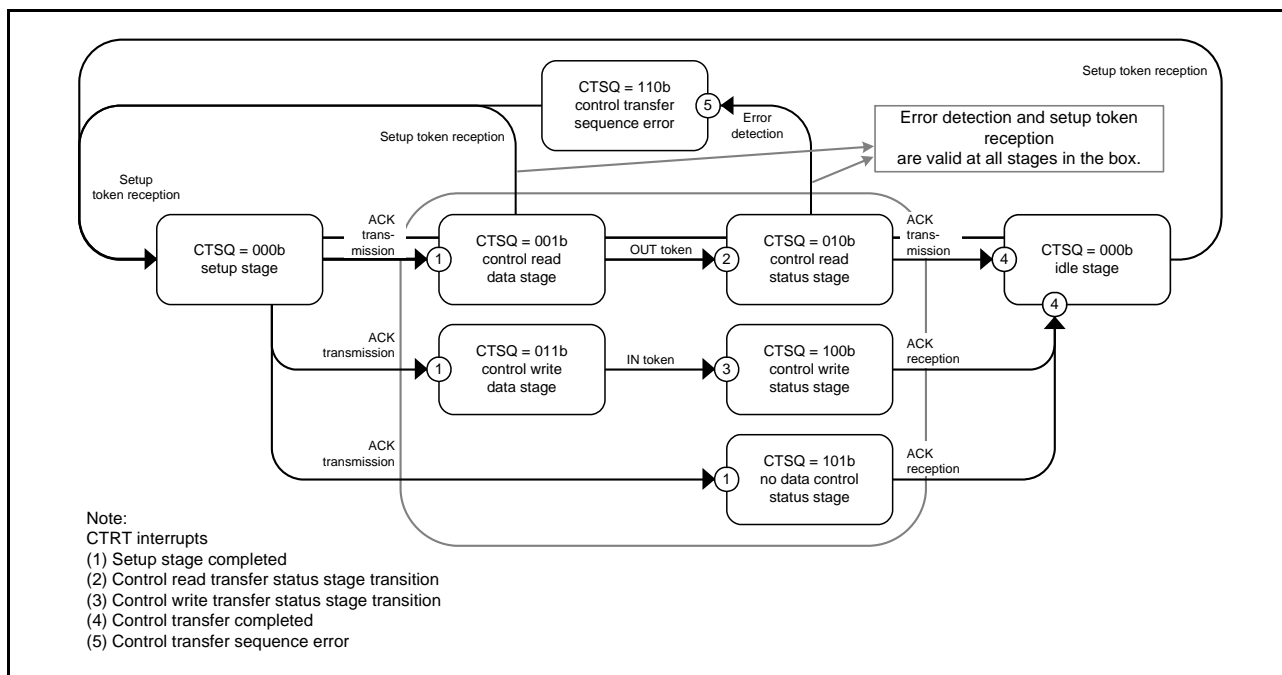


Figure 28.15 Control Transfer Stage Transitions

28.3.3.6 Frame Update Interrupt

With the host controller function selected, an interrupt is generated at the timing when the frame number is updated. With the function controller function selected, an SOFR interrupt is generated when the frame number is updated.

When the function controller function is selected, the USB module updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

28.3.3.7 VBUS Interrupt

When the USBm_VBUS pin level changes, a VBUS interrupt is generated. The level of the USBm_VBUS pin can be checked with the VBSTS bit in INTSTS0. 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 USBm_VBUS pin level.

28.3.3.8 Resume Interrupt

When the function controller function 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 function is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

28.3.3.9 OVRCCR Interrupt

For port 0, an OVRCCR interrupt is generated when the USBm_OVRCURA or USBm_OVRCURB pin level has changed. The levels of the USBm_OVRCURA and USBm_OVRCURB pins can be checked with the OVCMON[1:0] bits in SYSSTS0. The external power-supply IC can check whether overcurrent has been detected using the OVRCCR interrupt.

For On-The-Go connection, whether a change has been detected in the VBUS comparator can be checked using the OVRCCR interrupt.

28.3.3.10 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 function is selected. The BCHG interrupt is generated regardless of whether the host controller function or function controller function is selected.

28.3.3.11 DTCH Interrupt

A DTCH interrupt is generated when disconnection of the USB bus is detected while the host controller function is selected. The USB module detects bus disconnection based on USB Specification 2.0.

After detecting a DTCH interrupt, the USB module controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). Software should terminate all pipes in which communications are currently carried out for the pertinent port and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies the 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.

28.3.3.12 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 function selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

28.3.3.13 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 function selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

28.3.3.14 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 μ s with the host controller function 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 μ s.
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 μ s.

28.3.3.15 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 module controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). Software should terminate all pipes in which communications are currently carried out for the pertinent port and perform re-enumeration of the pertinent port.

Modifies the DVSTCTRn.UACT bit for the port in which an EOFERR interrupt has been detected to 0 (n = 0 or 1).

Puts the port in which an EOFERR interrupt has been generated into the idle state.

28.3.4 Pipe Control

Table 28.18 lists the pipe setting items in the USB module. With USB data transfer, data transfer has to be carried out using the logic pipe called the endpoint. The USB module has ten pipes that are used for data transfer.

Appropriate settings should be made for each of the pipes according to the specifications of the system.

Table 28.18 Pipe Setting Items

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 0000 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 function is selected.
	MXPS	Maximum packet size	Compliant with the USB standard.
PIPEPERI	IFIS	Buffer flush	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE9: Cannot be set
	IITV	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 function has been selected)
DCPCTR PIPECTR	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 PIPE3 to PIPE5.
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller function has been selected.
	SUREQCLR	SUREQ clear	Can be set only for the DCP. Can be controlled only when the host controller function has been selected.
	ATREPM	Auto response mode	PIPE1 to PIPE5: Can be set Can be set only when the function controller function 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	See section 28.3.4.6, 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

28.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is disabled (PID = NAK).

Registers that Should Not be Set in the USB Communication Enabled (PID = BUF) State:

- Bits in DCPCFG and DCPMAXP
- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEMAXP, and PIPEPERI
- ATREPM, ACLRM, SQCLR, and SQSET bits in PIPExCTR
- Bits in PIPExTRE and PIPExTRN

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 corresponding to the pipe to NAK.
3. Wait until the corresponding PBUSY bit is cleared to 0.
4. Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Registers that Should Not be Set When CURPIPE in FIFO-PORT is set:

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEMAXP and PIPEPERI

In order to modify pipe information, the CURPIPE 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 BCLR after the pipe information is modified.

28.3.4.2 Transfer Types

The TYPE bits in PIPEPCFG 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.

28.3.4.3 Endpoint Number

The EPNUM bits in PIPEPCFG are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- DCP: No setting is necessary (fixed at end point 0).
- PIPE1 to PIPE9: The endpoint numbers from 1 to 15 should be selected and set. These should be set so that the combination of the DIR bit and EPNUM bits is unique.

28.3.4.4 Maximum Packet Size Setting

The MXPS bits in DCPMAXP and PIPEMAXP are used to specify the maximum packet size for each pipe. DCP and PIPE1 to PIPE5 can be set to any of the maximum pipe sizes defined by the USB specification. For PIPE6 to PIPE9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

- DCP: Set 8, 16, 32, or 64.
- PIPE1 to PIPE5: Set 8, 16, 32, or 64 when using bulk transfer.
- PIPE1 and PIPE2: Set a value between 1 and 256 when using isochronous transfer.
- PIPE6 to PIPE9: Set a value between 1 and 64.

28.3.4.5 Transaction Counter (For PIPE1 to PIPE5 in Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USB module recognizes that the transfer has ended. Two transaction counters are provided: one is the TRNCNT 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 SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the PID of the corresponding PIPE is set to NAK and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the TRCLR bit. The information read from TRNCNT differs depending on the setting of the TRENB bit.

- TRENB = 0: The specified transaction counter value can be read.
- TRENB = 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.

28.3.4.6 Response PID

The PID bits in DCPCTR and PIPEnCTR are used to set the response PID for each pipe.

The following shows the USB module operation with various response PID settings:

- Response PID settings when the host controller function 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 SUREQ bit.

- Response PID settings when the function controller function is selected:

The response PID is used to specify the response to transactions from the host.

- NAK setting: The NAK response is always 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 always returned in response to the generated transaction.

Note: For setup transactions, an ACK response is always returned regardless of the PID setting, and the USB request is stored in the register.

The USB module may write to the PID bits, depending on the results of the transaction as described below.

- When the host controller function 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, see section 28.3.3.2, NRDY Interrupt.)
 - If a short packet is received when the SHTNAK bit in PIPECFG 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 module.
 - 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.
- When the function controller function has been selected and the response PID is set by hardware:
 - NAK setting: In the following cases, PID = NAK is set and NAK is always 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 SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.
 - BUF setting: There is no BUF writing by the USB module.
 - STALL setting: In the following cases, PID = STALL is set and STALL is always 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).

28.3.4.7 Data PID Sequence Bit

The USB module automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit switches at the timing of ACK handshake reception. When data is received, the sequence bit switches at the timing of ACK handshake transmission. The SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

When the function controller function has been selected and control transfer is used, the USB module automatically sets the sequence bit when a stage transition is made. DATA0 is returned when the setup stage is ended and DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller function 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 function or function controller function is selected.

28.3.4.8 Response PID = NAK Function

The USB module has a function that disables pipe operation (PID response = NAK) at the timing at which the final data packet of a transaction is received (the USB module automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the SHTNAK bit in PIPECFG 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.

28.3.4.9 Auto Response Mode

With the pipes for bulk transfer (PIPE1 to PIPE5), when the ATREPM bit in PIPEnCTR is set to 1, a transition is made to auto response mode. During an OUT transfer (DIR = 0), OUT-NAK mode is entered, and during an IN transfer (DIR = 1), null auto response mode is entered.

28.3.4.10 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 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.

28.3.4.11 Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the 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, 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 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 μ 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).

28.3.5 FIFO Buffer Memory

28.3.5.1 FIFO Buffer Memory

The USB module has FIFO buffer memory for data transfer. The memory area used for each pipe is managed by the USB module. The FIFO buffer memory has two states depending on whether the access right is assigned to the system (CPU side) or the USB module (SIE side).

- Buffer Status

Table 28.19 and Table 28.20 show the buffer status in the USB module. The buffer memory status can be confirmed using the BSTS bit in DCPCTR and the INBUFM bit in PIPEnCTR. The access direction for the buffer memory can be specified using either the DIR bit in PIPEnCFG or the ISEL bit in CFIFOSEL (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 (DTC or DMACA) is slow, software can use the INBUFM bit to confirm the end of transmission.

Table 28.19 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 28.20 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.

28.3.5.2 FIFO Buffer Clearing

Table 28.21 shows the clearing of the FIFO buffer memory by the USB module. The buffer memory can be cleared using the BCLR, DCLRM, and ACLRM bits in the port control register.

Table 28.21 List of Buffer Clearing Methods

Item	FIFO Buffer Clearing Mode		
	Clearing Buffer Memory on CPU Side	Mode for Automatically Clearing Buffer Memory after Reading Specified Pipe Data	Auto Buffer Clear Mode for Discarding All Received Packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(1) Auto Buffer Clear Mode Function

With the USB module, all received data packets are discarded if the ACLRM bit in PIPEnCTR 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.

However, an access cycle of at least 100 ns is required for the internal hardware sequence processing time between ACLRM = 1 and ACLRM = 0.

(2) Buffer Memory Specifications (Single or Double Setting)

Either a single or double buffer configuration can be selected for PIPE1 to PIPE5, using the DBLB bit in PIPEnCFG.

28.3.5.3 FIFO Port Functions

Table 28.22 shows the settings for the FIFO port functions of the USB module. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, the BVAL bit in C/DnFIFOCTR should be set to end writing. To send a zero-length packet, the BCLR bit in the register should be used to clear the buffer and then the BVAL bit set in order to end writing. In read access, reception of new packets is automatically enabled when all data has been read. Data cannot be read when a zero-length packet has been received (DTLN = 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 bits in C/DnFIFOCTR.

Table 28.22 FIFO Port Function Settings

Register Name	Bit Name	Function
CFIFOSEL, DnFIFOSEL (n = 0 or 1)	RCNT	Selects DTLN read mode.
	REW	Buffer memory rewind (re-read, rewrite).
	DCLRM	Automatically clears receive data for a specified pipe after the data has been read (only for DnFIFO).
	DREQE	Enables DMA transfers (only for DnFIFO).
	MBW	FIFO port access bit width.
	BIGEND	Selects FIFO port endian.
	ISEL	FIFO port access direction (only for DCP).
	CURPIPE	Selects the current pipe.
CFIFOCTR, DnFIFOCTR (n = 0 or 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 28.23 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed should be selected using the CURPIPE bits in C/DnFIFOSEL. After the pipe is selected, whether the written value can be correctly read from the CURPIPE bits should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by the USB controller.) Then, the FIFO port can be accessed after FRDY = 1 is checked. In addition, the bus width to be accessed should be selected using the MBW bit. The buffer memory access direction conforms to the DIR bit in PIPEnCFG. Only for the DCP, the ISEL bit determines the direction.

Table 28.23 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 or DMACA 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 C/DnFIFOSEL is used for this processing.

If a pipe is selected through the CURPIPE bits in C/DnFIFOSEL 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, FRDY = 1 should be checked after selecting a pipe.

28.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA Transfers

For PIPE1 to PIPE9, the FIFO port can be accessed using the DTC or DMACA*1. When accessing the buffer for the pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.

The unit of transfer to the FIFO port should be selected using the MBW bit in DnFIFOSEL and the pipe targeted for the DMA transfer should be selected using the CURPIPE bits. The selected pipe should not be changed during the DMA transfer.

(2) DnFIFO Auto Clear Mode (D0FIFO and D1FIFO Port Reading Direction)

If 1 is set in the DCLRM bit in DnFIFOSEL, the USB module automatically clears the buffer memory of the selected pipe when reading of data from the buffer memory has been completed.

Table 28.24 shows the packet reception and buffer memory clearing processing by software for each of the various settings. As shown in Table 28.24, the buffer clearing conditions depend on the value set in the BFRE bit. Using the DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.

The DnFIFO auto clear mode can be set only in the buffer memory reading direction.

Note 1. The DTC can be used in transmission or reception. Although the DMACA is only usable in transmission, limitations apply. For details, refer to section 11.7.1, Usage Notes on Transfer by the DTC and DMACA.

Table 28.24 Packet Reception and Buffer Memory Clearing Processing by Software

Buffer Status When Packet is Received	Register Setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Automatically cleared	Automatically cleared	Automatically cleared	Automatically cleared
Zero-length packet reception	Cleared by software	Cleared by software	Automatically cleared	Automatically cleared
Normal short packet reception	Automatically cleared	Cleared by software	Automatically cleared	Automatically cleared
Transaction count end	Automatically cleared	Cleared by software	Automatically cleared	Automatically cleared

28.3.6 Control Transfers (DCP)

In the data stage of control transfers, data is transferred using the default control pipe (DCP).

The DCP buffer memory is a 64-byte single buffer and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

28.3.6.1 Control Transfers when Host Controller Function 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 SUREQ bit in DCPCTR transmits the specified data for setup transactions. Upon completion of the transaction, the SUREQ bit is cleared 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 DEVSEL bits in DCPMAXP set to 0 and the USBSPD bit in DEVADD0 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 bits and the bits in DEVADDx corresponding to the specified USB address set appropriately. For example, when DEVSEL in PIPEMAXP = 0x2, make appropriate settings in DEVADD2; when DEVSEL = 0x5, 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 (SIGN1 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 SQMON bit in DCPCTR.

(2) Data Stage

Data is transferred using the DCP buffer memory.

The access direction of the DCP buffer memory should be specified using the ISEL bit in CFIFOSEL. The transfer direction should be specified using the DIR bit in DCPCFG.

For the first data packet of the data stage, the data PID should be transferred as DATA1. Set data PID = DATA1 in the SQSET bit and the PID bits = BUF in DCPCFG. 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 SQSET bit in DCPCFG.

For reception of a zero-length packet, the received data length should be confirmed using the DTLN bits in CFIFOCTR after a BRDY interrupt is generated, and the buffer memory should then be cleared using the BCLR bit.

28.3.6.2 Control Transfers when Function Controller Function is Selected

(1) Setup Stage

The USB module always sends an ACK response for a correct setup packet targeted to the USB module. The operation of the USB module in the setup stage is described below.

When receiving a new setup packet, the USB module sets the following bits.

- Set the VALID bit in INTSTS0 to 1.
- Set the PID bits in DCPCTR to NAK.
- Set the CCPL bit in DCPCTR to 0.

When receiving a data packet right after the setup packet, the USB module stores the USB request parameters in USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should always be carried out after setting VALID = 0. In the VALID = 1 state, PID = BUF cannot be set, and the data stage cannot be terminated.

Using the function of the VALID bit, the USB module can suspend the current request processing when receiving a new USB request during a control transfer, and can send a response to the newest request.

In addition, the USB module automatically detects the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the received USB request, distinguishes between control read transfer, control write transfer, and no-data control transfer, and controls stage transitions. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified of occurrence of the error. For the stage control of the USB module, see Figure 28.15.

(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 ISEL bit in CFIFOSEL.

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 CCPL bit to 1 while the PID bits in DCPCTR are set to BUF.

After the above settings have been made, the USB module automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

- For control read transfers
The USB module transmits a zero-length packet and receives an ACK response from the USB host.
- For control write transfers and no-data control transfers
The USB module receives a zero-length packet from the USB host and sends an ACK response.

(4) Control Transfer Auto Response Function

The USB module automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- Any transfer other than a control read transfer: bmRequestType 00h
- Request error : wIndex \neq 00h
- Any transfer other than a no-data control transfer: wLength \neq 00h
- Request error: wValue > 7Fh
- Control transfer of a device state error: DVSQ = 011b (Configured)

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

28.3.7 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 (BFRE bit: see section 28.3.3.1(2), When BRDYM = 0 and BFRE = 1)
- Transaction count function
(TRENb, TRCLR, and TRNCNT bits: see section 28.3.4.5, Transaction Counter (For PIPE1 to PIPE5 in Reading Direction))
- Response PID = NAK function (SHTNAK bit: see section 28.3.4.8, Response PID = NAK Function)
- Auto response mode (ATREPM bit: see section 28.3.4.9, Auto Response Mode)

28.3.8 Interrupt Transfers (PIPE6 to PIPE9)

When the function controller function is selected, the USB module carries out interrupt transfers in accordance with the timing controlled by the host controller.

When the host controller function is selected, the timing of issuing a token can be specified using the interval counter.

28.3.8.1 Interval Counter during Interrupt Transfers when Host Controller Function is Selected

For interrupt transfers, intervals between transactions are set in the IITV bits in PIPEPERI. The USB controller issues interrupt transfer tokens based on the specified intervals.

(1) Counter Initialization

The USB controller initializes the interval counter under the following conditions.

- Power-on reset:
The IITV bits are initialized.
- Buffer memory initialization using the ACLRM bit:
The IITV bits are not initialized but the count value is initialized. Setting the ACLRM bit to 0 starts counting from the value set in the IITV bits.

Note that the interval counter is not initialized in the following case.

- USB bus reset or USB suspended
The IITV 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 module cannot issue tokens even at token issuance timing in the following cases. In such a case, the USB module 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.

28.3.9 Isochronous Transfers (PIPE1 and PIPE2)

The USB module has the following functions for isochronous transfers.

- Notification of isochronous transfer error information
- Interval counter (specified by the IITV bits)
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function (specified by the IFIS bit)

28.3.9.1 Error Detection in Isochronous Transfers

The USB module has a function for detecting the error information described below, so that when errors occur in isochronous transfers, software can control them. Table 28.25 and Table 28.26 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 host controller function 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 function controller function 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 function 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 was received in frames other than the interval frame.

Table 28.25 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 function is selected and the function controller function is selected (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	No interrupts generated in both cases when the host controller function is selected and the function controller function is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the OVRN bit in both cases when host controller function is selected and function controller function is selected. When the function controller function 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 function is selected. It is not generated when the host controller function is selected.

Table 28.26 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 CRCE bit to 1 in both cases when the host controller function is selected and the function controller function is selected.
3	Maximum packet size exceeded errors	A BEMP interrupt is generated to set the PID bits to STALL in both cases when the host controller function is selected and the function controller function is selected.

28.3.9.2 DATA-PID

When the function controller function is selected, the USB module operates as follows in response to the received PID.

IN direction

- DATA0: Sent as data packet PID
- DATA1: Not sent
- DATA2: Not sent
- mData: Not sent

OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets are ignored
- mData: Packets are ignored

28.3.9.3 Interval Counter

The isochronous transfer interval can be set using the IITV bits in PIPEPERI. The interval counter enables the functions shown in Table 28.27 when the function controller function is selected. When the host controller function is selected, the USB module generates the token issuance timing. When the host controller function is selected, the interval counter operation is the same as that in the interrupt transfer.

Table 28.27 Interval Counter Function when the Function Controller Function 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 Function Controller Function is Selected

The USB module initializes the interval counter under the following conditions.

- Power-on Reset
The IITV bits are initialized.
- Buffer memory initialization using the ACLRM bit
The IITV bits are not initialized but the count value is initialized. Setting the ACLRM bit to 0 starts counting from the value set in the IITV bits.

After the interval counter has been initialized, counting is started under either of the following conditions 1 and 2 when a packet has been transferred successfully.

1. An SOF is received after transmission of data in response to an IN token in the PID = BUF state.
2. An SOF is received after reception of data of an OUT token in the PID = BUF state.

Note that the interval counter is not initialized under the following conditions.

When the PID bits are set to NAK or STALL

The interval timer does not stop. The USB module attempts transactions at the subsequent interval.

When the USB bus is reset or USB is suspended

The IITV 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 Host Controller Function is Selected

The USB module controls the interval between token issuance operations based on the IITV bit settings. Specifically, the USB module issues a token for a selected pipe once every 2^{IITV} frames.

The USB module starts counting the token issuance interval at the frame following the frame in which software has set the PID bits to BUF.

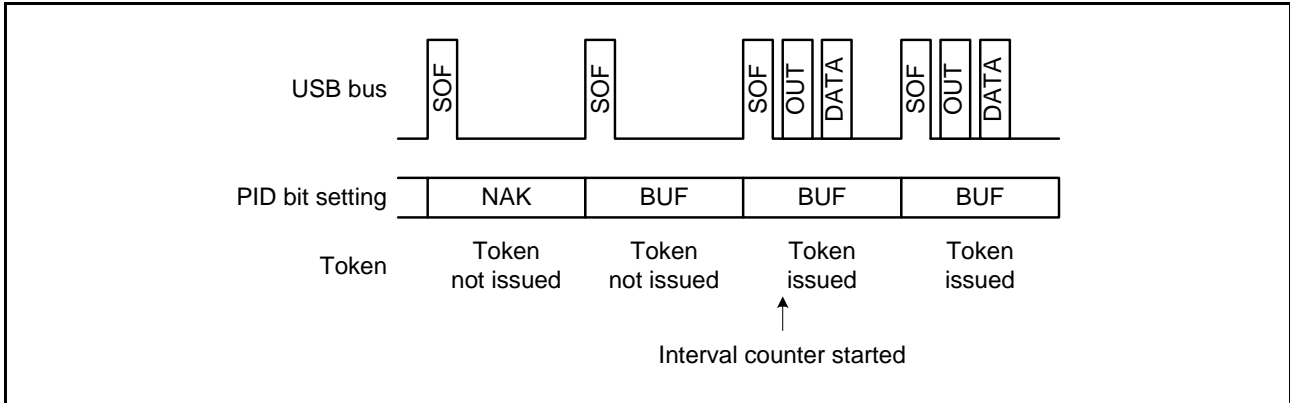


Figure 28.16 Token Issuance when IITV = 0

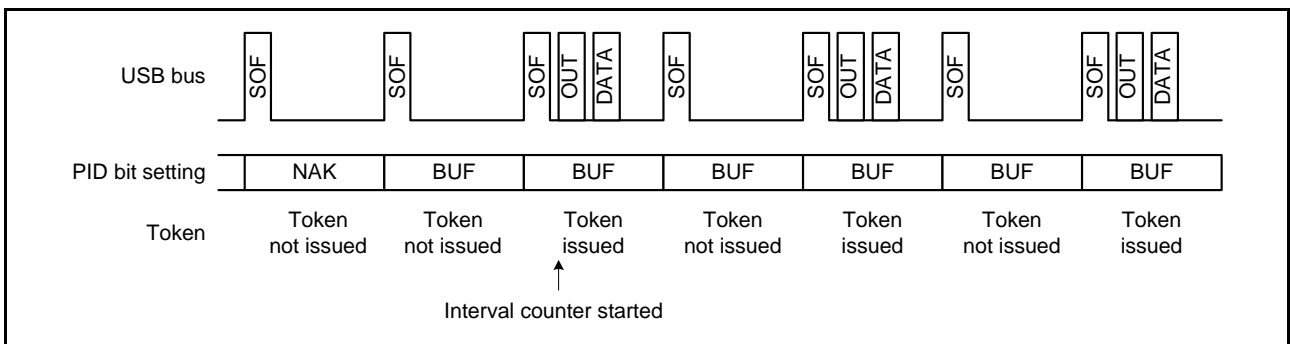


Figure 28.17 Token Issuance when IITV = 1

When the selected pipe is set for isochronous transfers, the USB module carries out the following operation in addition to controlling the token issuance interval. The USB module issues a token even when the NRDY interrupt generation condition is satisfied.

When the selected pipe is for isochronous IN transfers

The USB module generates an NRDY interrupt when the USB module issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).

The USB module sets the OVRN bit to 1 generating an NRDY interrupt when the time to issue an IN token comes while the USB module cannot receive data because the FIFO buffer is full (due to the fact that CPU (DTC or DMACA) is too slow to read data from the FIFO buffer).

When the selected pipe is for isochronous OUT transfers

The USB module 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 CPU (DTC or DMACA) 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 module is reset through a reset pin (The IITV bits are also cleared to 0).
- When software sets the ACLRM bit to 1

(3) Interval Counting and Transfer Control when Function Controller Function is Selected

When the selected pipe is for isochronous OUT transfers

The function controller function generates an NRDY interrupt when the USB module fails to receive a data packet within the interval set by the IITV bits.

The USB module also generates an NRDY interrupt when the USB module fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer being full.

The NRDY interrupt is generated at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the interrupt to be generated at the timing to receive the SOF packet.

However, when the IITV bits are set to a value other than 0, the USB module generates an NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation.

When the PID bits are set to NAK by software after starting the interval timer, the USB module does not generate an NRDY interrupt on receiving an SOF packet.

The timing to start interval counting depends on the IITV bit setting as follows.

- When IITV = 0: The interval counting starts at the frame following the frame in which software has set the PID bits for the selected pipe to BUF.

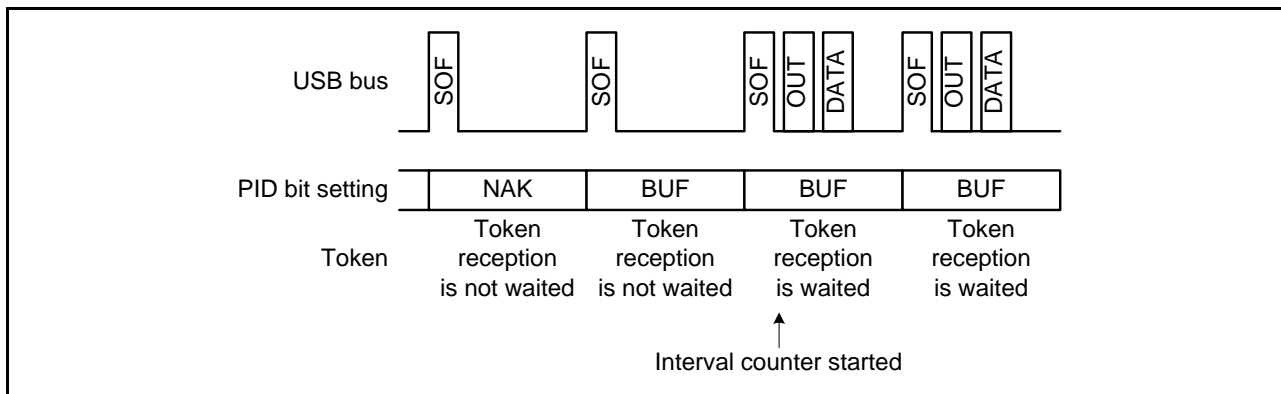


Figure 28.18 Relationship between Frames and Expected Token Reception when IITV = 0

- When IITV ≠ 0: The interval counting starts on completion of successful reception of the first data packet after the PID bits for the selected pipe have been modified to BUF.

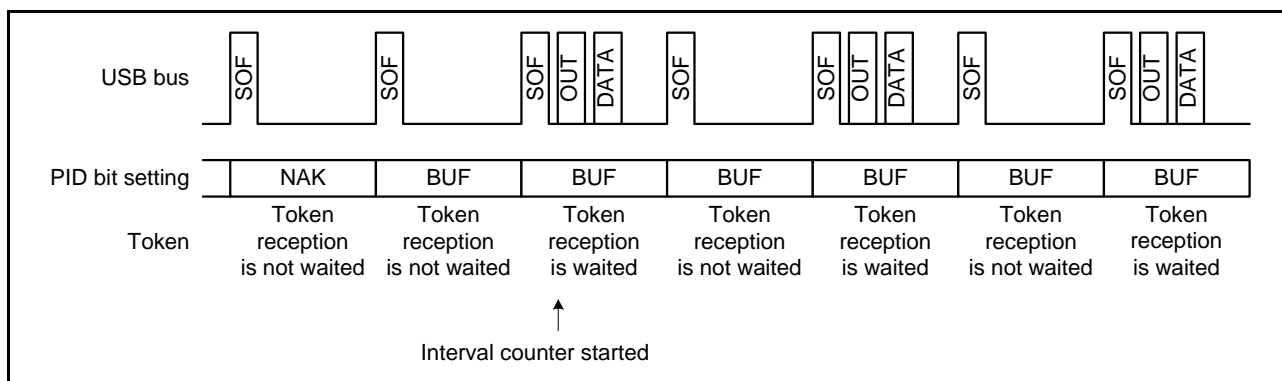


Figure 28.19 Relationship between Frames and Expected Token Reception when IITV ≠ 0

When the selected pipe is for isochronous IN transfers

The IFIS bit should be 1 for this use. When IFIS = 0, the USB module transmits a data packet in response to the received IN token irrespective of the IITV bit setting.

When IFIS = 1, the USB module clears the FIFO buffer when the USB module fails to receive an IN token in the frame at the interval set by the IITV bits while there is data to be transmitted in the FIFO buffer.

The USB module also clears the FIFO buffer when the USB module fails to receive an IN token successfully because of a bus error such as a CRC error contained in the IN token.

The FIFO buffer is cleared at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared at the timing to receive the SOF packet.

The timing to start interval counting depends on the IITV bit setting (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 module (here, the IITV bits are also cleared to 0).
- When software sets the ACLRM bit to 1.
- When the USB module detects a USB bus reset.

(4) Setup of Data to be Transmitted using Isochronous Transfer when Function Controller Function is Selected

With isochronous data transmission using the USB module in function controller function, 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 module transmits data as a normal response. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 28.20 shows an example of transmission using the isochronous transfer transmission data setup function with the USB module when IITV = 0 (every frame) has been set.

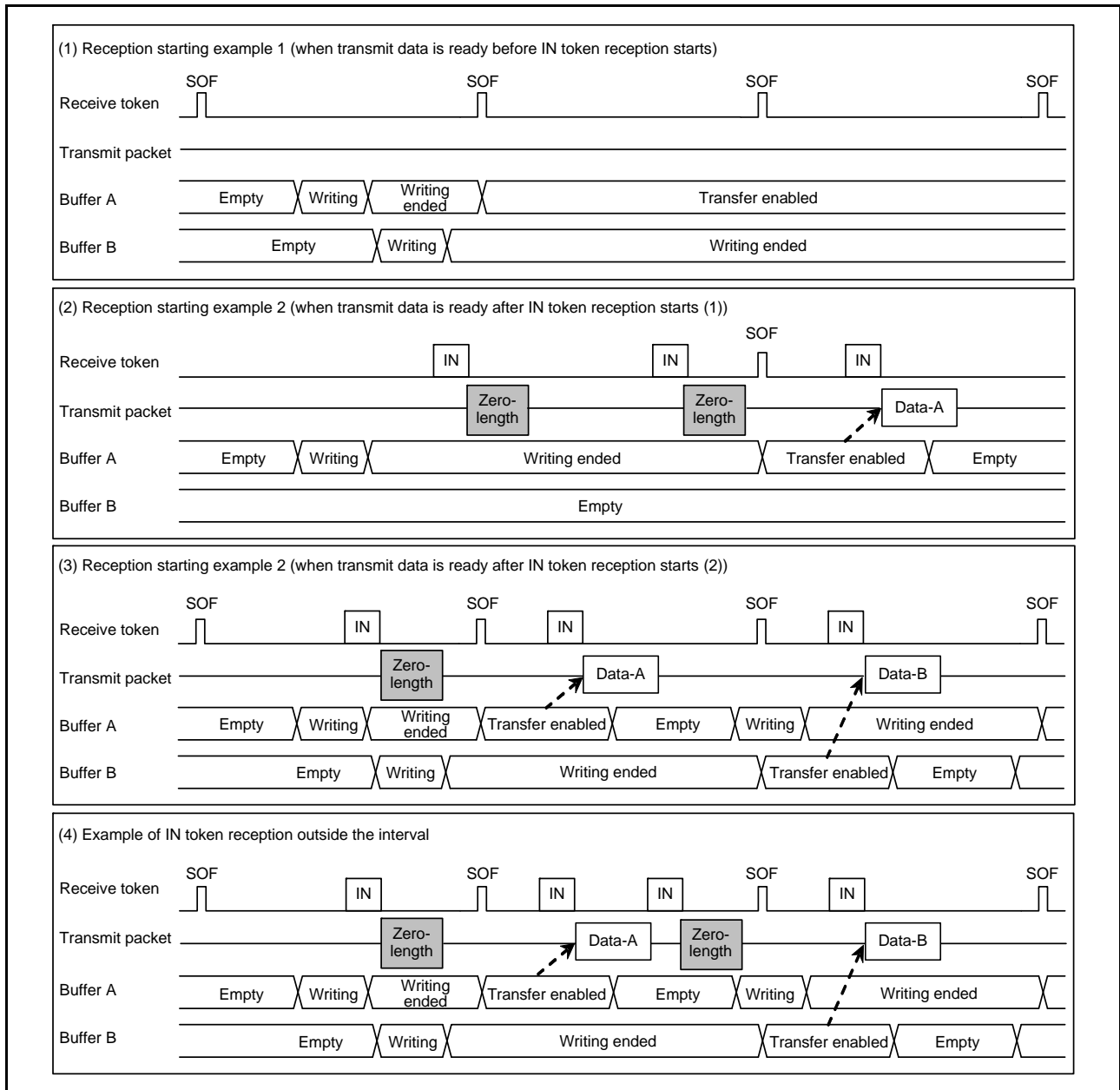


Figure 28.20 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmission Buffer Flush when Function Controller Function 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 module operates as if an IN token had been corrupted, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.

If a double buffer configuration is used and writing to both buffers has been completed, the buffer memory that was cleared is assumed as the data having been sent in the interval frame, and transmission is enabled for the buffer memory that is not cleared with SOF packet reception.

The timing of the buffer flush function depends on the IITV bit setting.

- When IITV = 0
The buffer flush operation starts from the next frame after the pipe becomes valid.
- When IITV ≠ 0
The buffer flush operation is carried out after the first successful transaction.

Figure 28.21 shows an example of the buffer flush function in the USB module. When an unanticipated token is received before the interval frame, the USB module sends the write data or a zero-length packet as an underrun error according to the data setup state.

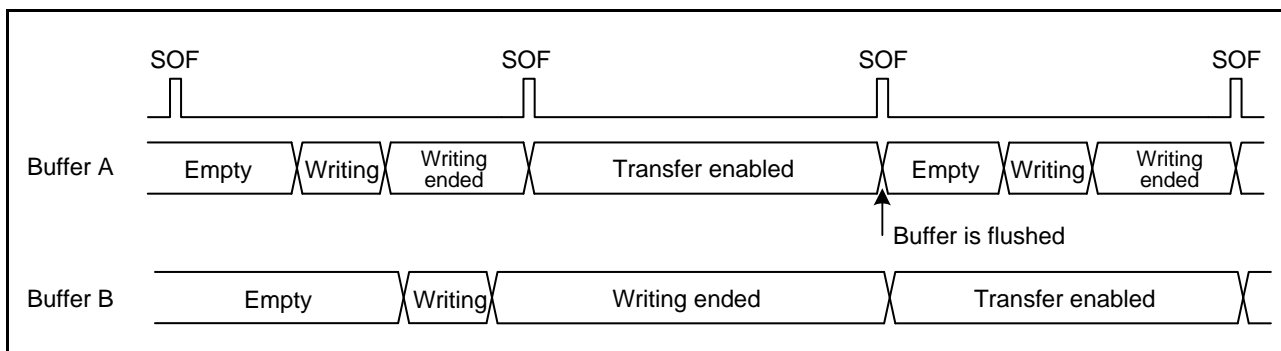


Figure 28.21 Example of Buffer Flush Operation

Figure 28.22 shows an example of interval error occurrence in the USB module. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by (1) in the figure, and the buffer flush function is activated.

If an interval error occurs during an IN transfers, the buffer flush function is activated; if it occurs during an OUT transfer, an NRDY interrupt is generated.

The OVRN bit 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.

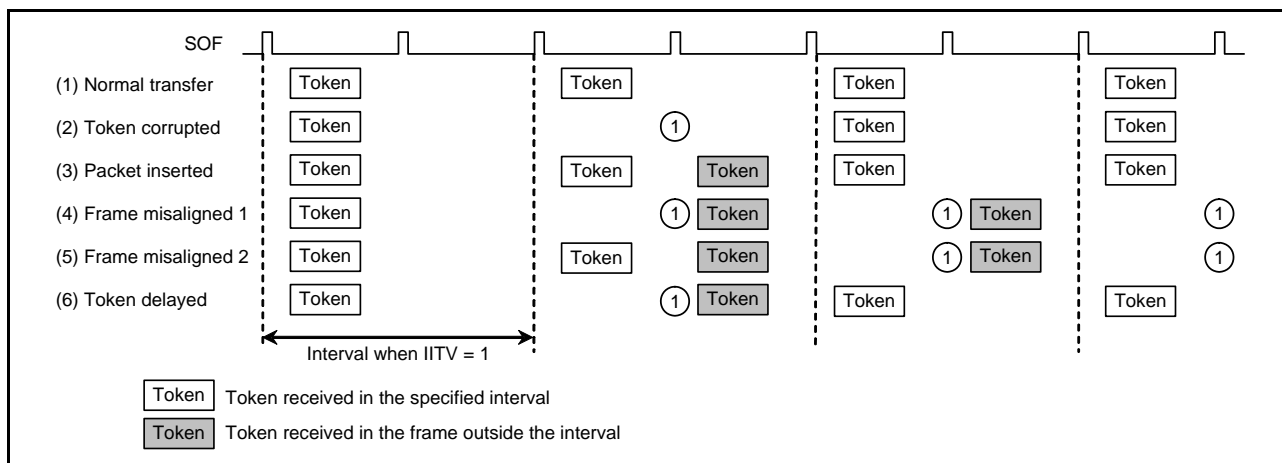


Figure 28.22 Example of Interval Error Occurrence when IITV = 1

28.3.10 SOF Interpolation Function

When the function controller function is selected and if data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB module interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in SYSCFG have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- Power-on reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB module supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing when full-speed operation is being used, the FRNM bit in FRMNUM0 is not updated.

28.3.11 Pipe Schedule

28.3.11.1 Conditions for Generating a Transaction

When the host controller function is selected and UACT has been set to 1, the USB module generates a transaction under the conditions shown in Table 28.28.

Table 28.28 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.

28.3.11.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USB module. After the USB module 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.

28.3.11.3 Enabling USB Communication

Setting the UACT bit of DVSTCTR 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.

29. Serial Communications Interface (SCIa)

The RX62N/RX621 Group has six independent serial communications interface (SCI) units.

The SCI 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).

The SCI also supports the smart card (IC card) interface (SMCI) conforming to ISO/IEC 7816-3 (standards for Identification Cards) as an extended asynchronous communications mode.

29.1 Overview

Table 29.1 lists the specifications of the SCI and Table 29.2 lists the functions of each SCI channel.

Figure 29.1 shows a block diagram of the SCI0 to SCI3, and the SMCI0 to SMCI3, SMCI5, and SMCI6. Figure 29.2 shows a block diagram of the SCI5 and SCI6.

Table 29.1 Specifications of SCI

Item		Specifications
Serial communications mode		<ul style="list-style-type: none"> • Asynchronous • Clock synchronous • Smart card interface
Transfer speed		Bit rate specifiable with on-chip baud rate generator.
Full-duplex communications		Transmitter: Enables continuous transmission by double-buffering. Receiver: Enables continuous reception by double-buffering.
Input/output pins		See Table 29.3.
Data transfer		Selectable from LSB-first or MSB-first transfer
Interrupt sources		Transmit-end, transmit-data-empty, receive-data-full, and receive error
Power consumption reduction function		Module stop state can be set for each unit.
Asynchronous mode	Data length	7 or 8 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even, odd, or none
	Receive error detection	Parity, overrun, and framing errors
	Break detection	Break can be detected by reading RxDn (n = 0 to 3, 5, 6) pin level directly in case of a framing error
	Clock source	Selectable from internal or external clock Enables transfer rate clock input from TMR (SCI5 and SCI6)
	Multi-processor communications function	Serial communication among multiple processors
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun errors
Smart card interface mode	Error processing	An error signal can be automatically transmitted on detection of a parity error during reception
		Data can be automatically re-transmitted on receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.

Table 29.2 Function List of SCI Channels

Item	SCI0 to SCI3	SCI5 and SCI6
Asynchronous mode	Possible	Possible
Clock synchronous mode	Possible	Possible
Smart card interface mode	Possible	Possible
TMR clock input	Not possible	Possible

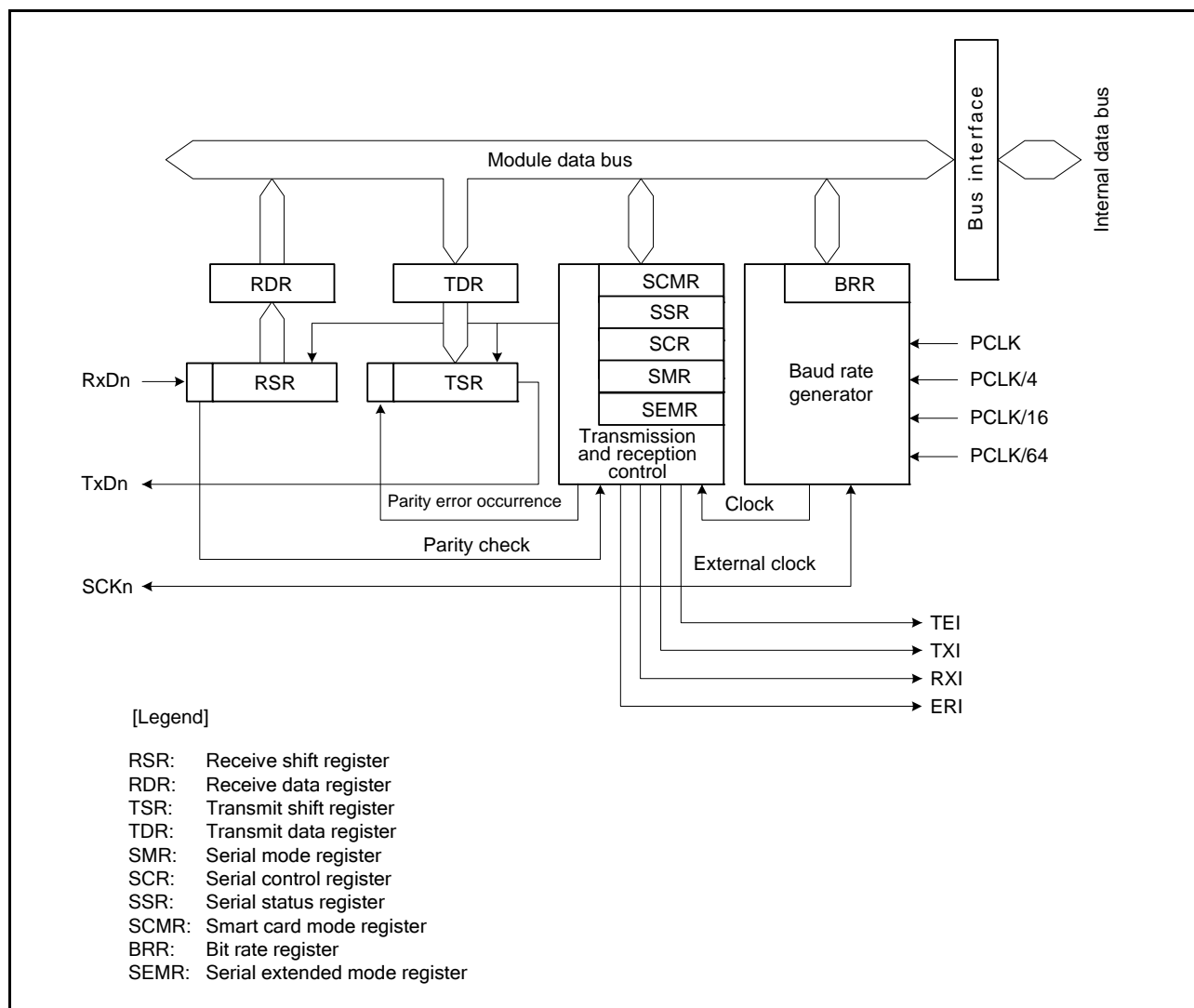


Figure 29.1 Block Diagram of SCI0 to SCI3 and SMCI0 to SMCI3, SMCI5, and SMCI6

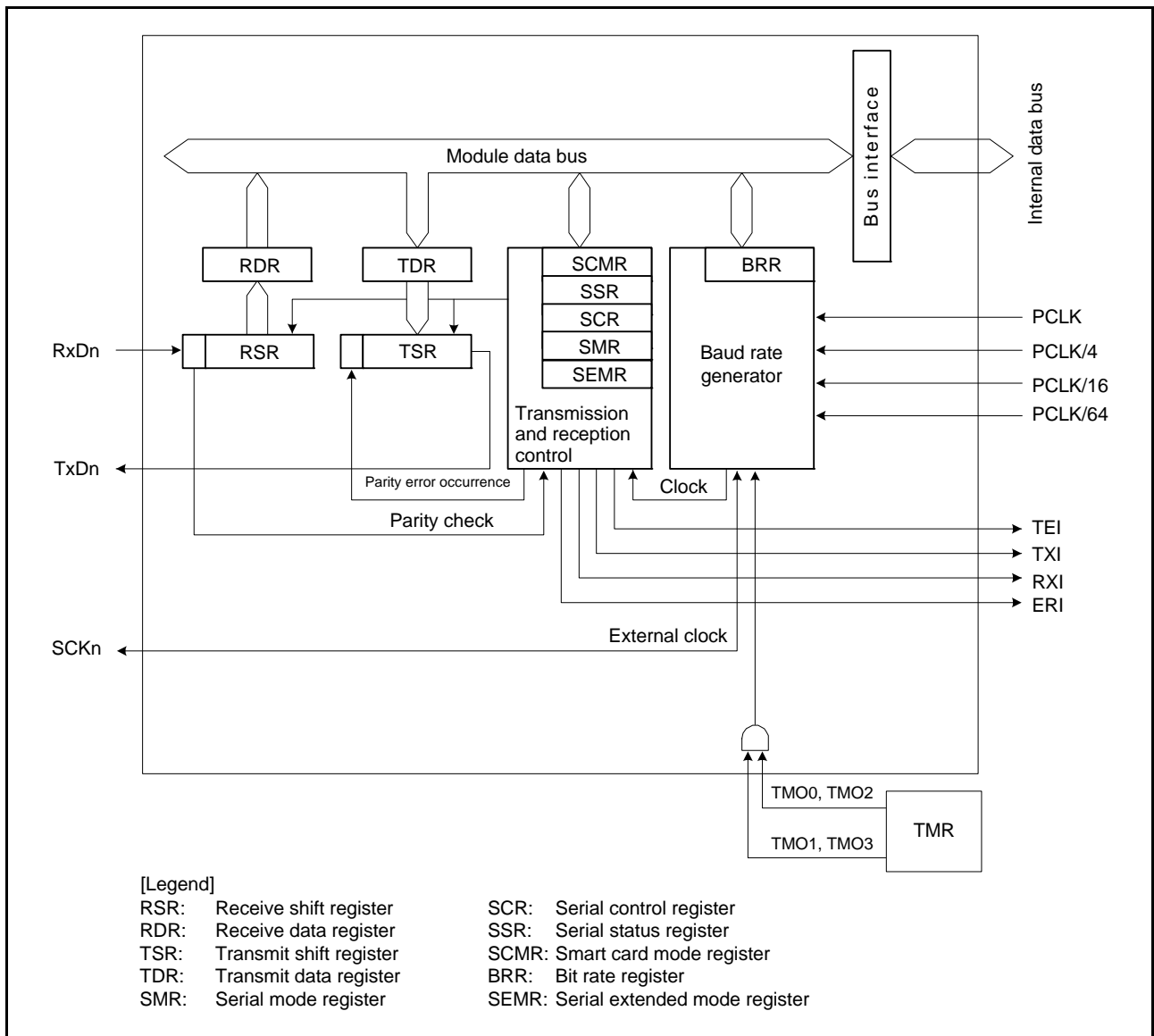


Figure 29.2 Block Diagram of SCI5 and SCI6

Table 29.3 lists the pin configuration of the SCI/SMCI.

Table 29.3 Pin Configuration of SCI/SMCI

Channel	Pin Name	I/O	Function
SCI0/SMCI0	SCK0	I/O	SCI0/SMCI0 clock input/output
	RxD0	Input	SCI0/SMCI0 receive data input
	TxD0	Output	SCI0/SMCI0 transmit data output
SCI1/SMCI1	SCK1	I/O	SCI1/SMCI1 clock input/output
	RxD1	Input	SCI1/SMCI1 receive data input
	TxD1	Output	SCI1/SMCI1 transmit data output
SCI2/SMCI2	SCK2	I/O	SCI2/SMCI2 clock input/output
	RxD2	Input	SCI2/SMCI2 receive data input
	TxD2	Output	SCI2/SMCI2 transmit data output
SCI3/SMCI3	SCK3	I/O	SCI3/SMCI3 clock input/output
	RxD3	Input	SCI3/SMCI3 receive data input
	TxD3	Output	SCI3/SMCI3 transmit data output
SCI5/SMCI5	SCK5	I/O	SCI5/SMCI5 clock input/output
	RxD5	Input	SCI5/SMCI5 receive data input
	TxD5	Output	SCI5/SMCI5 transmit data output
SCI6/SMCI6	SCK6	I/O	SCI6/SMCI6 clock input/output
	RxD6	Input	SCI6/SMCI6 receive data input
	TxD6	Output	SCI6/SMCI6 transmit data output

29.2 Serial Communications Interface Mode

When the SCMR.SMIF bit is 0, the SCI operates in serial communications interface mode.

29.2.1 Register Descriptions

Table 29.4 lists the registers of the SCI.

Table 29.4 Registers of SCI (1 / 2)

Channel	Register Name	Symbol	Value after Reset	Address	Access Size
SCI0	Serial mode register	SMR	00h	0008 8240h	8
	Bit rate register	BRR	FFh	0008 8241h	8
	Serial control register	SCR	00h	0008 8242h	8
	Transmit data register	TDR	FFh	0008 8243h	8
	Serial status register	SSR	84h	0008 8244h	8
	Receive data register	RDR	00h	0008 8245h	8
	Smart card mode register	SCMR	F2h	0008 8246h	8
	Serial extended mode register	SEMR	00h	0008 8247h	8
SCI1	Serial mode register	SMR	00h	0008 8248h	8
	Bit rate register	BRR	FFh	0008 8249h	8
	Serial control register	SCR	00h	0008 824Ah	8
	Transmit data register	TDR	FFh	0008 824Bh	8
	Serial status register	SSR	84h	0008 824Ch	8
	Receive data register	RDR	00h	0008 824Dh	8
	Smart card mode register	SCMR	F2h	0008 824Eh	8
	Serial extended mode register	SEMR	00h	0008 824Fh	8
SCI2	Serial mode register	SMR	00h	0008 8250h	8
	Bit rate register	BRR	FFh	0008 8251h	8
	Serial control register	SCR	00h	0008 8252h	8
	Transmit data register	TDR	FFh	0008 8253h	8
	Serial status register	SSR	84h	0008 8254h	8
	Receive data register	RDR	00h	0008 8255h	8
	Smart card mode register	SCMR	F2h	0008 8256h	8
	Serial extended mode register	SEMR	00h	0008 8257h	8
SCI3	Serial mode register	SMR	00h	0008 8258h	8
	Bit rate register	BRR	FFh	0008 8259h	8
	Serial control register	SCR	00h	0008 825Ah	8
	Transmit data register	TDR	FFh	0008 825Bh	8
	Serial status register	SSR	84h	0008 825Ch	8
	Receive data register	RDR	00h	0008 825Dh	8
	Smart card mode register	SCMR	F2h	0008 825Eh	8
	Serial extended mode register	SEMR	00h	0008 825Fh	8

Table 29.4 Registers of SCI (2 / 2)

Channel	Register Name	Symbol	Value after Reset	Address	Access Size
SCI5	Serial mode register	SMR	00h	0008 8268h	8
	Bit rate register	BRR	FFh	0008 8269h	8
	Serial control register	SCR	00h	0008 826Ah	8
	Transmit data register	TDR	FFh	0008 826Bh	8
	Serial status register	SSR	84h	0008 826Ch	8
	Receive data register	RDR	00h	0008 826Dh	8
	Smart card mode register	SCMR	F2h	0008 826Eh	8
	Serial extended mode register	SEMR	00h	0008 826Fh	8
SCI6	Serial mode register	SMR	00h	0008 8270h	8
	Bit rate register	BRR	FFh	0008 8271h	8
	Serial control register	SCR	00h	0008 8272h	8
	Transmit data register	TDR	FFh	0008 8273h	8
	Serial status register	SSR	84h	0008 8274h	8
	Receive data register	RDR	00h	0008 8275h	8
	Smart card mode register	SCMR	F2h	0008 8276h	8
	Serial extended mode register	SEMR	00h	0008 8277h	8

29.2.1.1 Receive Shift Register (RSR)

RSR is a shift register which is used to receive serial data input from the RxDn pin and converts it into parallel data. When one frame of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

29.2.1.2 Receive Data Register (RDR)

Addresses: SCI0.RDR 0008 8245h, SCI1.RDR 0008 824Dh, SCI2.RDR 0008 8255h, SCI3.RDR 0008 25Dh
 SCI5.RDR 0008 826Dh, SCI6.RDR 0008 8275h



RDR is an 8-bit register that stores receive data. When the SCI has received one frame of serial data, it transfers the received serial data from RSR to RDR where it is stored. This allows RSR to receive the next data. Since RSR and RDR function as a double buffer in this way, continuous receive operations can be performed. Read RDR only once after a receive data full interrupt (RXI) has occurred. Note that if next one frame of data is received before reading receive data from RDR, an overrun error occurs. RDR cannot be written to by the CPU.

29.2.1.3 Transmit Data Register (TDR)

Addresses: SCI0.TDR 0008 8243h, SCI1.TDR 0008 824Bh, SCI2.TDR 0008 8253h, SCI3.TDR 0008 25Bh
 SCI5.TDR 0008 826Bh, SCI6.TDR 0008 8273h



TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR when one frame of data is transmitted, the SCI transfers the written data to TSR to continue transmission. The CPU is able to read from or write to TDR at any time. Only write data for transmission to TDR once after each instance of the transmit data empty interrupt (TXI).

29.2.1.4 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.

29.2.1.5 Serial Mode Register (SMR)

Addresses: SCI0.SMR 0008 8240h, SCI1.SMR 0008 8248h, SCI2.SMR 0008 8250h, SCI3.SMR 0008 8258h
SCI5.SMR 0008 8268h, SCI6.SMR 0008 8270h

	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	Function	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 00: PCLK clock (n = 0)*1 01: PCLK/4 clock (n = 1)*1 10: PCLK/16 clock (n = 2)*1 11: PCLK/64 clock (n = 3)*1	R/W*4
b2	MP	Multi-Processor Mode	(Valid only in asynchronous mode) 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W*4
b3	STOP	Stop Bit Length	(Valid only in asynchronous mode) 0: 1 stop bit 1: 2 stop bits	R/W*4
b4	PM	Parity Mode	(Valid only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity	R/W*4
b5	PE	Parity Enable	(Valid only in asynchronous mode) - 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	R/W*4

Note 1. n is the decimal notation of the value of n in BRR (see section 29.2.1.9, Bit Rate Register (BRR)).

Note 2. In clock synchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.

Note 3. LSB-first is fixed and the MSB (bit 7) in TDR is not transmitted in transmission.

Note 4. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

SMR is used to set the SCI's serial transfer format and select the clock source for the on-chip baud rate generator.

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relation between the settings of these bits and the baud rate, see section 29.3.1.4, Bit Rate Register (BRR)).

MP Bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

STOP Bit (Stop Bit Length)

Selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

PM Bit (Parity Mode)

Selects the parity mode (even or odd) for transmission and reception.

The setting of the PM bit is invalid in multi-processor mode.

PE Bit (Parity Enable)

When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.

Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

CHR Bit (Character Length)

Selects the data length for transmission and reception.

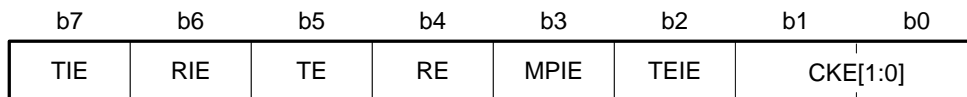
In clock synchronous mode, a fixed data length of 8 bits is used.

CM Bit (Communications Mode)

Selects asynchronous or clock synchronous mode.

29.2.1.6 Serial Control Register (SCR)

Addresses: SCI0.SCR 0008 8242h, SCI1.SCR 0008 824Ah, SCI2.SCR 0008 8252h, SCI3.SCR 0008 825Ah
 SCI5.SCR 0008 826Ah, SCI6.SCR 0008 8272h



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

Bit	Symbol	Bit Name	Function	R/W
b1, b0	CKE[1:0]	Clock Enable	<ul style="list-style-type: none"> • For SCI0 to SCI3 Asynchronous mode b1 b0 0 0: On-chip baud rate generator The SCKn pin functions as I/O port. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 0: External clock The clock with a frequency 16 times the bit rate should be input from the SCKn pin. If the SEMR.ABCS bit is 1, input a clock signal at eight times the frequency. 1 1: External clock The clock with a frequency 16 times the bit rate should be input from the SCKn pin. If the SEMR.ABCS bit is 1, input a clock signal at eight times the frequency. Clock synchronous mode b1 b0 0 0: Internal clock The SCKn pin functions as the clock output pin. 0 1: Internal clock The SCKn pin functions as the clock output pin. 1 0: External clock The SCKn pin functions as the clock input pin. 1 1: External clock The SCKn pin functions as the clock input pin.	R/W*1

Bit	Symbol	Bit Name	Function	R/W
b1, b0	CKE[1:0]	Clock Enable	<ul style="list-style-type: none"> For SCI5 and SCI6 Asynchronous mode b1 b0 0 0: On-chip baud rate generator The SCKn pin functions as I/O port. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 0: External clock or TMR clock <ul style="list-style-type: none"> When an external clock is used, the clock with a frequency 16 times the bit rate should be input from the SCKn pin. If the SEMR.ABCS bit is 1, input a clock signal at eight times the frequency. The TMR clock can be used. 1 1: External clock or TMR clock <ul style="list-style-type: none"> When an external clock is used, the clock with a frequency 16 times the bit rate should be input from the SCKn pin. If the SEMR.ABCS bit is 1, input a clock signal at eight times the frequency. The TMR clock can be used. Clock synchronous mode b1 b0 0 0: Internal clock The SCKn pin functions as the clock output pin. 0 1: Internal clock The SCKn pin functions as the clock output pin. 1 0: External clock The SCKn pin functions as the clock input pin. 1 1: External clock The SCKn pin functions as the clock input pin.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	0: A TEI interrupt request is disabled 1: A TEI interrupt request is enabled	R/W
b3	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode when SMR.MP = 1) 0: Normal reception 1: When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags ORER and FER in SSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. A 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.

SCR is a register that enables or disables the SCI transfer operations and selects the transfer clock source.

CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.

TEIE Bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.

A TEI interrupt request is disabled by clearing the TEIE bit to 0.

MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags ORER and FER in SSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, see section 29.2.3, Multi-Processor Communications Function.

When the receive data includes the MPB bit is SSR set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.

When the receive data includes the MPB bit set to 1, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the RIE bit in SCR is set to 1), and setting the flags ORER and FER to 1 is enabled.

MPIE should be set to 0 if multi-processor communications function is not to be used.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Note that SMR should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by clearing the RE bit to 0, the ORER, FER, and PER flags in SSR are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to TDR. Note that SMR should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by clearing the RIE bit to 0.

An ERI interrupt request can be cancelled by reading 1 from the ORER, FER, or PER flag in SSR and then clearing the flag to 0, or clearing the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables notification of a TXI interrupt request.

Notification of a TXI interrupt request is disabled by clearing the TIE bit to 0.

29.2.1.7 Serial Status Register (SSR)

Addresses: SCI0.SSR 0008 8244h, SCI1.SSR 0008 824Ch, SCI2.SSR 0008 8254h, SCI3.SSR 000 825Ch
SCI5.SSR 0008 826Ch, SCI6.SSR 0008 8274h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

[Legend] x: Undefined

Bit	Symbol	Bit Name	Function	R/W
b0	MPBT	Multi-Processor Bit Transfer	Sets the multi-processor bit for adding to the transmission frame	R/W
b1	MPB	Multi-Processor	Value of the multi-processor bit in the reception frame	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W)*1
b4	FER	Framing Error Flag	0: No framing error occurred 1: A framing error has occurred	R/(W)*1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W)*1
b6	RDRF	Receive Data Full Flag	0: When data is transferred from RDR 1: When data has been received normally, and transferred from RSR to RDR	R/(W)*2
b7	TDRE	Transmit Data Empty Flag	0: When data is transferred to TDR 1: When data is transferred from TDR to TSR	R/(W)*2

Note 1. Only 0 can be written to this bit, to clear the flag.

Note 2. Write 1 when writing is necessary.

SSR is a register containing status flags of the SCI and the transmission/reception multi-processor bit.

MPBT Bit (Multi-Processor Bit Transfer)

Sets the value of the multi-processor bit for adding to the transmission frame.

MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the RE bit in SCR is 0.

TEND Flag (Transmission End Flag)

Indicates completion of transmission.

[Setting conditions]

- Clearing of the SCR.TE bit to 0 (disabling serial transmission operations)
- The TDR is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- Writing of further data for transmission to the TDR

When the TEND flag is cleared by writing the data for transmission to the TDR, read the TEND flag and check that it has actually been cleared to 0.

PER Bit (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent serial reception cannot be transferred to RDR.

[Clearing condition]

- When a 0 is written to PER after reading PER = 1 (after writing a 0 to it, read the PER bit to check that it has actually been cleared to 0.)
Even when the RE bit in SCR is cleared to 0 (which indicates that serial reception is disabled), the PER flag is not affected and retains its previous value.

FER Bit (Framing Error Flag)

Indicates that a framing error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When the stop bit is 0
In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. Note that although receive data when the framing error occurs is transferred to RDR, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent serial reception cannot be transferred to RDR.

[Clearing condition]

- When a 0 is written to FER after reading FER = 1 (After writing a 0 to it, read the FER bit to check that it has actually been cleared to 0.)
Even when the RE bit in SCR is cleared to 0, the FER flag is not affected and retains its previous value.

ORER Bit (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR
In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous mode, serial transmission also cannot continue.

[Clearing condition]

- When a 0 is written to ORER after reading ORER = 1 (After writing a 0 to it, read the ORER bit to check that it has actually been cleared to 0.)
Even when the RE bit in SCR is cleared to 0, the ORER flag is not affected and retains its previous value.

RDRF Bit (Receive Data Full Flag)

Indicates whether RDR has received data.

[Setting condition]

- When data has been received normally, and transferred from RSR to RDR

[Clearing condition]

- When data is transferred from RDR

TDRE Bit (Transmit Data Empty Flag)

Indicates whether TDR has data to be transmitted.

[Setting condition]

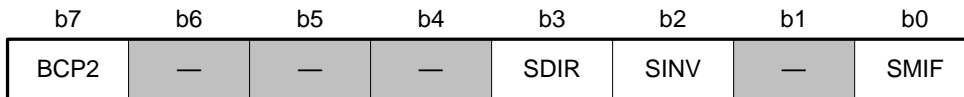
- When data is transferred from TDR to TSR

[Clearing condition]

- When data is transferred to TDR

29.2.1.8 Smart Card Mode Register (SCMR)

Addresses: SCI0.SCMR 0008 8246h, SCI1.SCMR 0008 824Eh, SCI2.SCMR 0008 8256h, SCI3.SCMR 0008 25Eh, SCI5.SCMR 0008 826Eh, SCI6.SCMR 0008 8276h



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

Bit	Symbol	Bit Name	Function	R/W																																				
b0	SMIF	Smart Card Interface Mode Select	0: Serial communications interface mode 1: Smart card interface mode	R/W*1																																				
b1	—	Reserved	This bit is always read as 1. The write value should always be 1.	R/W																																				
b2	SINV	Smart Card 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	Smart Card Data Transfer Direction	0: Transfer with LSB-first 1: Transfer with MSB-first	R/W*1																																				
b6 to b4	—	Reserved	This bit is always read as 1. The write value should always be 1.	R/W																																				
b7	BCP2	Base Clock Pulse 2	Selects the number of base clock cycles in combination with the BCP1 and BCP0 bits in SMR. Setting values in BCP2 bit in SCMR and BCP1 and BCP0 bits in SMR <table style="font-size: small; margin-left: 20px;"> <tr> <td>BCP2</td> <td>BCP1</td> <td>BCP0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0: 93 clock cycles (S = 93)*2</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1: 128 clock cycles (S = 128)*2</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0: 186 clock cycles (S = 186)*2</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1: 512 clock cycles (S = 512)*2</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0: 32 clock cycles (S = 32)*2 (Initial Value)</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1: 64 clock cycles (S = 64)*2</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0: 372 clock cycles (S = 372)*2</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1: 256 clock cycles (S = 256)*2</td> </tr> </table>	BCP2	BCP1	BCP0		0	0	0	0: 93 clock cycles (S = 93)*2	0	0	1	1: 128 clock cycles (S = 128)*2	0	1	0	0: 186 clock cycles (S = 186)*2	0	1	1	1: 512 clock cycles (S = 512)*2	1	0	0	0: 32 clock cycles (S = 32)*2 (Initial Value)	1	0	1	1: 64 clock cycles (S = 64)*2	1	1	0	0: 372 clock cycles (S = 372)*2	1	1	1	1: 256 clock cycles (S = 256)*2	R/W*1
BCP2	BCP1	BCP0																																						
0	0	0	0: 93 clock cycles (S = 93)*2																																					
0	0	1	1: 128 clock cycles (S = 128)*2																																					
0	1	0	0: 186 clock cycles (S = 186)*2																																					
0	1	1	1: 512 clock cycles (S = 512)*2																																					
1	0	0	0: 32 clock cycles (S = 32)*2 (Initial Value)																																					
1	0	1	1: 64 clock cycles (S = 64)*2																																					
1	1	0	0: 372 clock cycles (S = 372)*2																																					
1	1	1	1: 256 clock cycles (S = 256)*2																																					

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).
 2. S is the value of S in BRR (see section 29.2.1.9, Bit Rate Register (BRR)).

SCMR selects smart card interface mode and its format.

SMIF Bit (Smart Card Interface Mode Select)

When this bit is set to 1, smart card interface mode is selected.

When this bit is set to 0, asynchronous or clock synchronous mode is selected.

SINV Bit (Smart Card Data Invert)

Inverts the transmit/receive data logic level. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the PM bit in SMR.

SDIR Bit (Smart Card Data Transfer Direction)

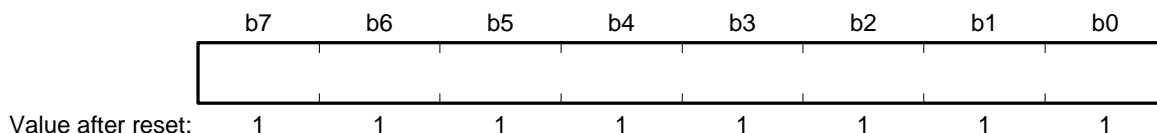
Selects the serial/parallel conversion format.

BCP2 Bit (Base Clock Pulse 2)

Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the BCP1 and BCP0 bits in SMR.

29.2.1.9 Bit Rate Register (BRR)

Addresses: SCI0.BRR 0008 8241h, SCI1.BRR 0008 8249h, SCI2.BRR 0008 8251h,
 SCI3.BRR 0008 8259h, SCI5.BRR 0008 8269h, SCI6.BRR 0008 8271h



BRR is an 8-bit register that adjusts the bit rate.

As the SCI performs baud rate generator control independently for each channel, different bit rates can be set for each channel. Table 29.5 shows the relationships between the N setting in BRR and bit rate B for normal asynchronous mode and clock synchronous mode, and smart card interface mode.

The initial value of BRR is FFh.

BRR can be read from by the CPU at all times, but it can be written to only when the TE and RE bits in SCR are 0.

Table 29.5 Relationships between N Setting in BRR and Bit Rate B

Mode	ABCS Bit in SEMR	BRR Setting	Error
Asynchronous	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	Error (%) = $\left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	Error (%) = $\left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous		$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	

[Legend]

- B: Bit rate (bps)
- N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
- PCLK: Operating frequency (MHz)
- n and S: Determined by the SMR setting shown in the following table.

SMR Setting		
CKS[1:0] Bits	Clock Source	n
0 0	PCLK clock	0
0 1	PCLK/4 clock	1
1 0	PCLK/16 clock	2
1 1	PCLK/64 clock	3

SCMR Setting		SMR Setting	
BSP2 Bit	BCP[1:0] Bits	Base Clock	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 29.6 shows sample N settings in BRR in normal asynchronous mode. Table 29.7 shows the maximum bit rate settable for each operating frequency. Table 29.9 shows sample N settings in BRR in clock synchronous mode. Table 29.8 and Table 29.10 show the maximum bit rates with external clock input.

When the asynchronous mode base clock select bit (ABCS) in the serial extended mode register (SEMR) is set to 1 in asynchronous mode, the bit rate is two times that of shown in Table 29.6.

Table 29.6 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	8			9.8304			10			12		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	-0.26	2	177	-0.25	2	212	0.03
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	25	0.16	0	31	0.00	0	32	-1.36	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	-2.34
31250	0	7	0.00	0	9	-1.70	0	9	0.00	0	11	0.00
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	-2.34

Bit Rate (bps)	Operating Frequency PCLK (MHz)								
	12.288			14			16		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	-0.17	3	70	0.03
150	2	159	0.00	2	181	0.16	2	207	0.16
300	2	79	0.00	2	90	0.16	2	103	0.16
600	1	159	0.00	1	181	0.16	1	207	0.16
1200	1	79	0.00	1	90	0.16	1	103	0.16
2400	0	159	0.00	0	181	0.16	0	207	0.16
4800	0	79	0.00	0	90	0.16	0	103	0.16
9600	0	39	0.00	0	45	-0.93	0	51	0.16
19200	0	19	0.00	0	22	-0.93	0	25	0.16
31250	0	11	2.40	0	13	0.00	0	15	0.00
38400	0	9	0.00	—	—	—	0	12	0.16

Note: This is an example when the ABCS bit in SEMR is 0.
When the ABCS bit is set to 1, the bit rate is two times.

Table 29.6 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	17.2032			18			19.6608			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	75	0.48	3	79	-0.12	3	86	0.31	3	88	-0.25
150	2	223	0.00	2	233	0.16	2	255	0.00	3	64	0.16
300	2	111	0.00	2	116	0.16	2	127	0.00	2	129	0.16
600	1	223	0.00	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	111	0.00	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	223	0.00	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	111	0.00	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	55	0.00	0	58	-0.69	0	63	0.00	0	64	0.16
19200	0	27	0.00	0	28	1.02	0	31	0.00	0	32	-1.36
31250	0	16	1.20	0	17	0.00	0	19	-1.70	0	19	0.00
38400	0	13	0.00	0	14	-2.34	0	15	0.00	0	15	1.73

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	25			30			33			50		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	110	-0.02	3	132	0.13	3	145	0.33	3	221	-0.02
150	3	80	0.47	3	97	-0.35	3	106	0.39	3	162	-0.15
300	2	162	-0.15	2	194	0.16	2	214	-0.07	3	80	0.47
600	2	80	0.47	2	97	-0.35	2	106	0.39	2	162	-0.15
1200	1	162	-0.15	1	194	0.16	1	214	-0.07	2	80	0.47
2400	1	80	0.47	1	97	-0.35	1	106	0.39	1	162	-0.15
4800	0	162	-0.15	0	194	0.16	0	214	-0.07	1	80	0.47
9600	0	80	0.47	0	97	-0.35	0	106	0.39	1	40	-0.77
19200	0	40	-0.76	0	48	-0.35	0	53	-0.54	0	80	0.47
31250	0	24	0.00	0	29	0	0	32	0	0	49	0.00
38400	0	19	1.73	0	23	1.73	0	26	-0.54	0	40	-0.77

Note: This is an example when the ABCS bit in SEMR is 0.
When the ABCS bit is set to 1, the bit rate is two times.

Table 29.7 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N	PCLK (MHz)	Maximum Bit Rate (bps)	n	N
8	250000	0	0	17.2032	537600	0	0
9.8304	307200	0	0	18	562500	0	0
10	312500	0	0	19.6608	614400	0	0
12	375000	0	0	20	625000	0	0
12.288	384000	0	0	25	781250	0	0
14	437500	0	0	30	937500	0	0
16	500000	0	0	33	1031250	0	0
				50	1562500	0	0

Note: When the ABCS bit in SEMR is set to 1, the bit rate is two times.

Table 29.8 Maximum Bit Rate with External Clock Input (Asynchronous Mode) (1)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bps)	PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bps)
8	2.0000	125000	17.2032	4.3008	268800
9.8304	2.4576	153600	18	4.5000	281250
10	2.5000	156250	19.6608	4.9152	307200
12	3.0000	187500	20	5.0000	312500
12.288	3.0720	192000	25	6.2500	390625
14	3.5000	218750	30	7.5000	468750
16	4.0000	250000	33	8.2500	515625
			50	12.500	781250

Note: This is an example when the ABCS bit in SEMR is 0.

Table 29.8 Maximum Bit Rate with External Clock Input (Asynchronous Mode) (2)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bps)	PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bps)
8	2.0000	250000	17.2032	4.3008	537600
9.8304	2.4576	307200	18	4.5000	562500
10	2.5000	312500	19.6608	4.9152	614400
12	3.0000	375000	20	5.0000	625000
12.288	3.0720	384000	25	6.2500	781250
14	3.5000	437500	30	7.5000	937500
16	4.0000	500000	33	8.2500	1031250
			50	12.500	1562500

Note: This is an example when the ABCS bit in SEMR is 1.

Table 29.9 BRR Settings for Various Bit Rates (Clock Synchronous Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)															
	8		10		16		20		25		30		33		50	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110																
250	3	124	—	—	3	249										
500	2	249	—	—	3	124	—	—			3	233				
1k	2	124	—	—	2	249	—	—	3	97	3	116	3	128	3	194
2.5k	1	199	1	249	2	99	2	124	2	155	2	187	2	205	3	77
5k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2	155
10k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	2	77
25k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1	124
50k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	1	61
100k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0	124
250k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0	49
500k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	0	24
1M	0	1	—	—	0	3	0	4	—	—	—	—	—	—	—	—
2M	0	0*	—	—	0	1	—	—	—	—	—	—	—	—	—	—
2.5M			0	0*	—	—	0	1	-	-	0	2	-	-	0	4
4M					0	0	—	—	—	—	—	—	—	—	—	—
5M							0	0*	—	—	—	—	—	—	—	—
6.25M									0	0*	—	—	—	—	0	1
7.5M											0	0*	—	—	—	—
8.25M													0	0*	—	—
12.5M															0	0*

[Legend]

Space: Setting prohibited.

—: Can be set, but there will be error.

Note: * Continuous transmission or reception is not possible.

Table 29.10 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bps)	PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bps)
8	1.3333	1333333.3	20	3.3333	3333333.3
10	1.6667	1666666.7	25	4.1667	4166666.7
12	2.0000	2000000.0	30	5.0000	5000000.0
14	2.3333	2333333.3	33	5.5000	5500000.0
16	2.6667	2666666.7	50	8.3333	8333333.3
18	3.0000	3000000.0			

29.2.1.10 Serial Extended Mode Register (SEMR)

Addresses: SCI0.SEMR 0008 8247h, SCI1.SEMR 0008 824Fh, SCI2.SEMR 0008 8257h, SCI3.SEMR 0008 25Fh
 SCI5.SEMR 0008 826Fh, SCI6.SEMR 0008 8277h



Bit	Symbol	Bit Name	Function	R/W									
b0	ACS0	Asynchronous Mode Clock Source Select	(Valid only in asynchronous mode) 0: External clock input (SCI0 to SCI3, SCI5, and SCI6) 1: TMR clock input (valid only for SCI5 and SCI6) The following table shows the correspondence between SCI channels and compare match outputs. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>SCI</th> <th>TMR</th> <th>Compare Match Output</th> </tr> </thead> <tbody> <tr> <td>SCI5</td> <td>Unit 0</td> <td>TMO0, TMO1</td> </tr> <tr> <td>SCI6</td> <td>Unit 1</td> <td>TMO2, TMO3</td> </tr> </tbody> </table>	SCI	TMR	Compare Match Output	SCI5	Unit 0	TMO0, TMO1	SCI6	Unit 1	TMO2, TMO3	R/W*
SCI	TMR	Compare Match Output											
SCI5	Unit 0	TMO0, TMO1											
SCI6	Unit 1	TMO2, TMO3											
b3 to b1	—	Reserved	This bit is always read as 0. The write value should always 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*									
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W									

Note : Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

SEMR selects the clock source for 1-bit period in asynchronous mode.

For SCI5 and SCI6, the TMO_n output (n = 0 to 3) of TMR units 0 and 1 can be set as the serial transfer base clock.

Figure 29.3 shows a setting example when the TMO_n output of TMR_n (n = 0 to 3) is selected.

ACS0 Bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the asynchronous mode.

The ACS0 bit is valid in asynchronous mode (CM bit in SMR = 0) and when an external clock input is selected (CKE[1:0] bits in SCR = 10b or 11b). An external clock input or internal TMR clock input can be selected.

These bits for the other SCI channels than SCI5 and SCI6 are reserved. The write values to these bits for other than SCI5 and SCI6 should always be 0.

ABCS Bit (Asynchronous Mode Base Clock Select)

Selects the number of base clock pulses for 1-bit period.

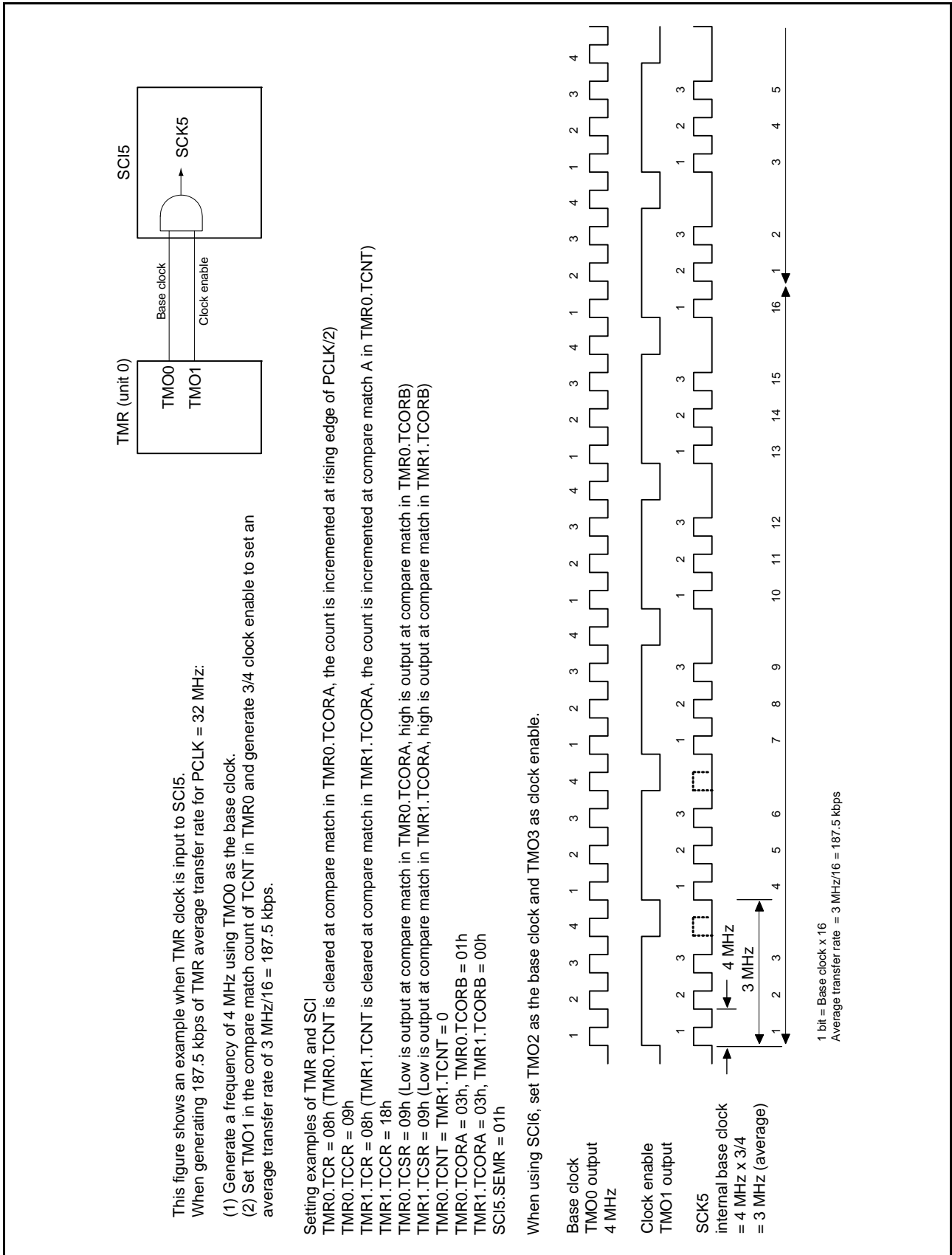


Figure 29.3 Example of Average Transfer Rate Setting when TMR Clock is Input

29.2.2 Operation in Asynchronous Mode

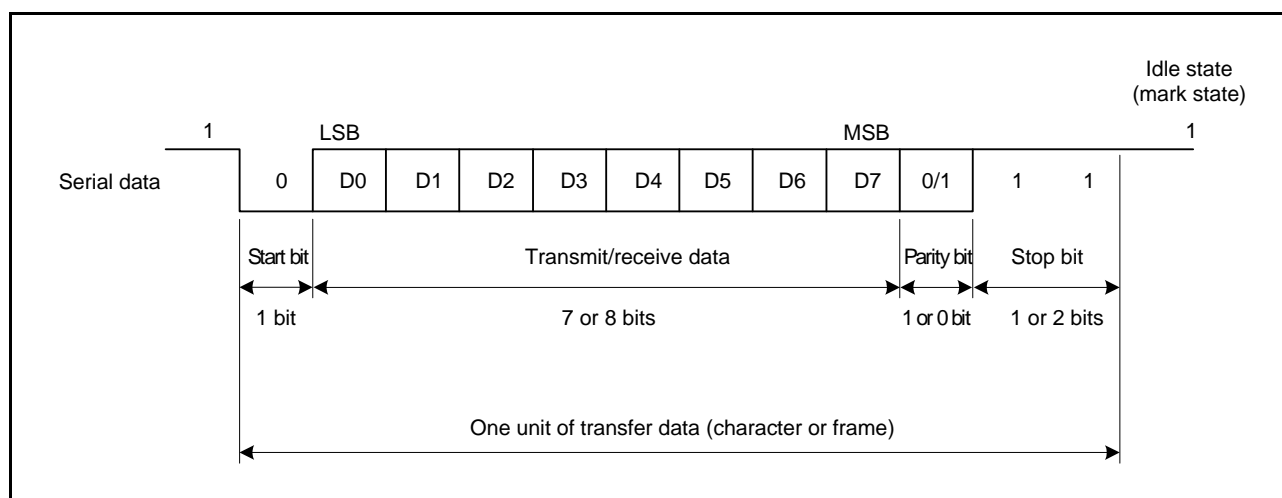
Figure 29.4 shows the general format for asynchronous serial communications.

One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).

In asynchronous serial communications, the communications line is usually held in the mark state (high level).

The SCI monitors the communications line, and when it goes to the space state (low level), recognizes a start bit and starts serial communications.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.



**Figure 29.4 Data Format in Asynchronous Serial Communications
(Example with 8-Bit Data, Parity, Two Stop Bits)**

29.2.2.1 Serial Data Transfer Format

Table 29.11 shows the serial data transfer formats that can be used in asynchronous mode.

Any of 12 transfer formats can be selected according to the SMR setting. For details of multi-processor function, see section 29.2.3, Multi-Processor Communications Function.

Table 29.11 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		

[Legend] S:Start bit
 STOP: Stop bit
 P: Parity bit
 MPB: Multi-processor bit

29.2.2.2 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* 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* of the base clock, data is latched at the middle of each bit, as shown in Figure 29.5. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1+F) \right| \times 100 [\%] \quad \cdots \text{Formula (1)}$$

[Legend]

M: Reception margin

N: Ratio of bit rate to clock (N = 16 when ABCS in SEMR = 0, N = 8 when ABCS in SEMR = 1)

D: Duty cycle of clock (D = 0.5 to 1.0)

L: Frame length (L = 9 to 12)

F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100 (\%) = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note: * 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.

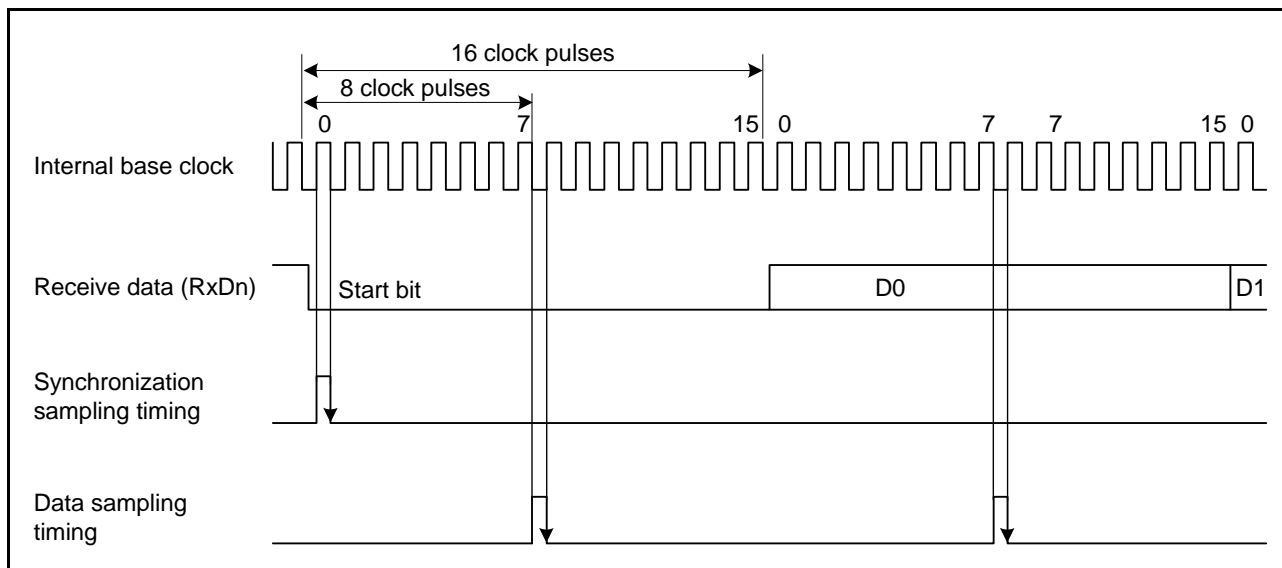


Figure 29.5 Receive Data Sampling Timing in Asynchronous Mode

29.2.2.3 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 CA bit in SMR and the CKE[1:0] bits in SCR.

When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when ABCS in SEMR = 0) and 8 times the bit rate (when ABCS in SEMR = 1). In addition, when an external clock is specified, the base clock of TMR0 and TMR1 can be selected by the ACS0 bit in SEMR of SCIm (m = 5, 6).

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 29.6.

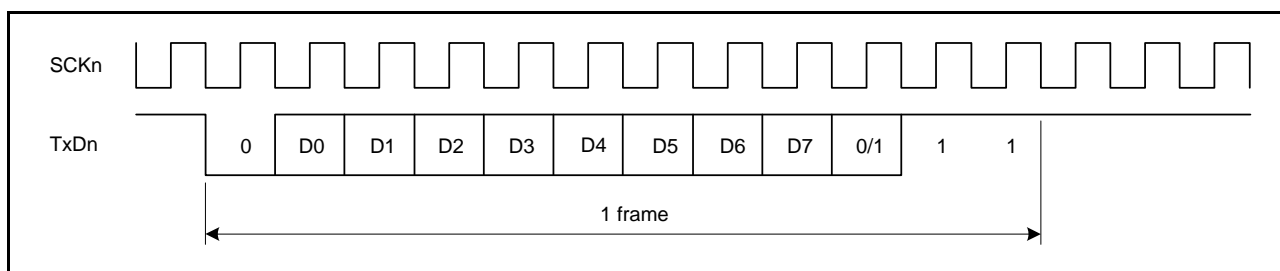


Figure 29.6 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode)

29.2.2.4 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value "00h" to the SCR and then continue through the procedure for SCI given in the sample flowchart (Figure 29.7). Whenever the operating mode or transfer format is changed, the SCR must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.

Note that clearing the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR nor RDR.

Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.

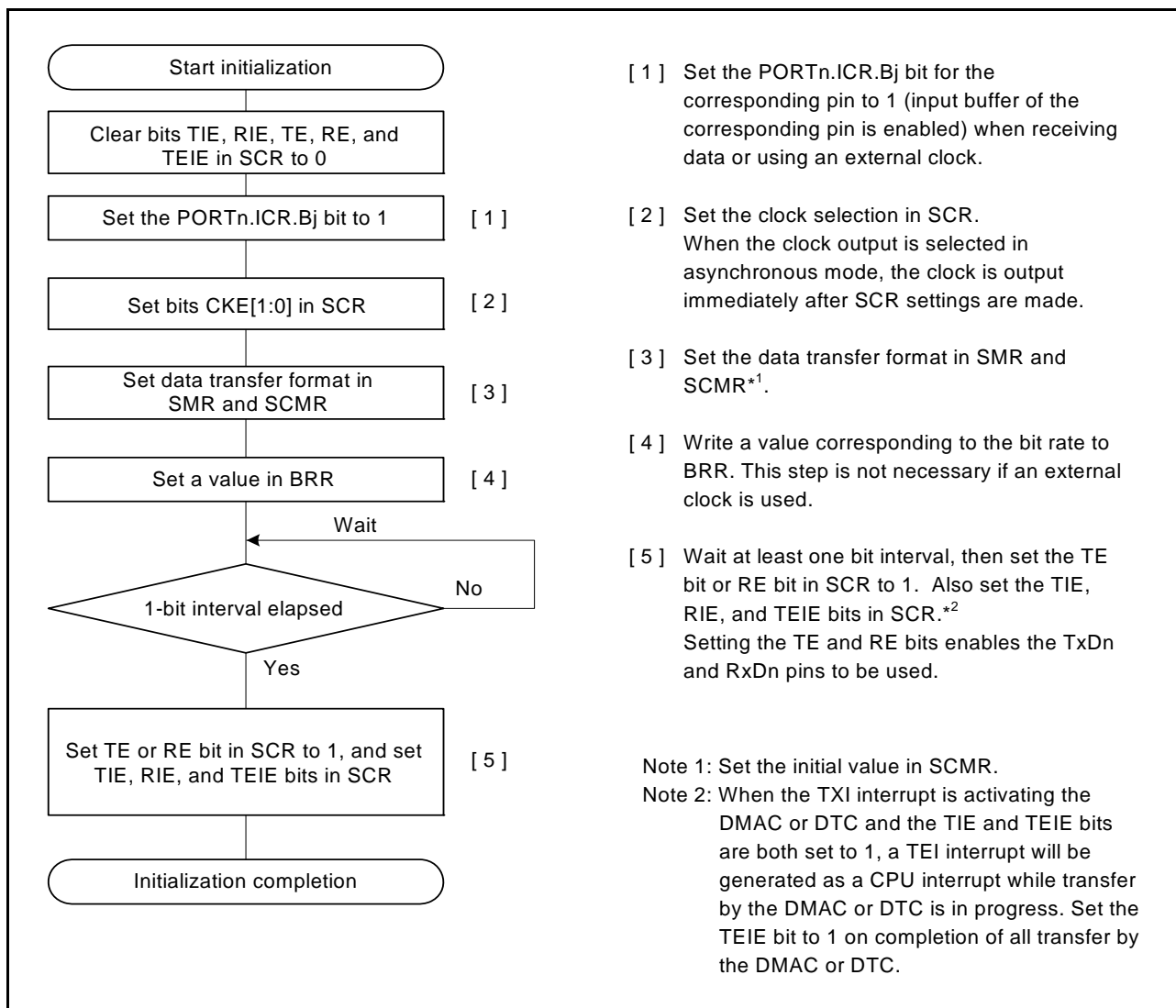


Figure 29.7 Sample SCI Initialization Flowchart (Asynchronous Mode)

29.2.2.5 Serial Data Transmission (Asynchronous Mode)

Figure 29.8 shows an example of the operation for serial transmission in asynchronous mode.

In serial transmission, the SCI operates as described below.

1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt processing routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. After transferring data from TDR to TSR, the SCI starts transmission. If the TIE bit in SCR is 1 at this time, a TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in the TXI interrupt processing routine before transmission of the current transmit data is completed.
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, the next transmit data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.
6. If TDR is not updated, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the TEIE flag in SCR is 1 at this time, the TEND flag in SSR is set to 1 and a TEI interrupt request is generated.

Figure 29.9 shows a sample flowchart for serial transmission in asynchronous mode.

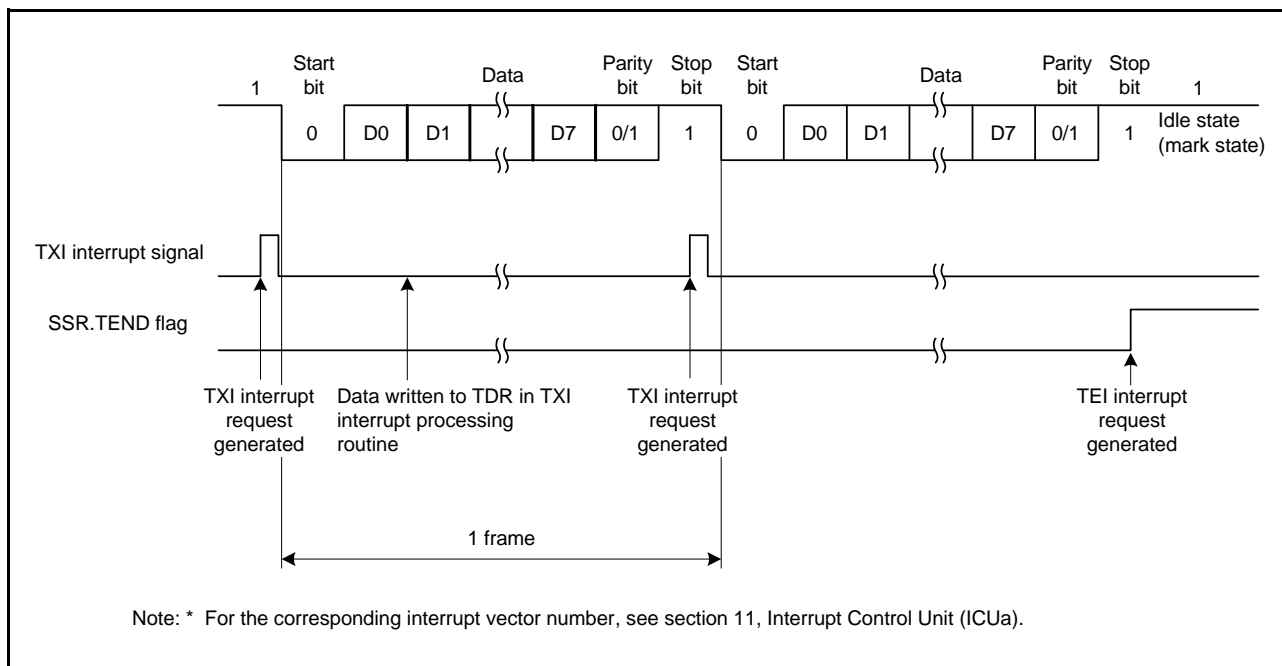


Figure 29.8 Example of Operation for Serial Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

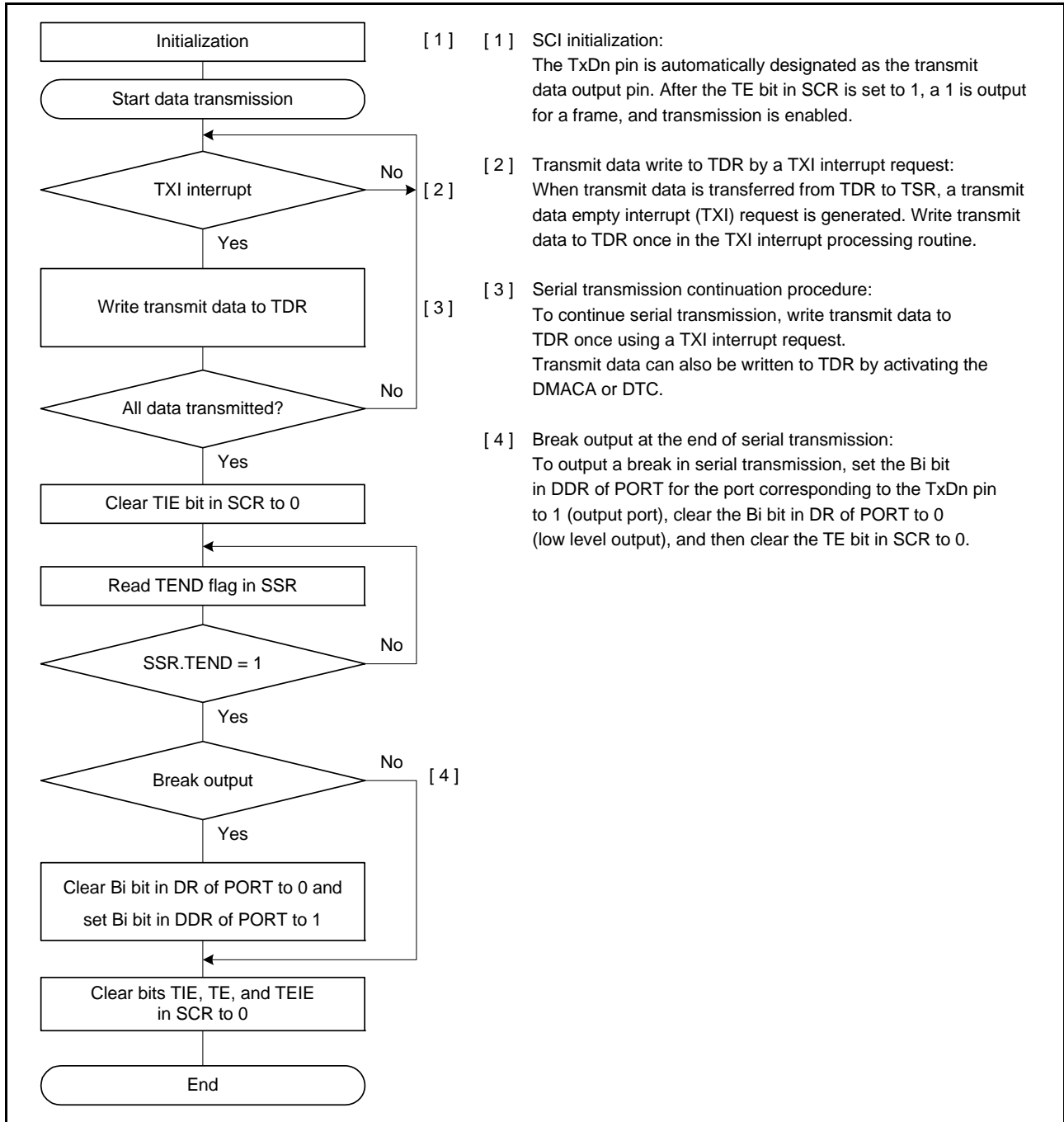


Figure 29.9 Example of Serial Transmission Flowchart in Asynchronous Mode

29.2.2.6 Serial Data Reception (Asynchronous Mode)

Figure 29.10 shows 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 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.
2. 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.
3. 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.
4. 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.
5. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt processing routine before reception of the next receive data is completed.

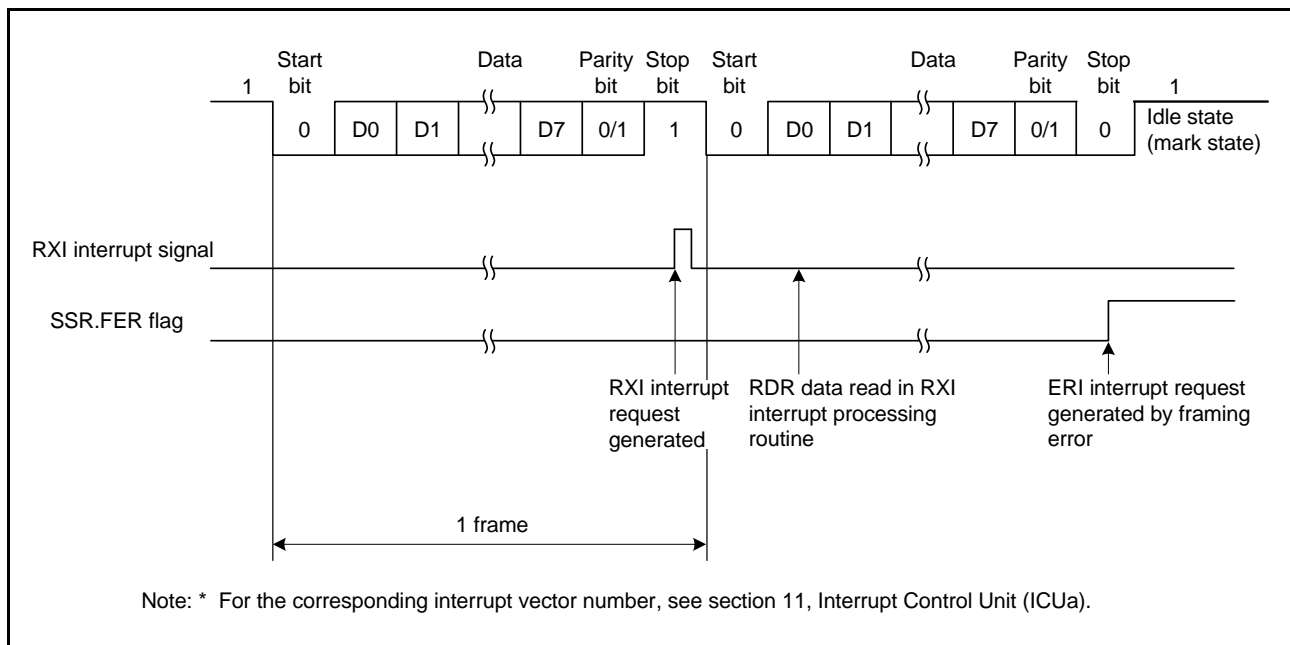


Figure 29.10 Example of SCI Operation for Serial Reception in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

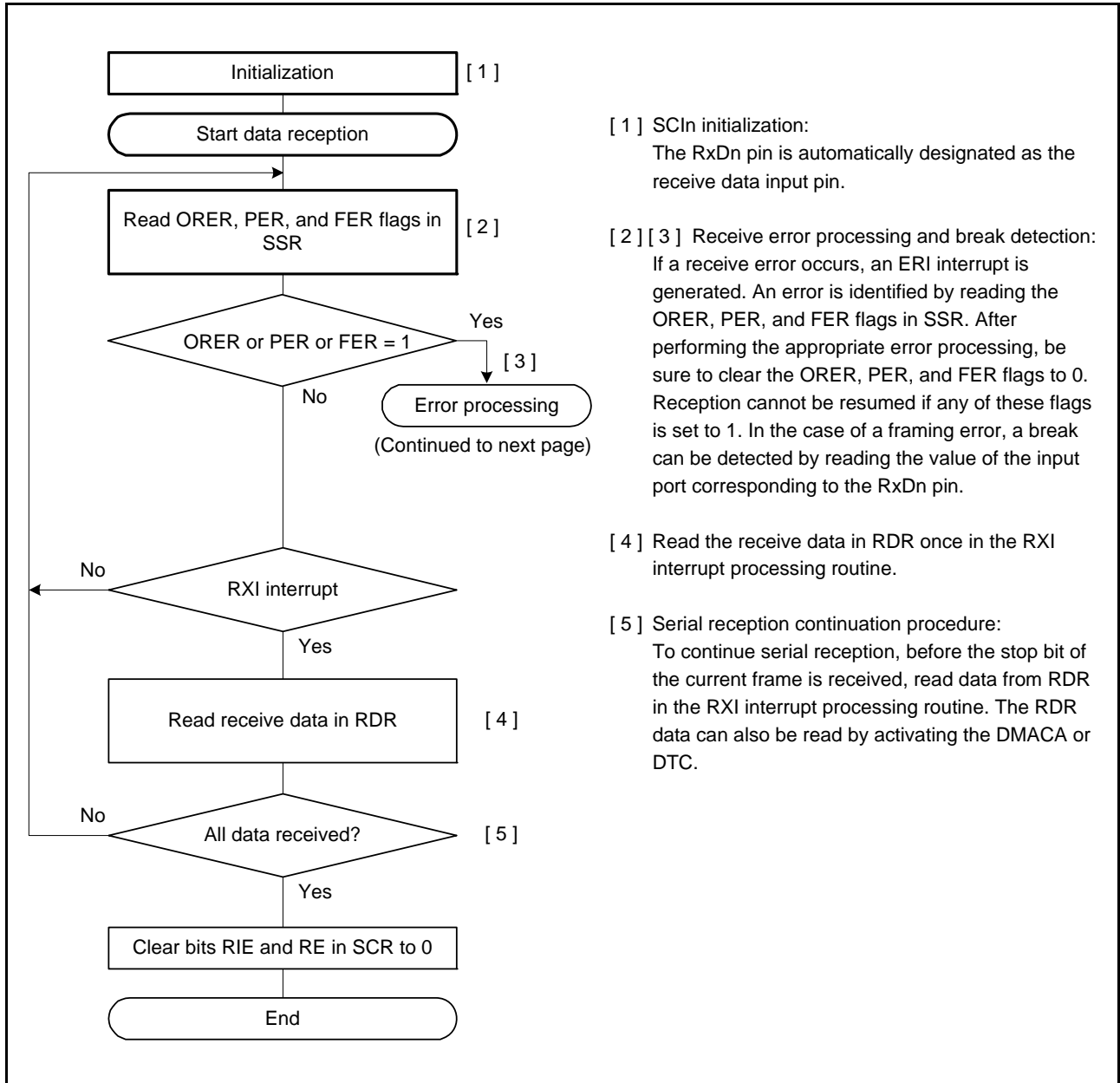
Table 29.12 shows the states of the SSR status flags and receive data handling when a receive error is detected.

If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, clear the ORER, FER, and PER bits to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing.

Figure 29.11 shows samples of flowcharts for serial data reception.

Table 29.12 SSR Status Flags and Receive Data Handling

SSR Status Flag			Receive Data	Receive Error Type
ORER	FER	PER		
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error



[1] SCIn initialization:

The RxDn pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing and break detection:

If a receive error occurs, an ERI interrupt is generated. An error is identified by reading the ORER, PER, and FER flags in SSR. After performing the appropriate error processing, be sure to clear the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxDn pin.

[4] Read the receive data in RDR once in the RXI interrupt processing routine.

[5] Serial reception continuation procedure:

To continue serial reception, before the stop bit of the current frame is received, read data from RDR in the RXI interrupt processing routine. The RDR data can also be read by activating the DMACA or DTC.

Figure 29.11 Example of Serial Reception Flowchart (1) (Asynchronous Mode)

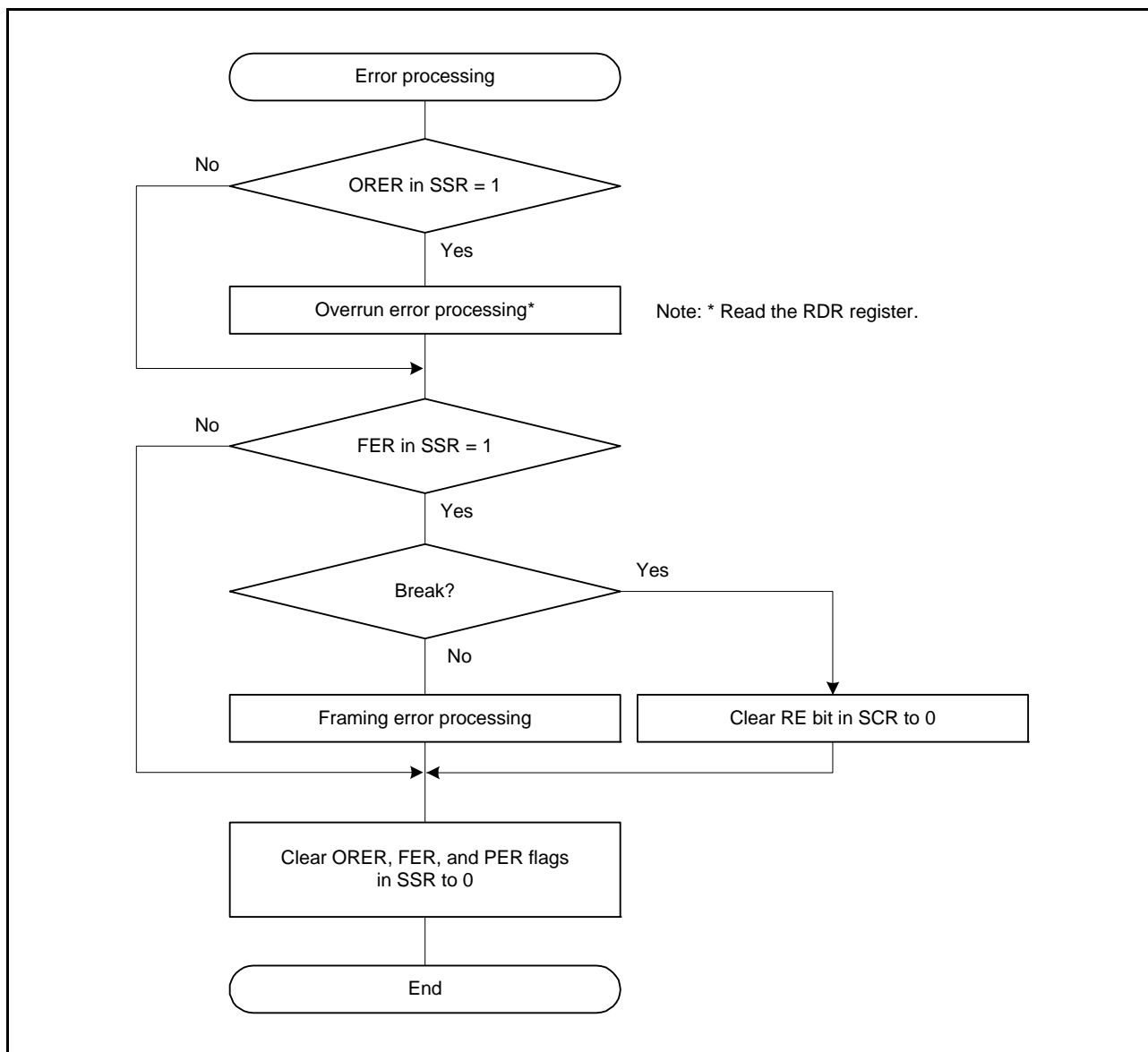


Figure 29.11 Example of Serial Reception Flowchart (2) (Asynchronous Mode)

29.2.3 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 29.12 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmission data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two match, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.

For supporting this function, the SCI provides the MPIE bit in SCR. When the MPIE bit is set to 1, transfer of reception data from the RSR to the RDR, detection of a reception error, and setting the respective status flags ORER and FER in SSR are disabled until reception of data in which the multi-processor bit is set to 1. Upon receiving a reception character in which the multi-processor bit is set to 1, the MPB 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.

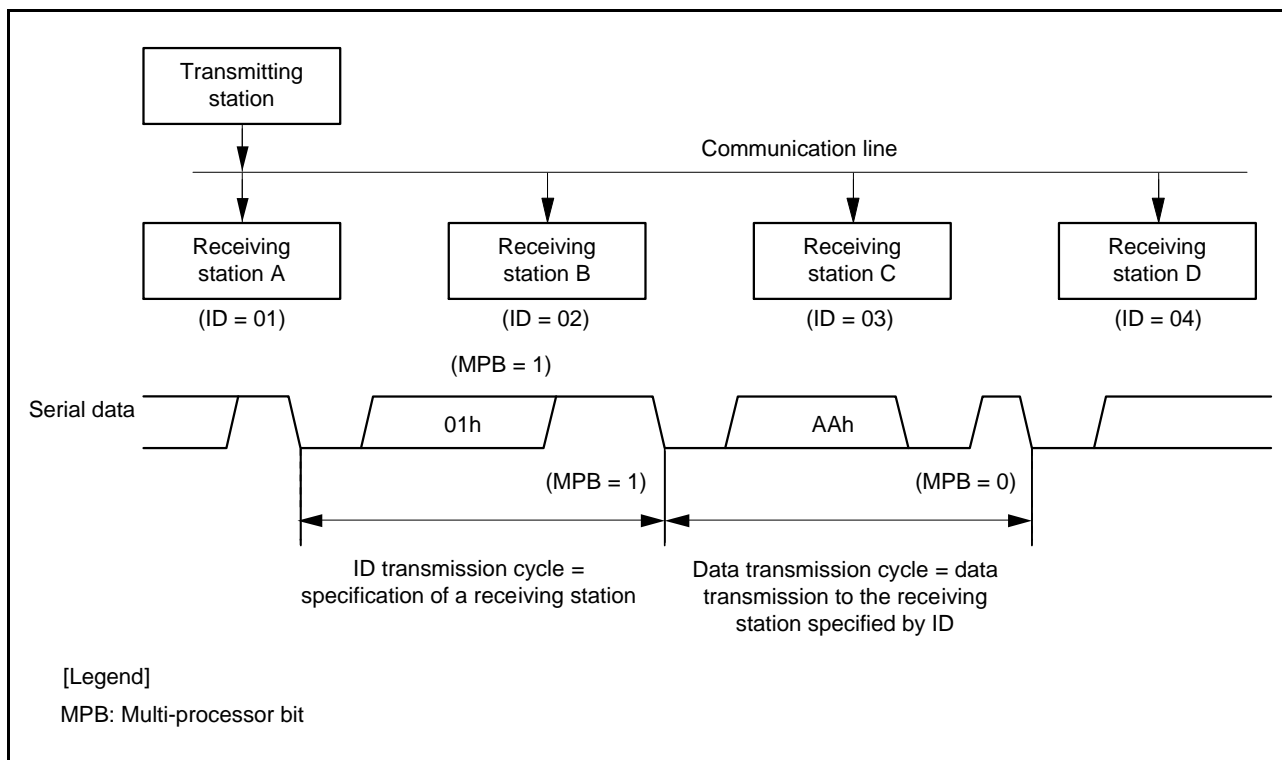


Figure 29.12 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

29.2.3.1 Multi-Processor Serial Data Transmission

Figure 29.13 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.

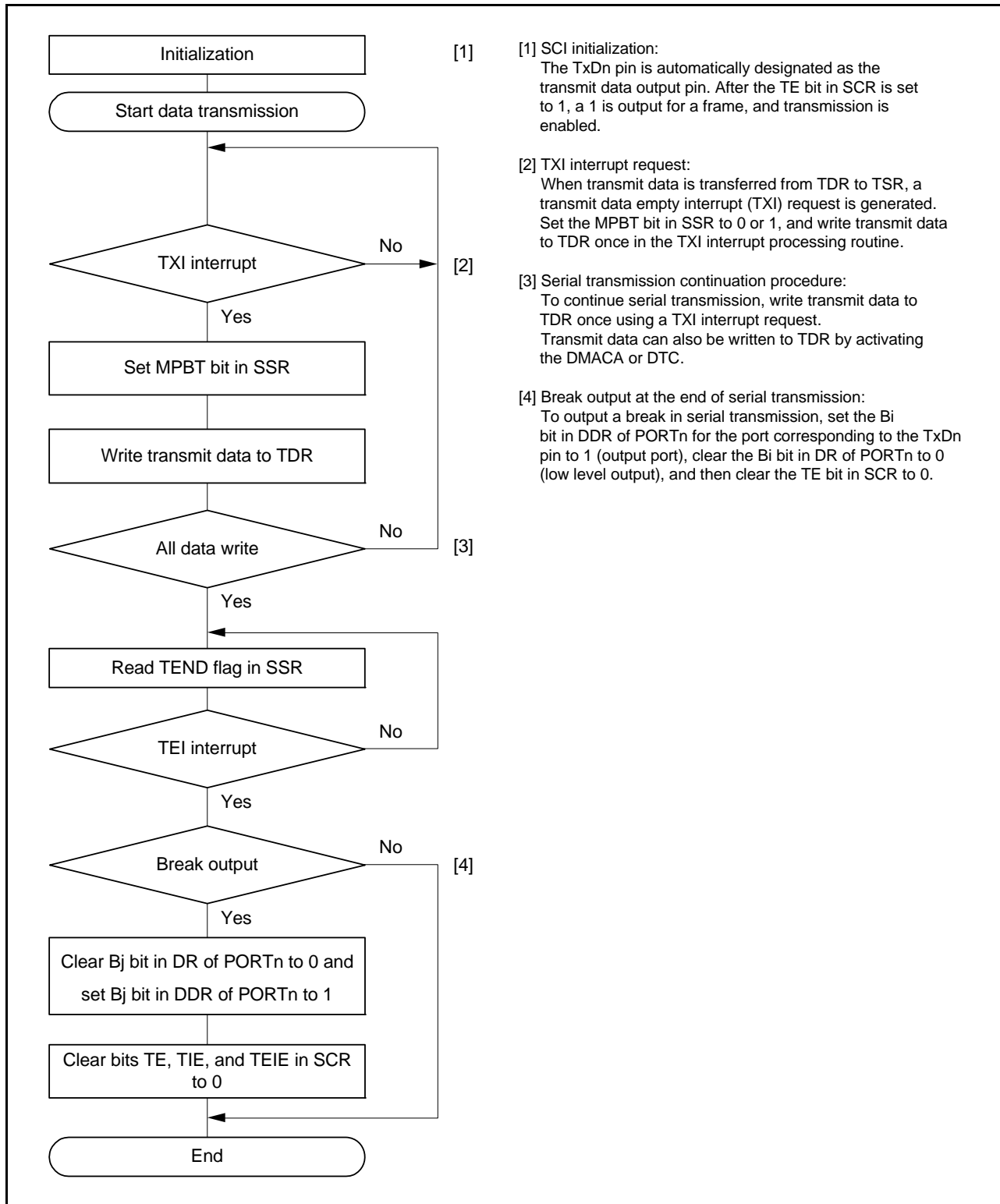


Figure 29.13 Example of Multi-Processor Serial Transmission Flowchart

29.2.3.2 Multi-Processor Serial Data Reception

Figure 29.15 and Figure 29.16 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 29.14 is the example of operation for reception.

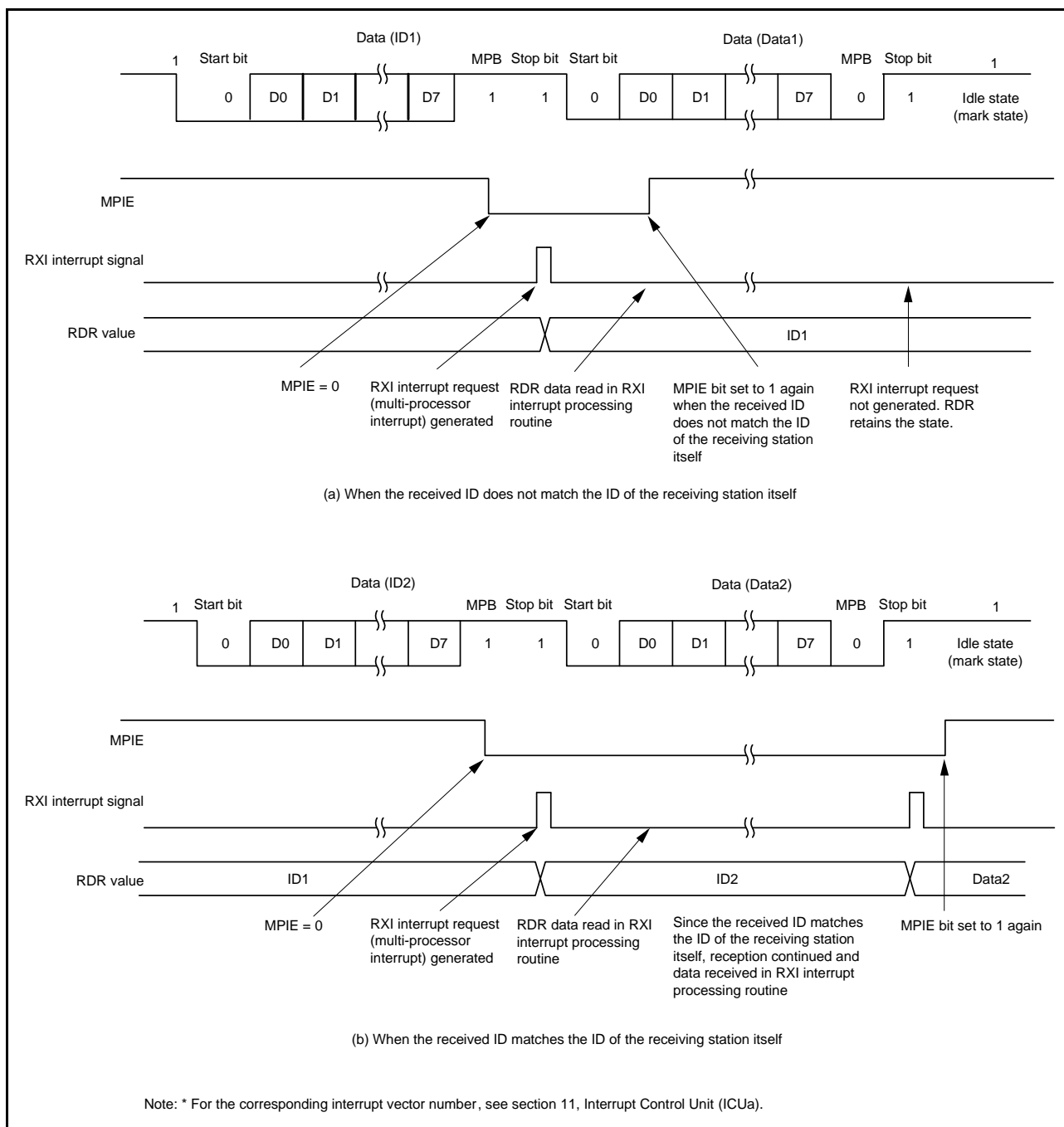


Figure 29.14 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/One Stop Bit)

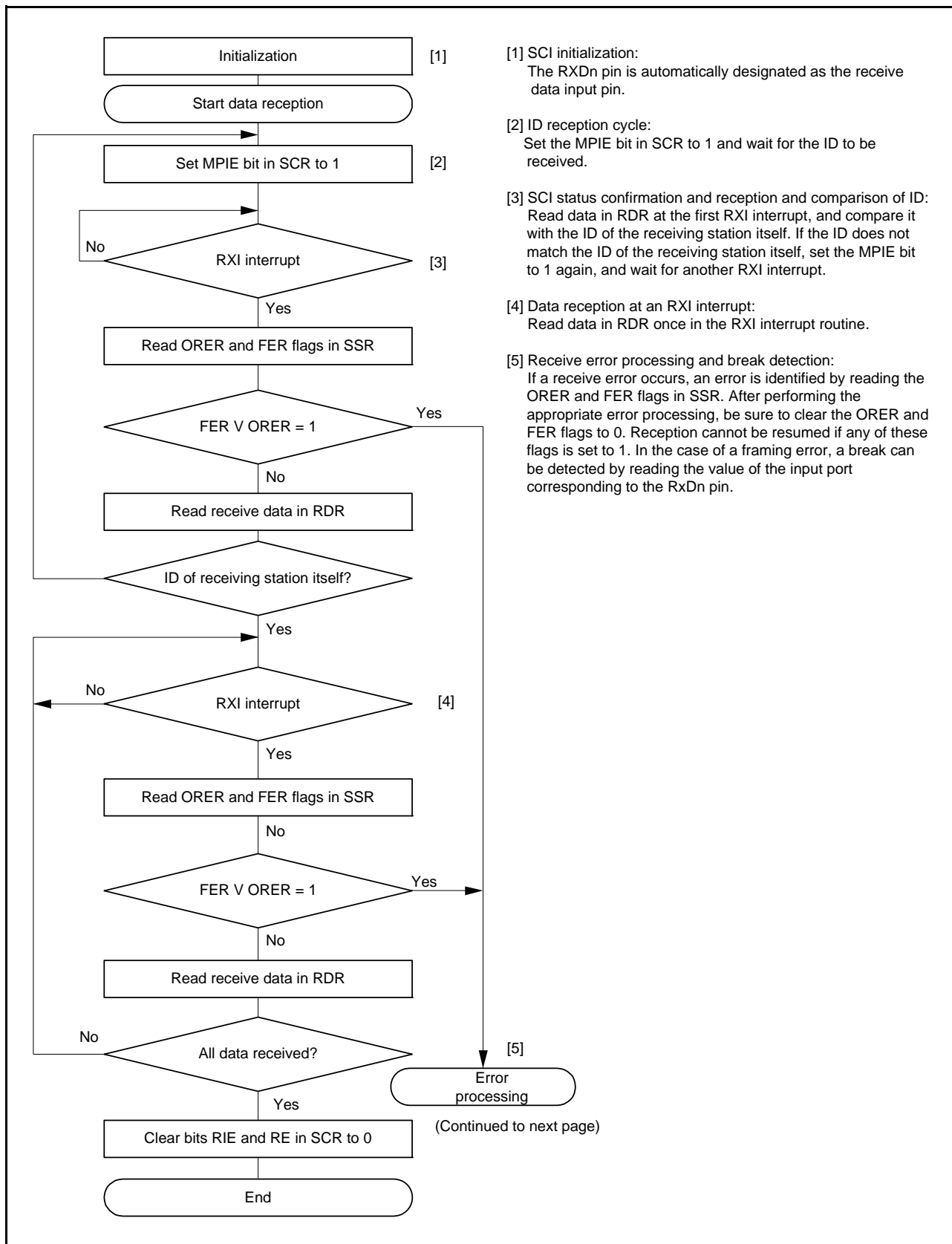


Figure 29.15 Example of Multi-Processor Serial Reception Flowchart (1)

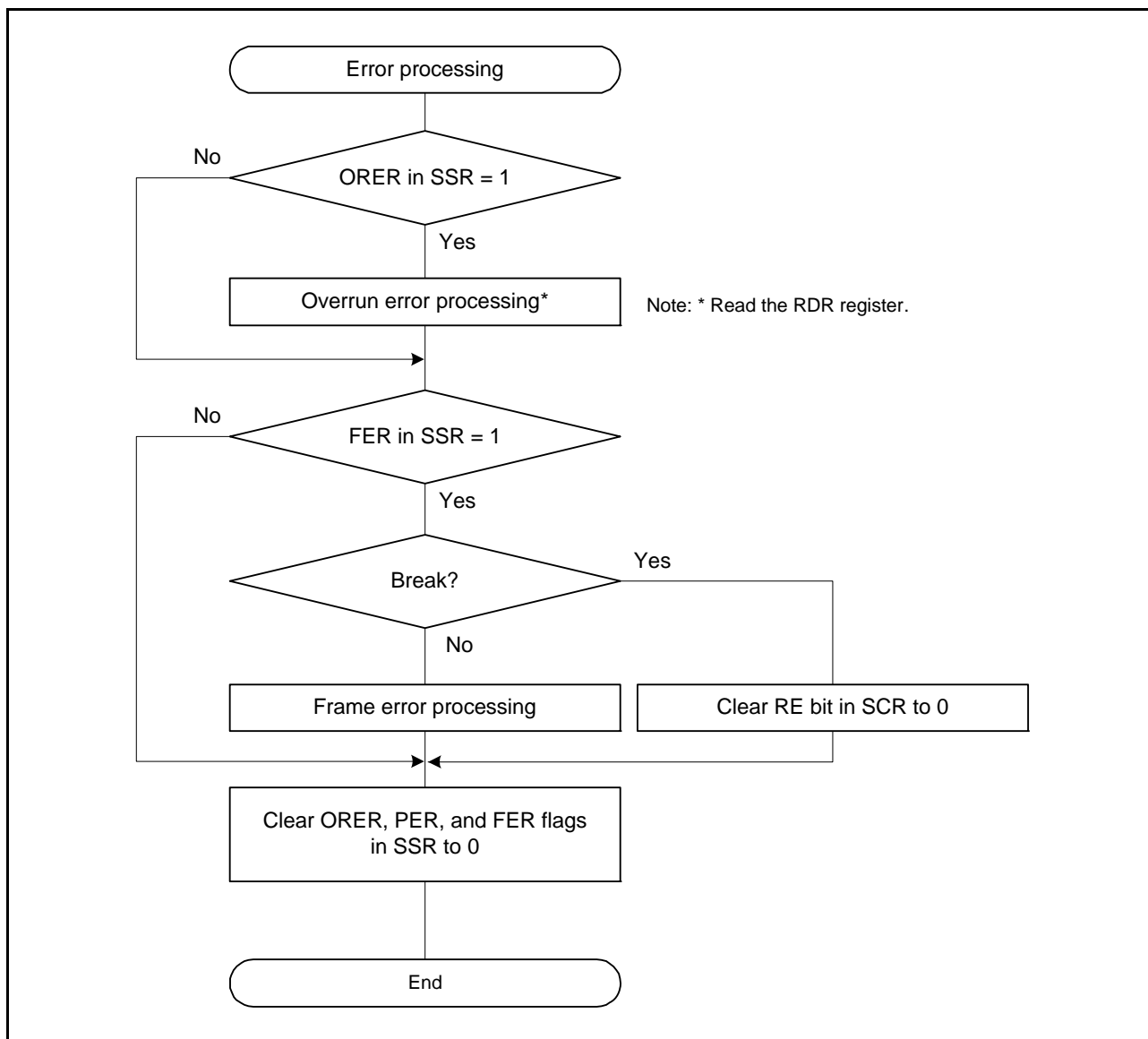


Figure 29.16 Example of Multi-Processor Serial Reception Flowchart (2)

29.2.4 Operation in Clock Synchronous Mode

Figure 29.17 shows the data format for clock synchronous serial data communications.

In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.

In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the MSB 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.

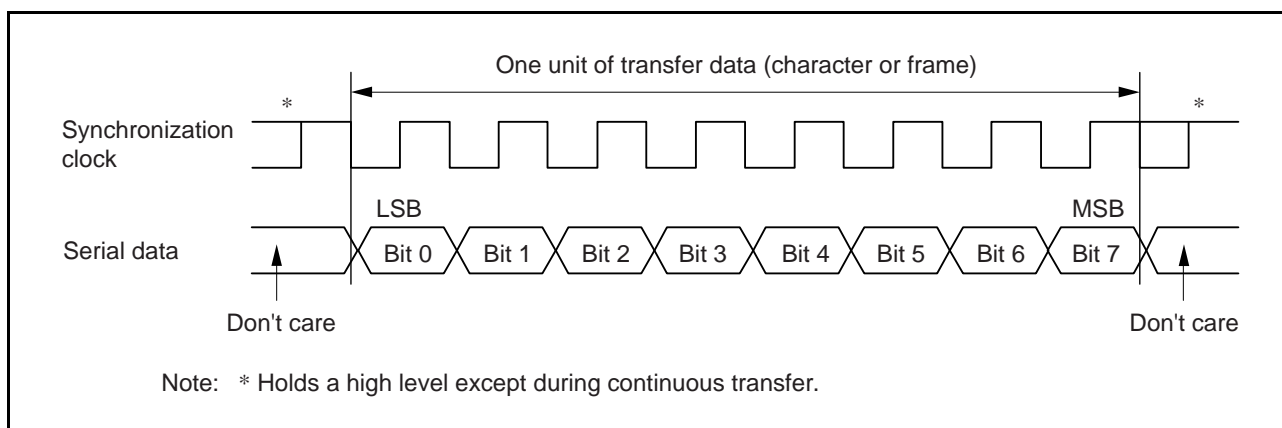


Figure 29.17 Data Format in Clock Synchronous Serial Communications (LSB-First)

29.2.4.1 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 CKE[1:0] bits in SCR.

When the SCI is operated on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is held high. However, the synchronization clock is continuously output only during data reception until an overrun error occurs or the RE bit in SCR is cleared to 0.

29.2.4.2 SCI Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value "00h" to the SCR and then continue through the procedure for SCI given in the sample flowchart (Figure 29.18). Whenever the operating mode or transfer format is changed, the SCR must be initialized before the change is made.

Note that clearing the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR nor RDR.

Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.

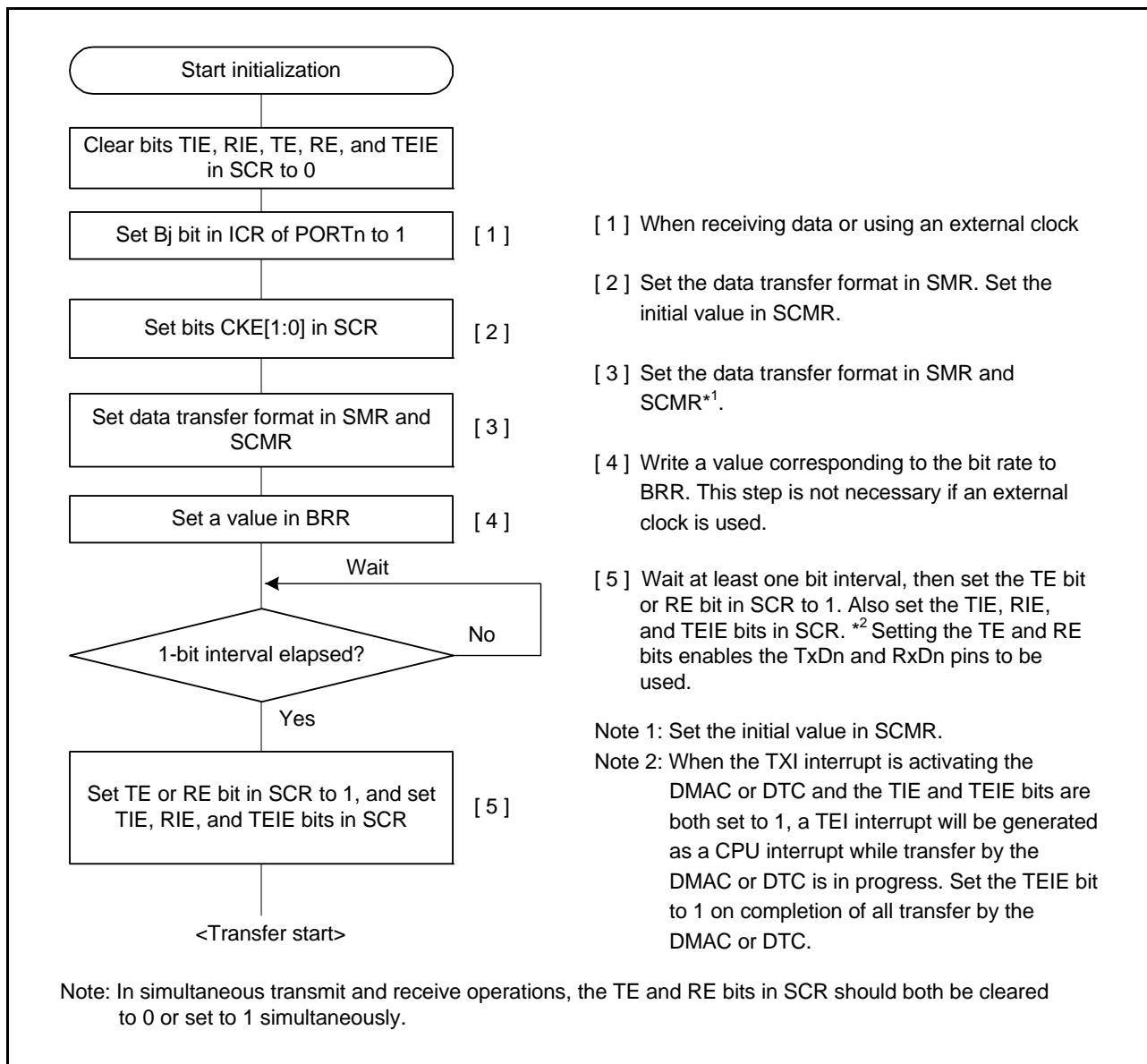


Figure 29.18 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

29.2.4.3 Serial Data Transmission (Clock Synchronous Mode)

Figure 29.19 shows an example of the operation for serial transmission in clock synchronous mode.

In serial data transmission, the SCI operates as described below.

1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt processing routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. After transferring data from TDR to TSR, the SCI starts transmission. When the TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in this TXI interrupt processing routine before transmission of the current transmit data has finished.
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.
4. The SCI checks for updating of (writing to) the TDR at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If TDR is not updated, set the SSR flag in TEND to 1 and the TxDn pin retains the output state of the last bit. If the TEIE bit in SCR is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 29.20 shows a sample flowchart of serial data transmission.

Transmission will not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Be sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit in SCR to 0 does not clear the receive error flags.

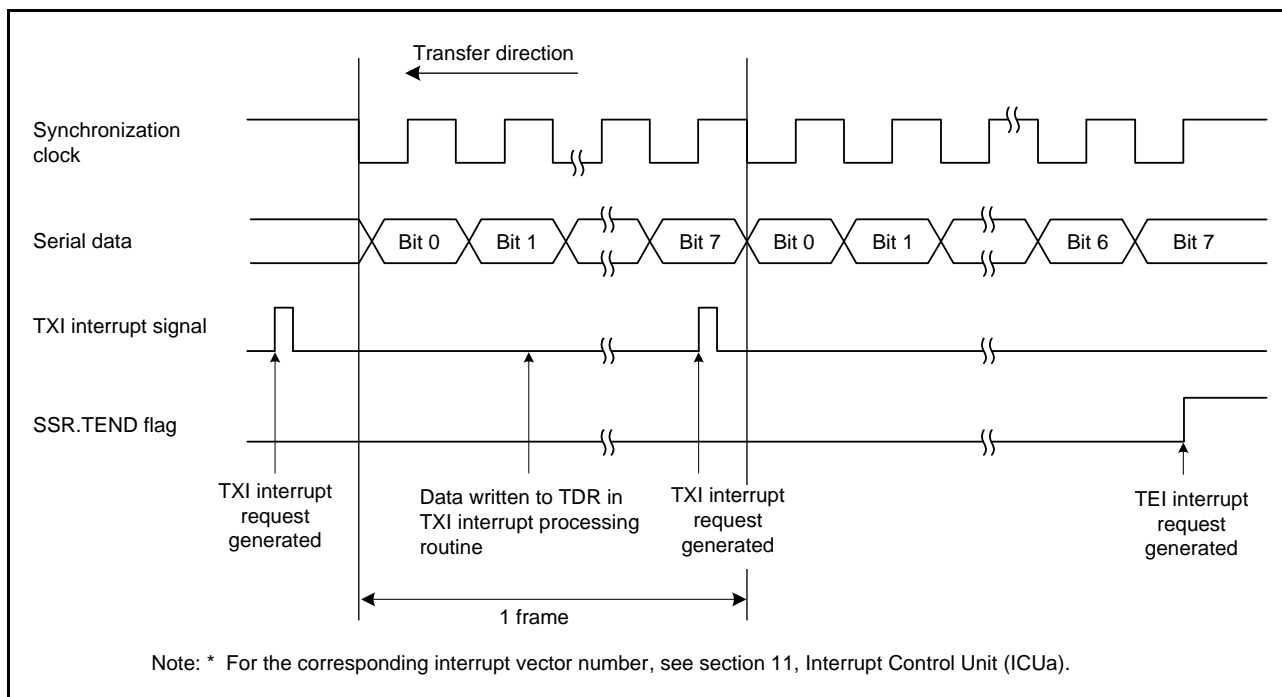


Figure 29.19 Example of Operation for Serial Transmission in Clock Synchronous Mode

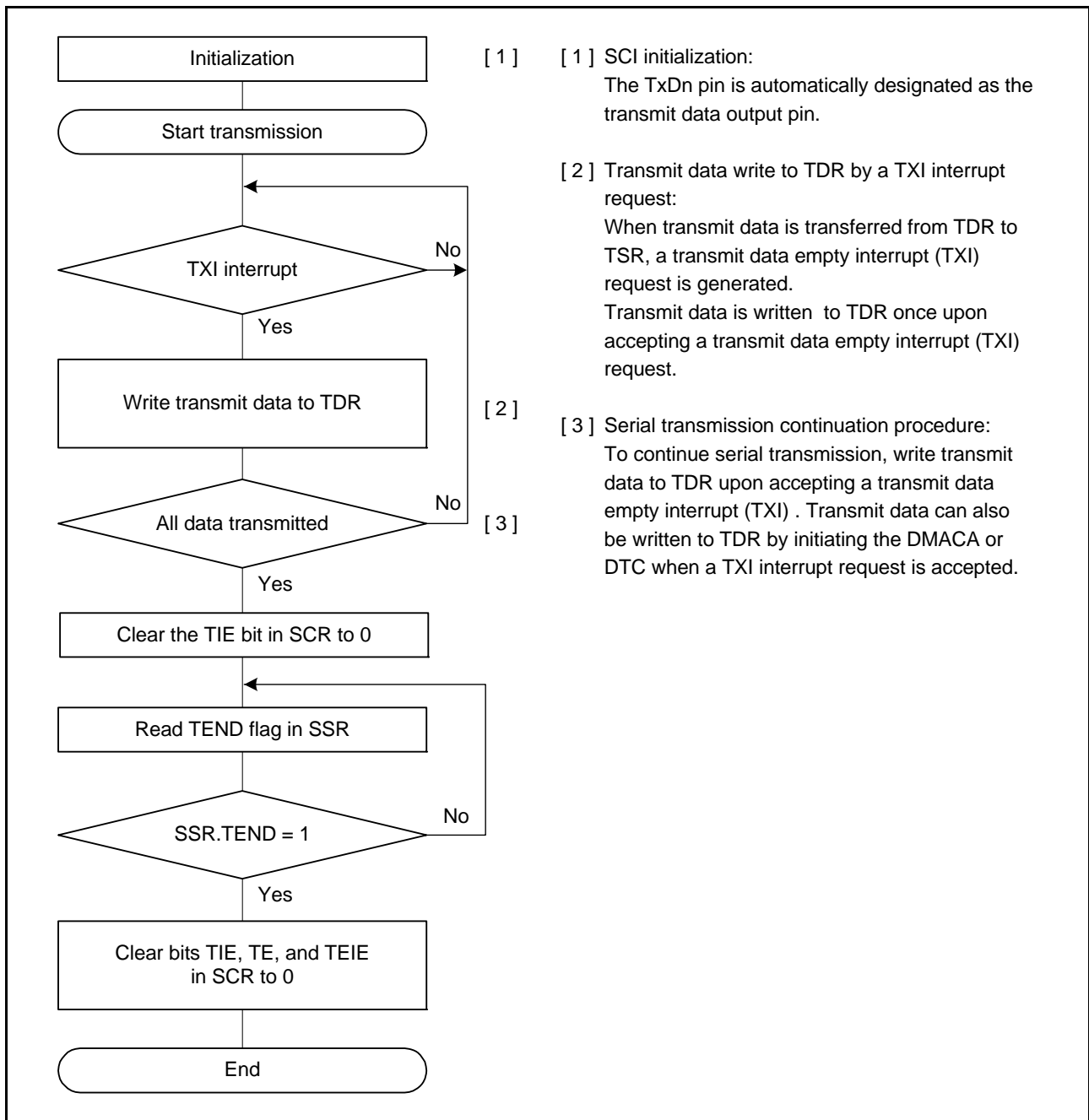


Figure 29.20 Example of Serial Transmission Flowchart (Clock Synchronous Mode)

29.2.4.4 Serial Data Reception (Clock Synchronous Mode)

Figure 29.21 shows an example of SCI operation for serial reception in clock synchronous mode.

In serial data reception, the SCI operates as described below.

1. 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.
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.
2. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt processing routine before reception of the next receive data is completed.

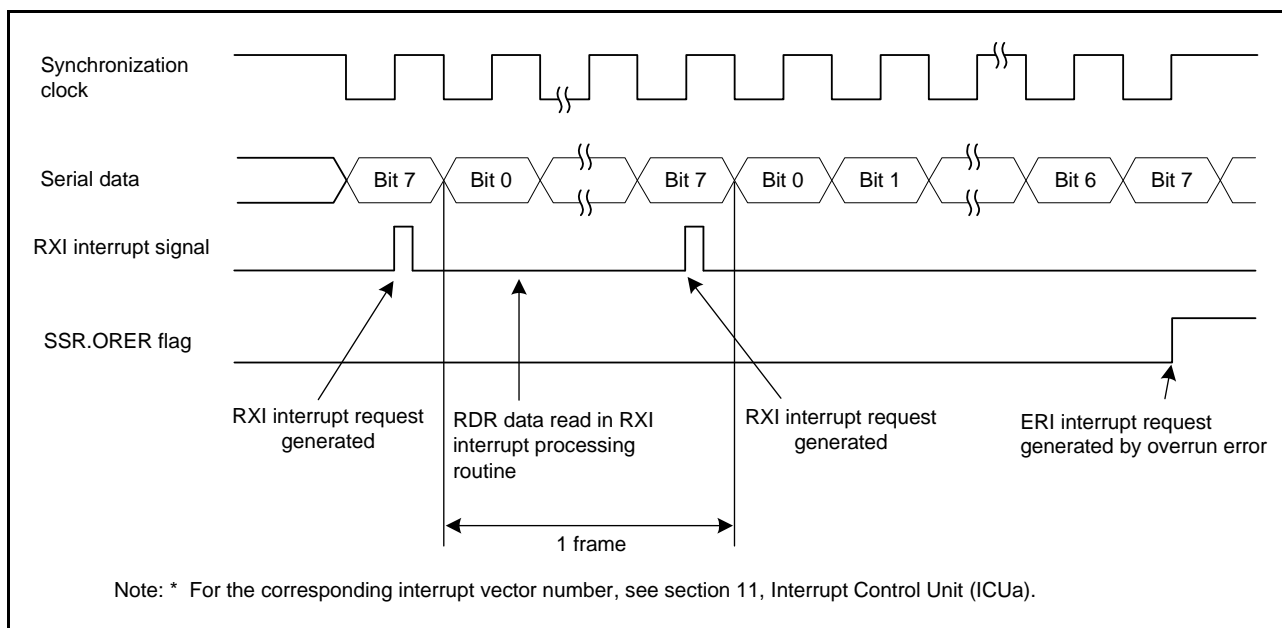


Figure 29.21 Example of Operation for Serial Reception in Clock Synchronous Mode

Data transfer cannot be resumed while a receive error flag is 1. Accordingly, clear the ORER, FER, and PER bits in SSR to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing.

Figure 29.22 shows a sample flowchart for serial data reception.

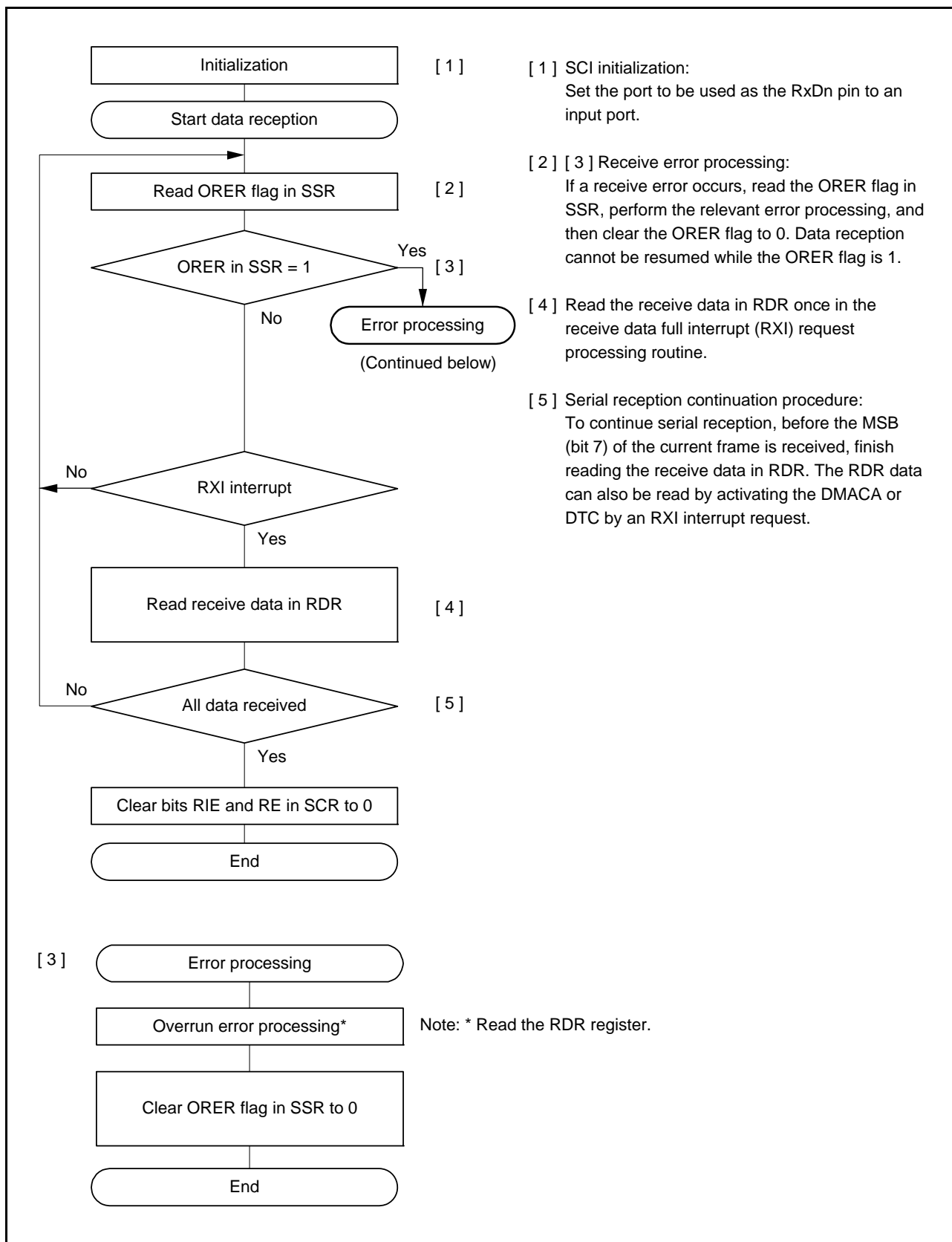


Figure 29.22 Example of Serial Reception Flowchart (Clock Synchronous Mode)

29.2.4.5 Serial Data Full-Duplex Communications (Clock Synchronous Mode)

Figure 29.23 shows a sample flowchart for serial data full-duplex communications in clock synchronous mode.

After initializing the SCI, the following procedure should be used for serial data full-duplex communications.

To switch from transmit mode to serial data full-duplex communications, 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, RE, and TEIE bits in SCR to 1 simultaneously by a single instruction.

To switch from receive mode to serial data full-duplex communications, check that the SCI has finished reception, and then clear the RIE and RE bits to 0. Then check that the receive error flags (ORER, FER, and PER in SSR) are 0, and then set the TIE, RIE, TE, RE, and TEIE bits in SCR to 1 simultaneously by a single instruction.

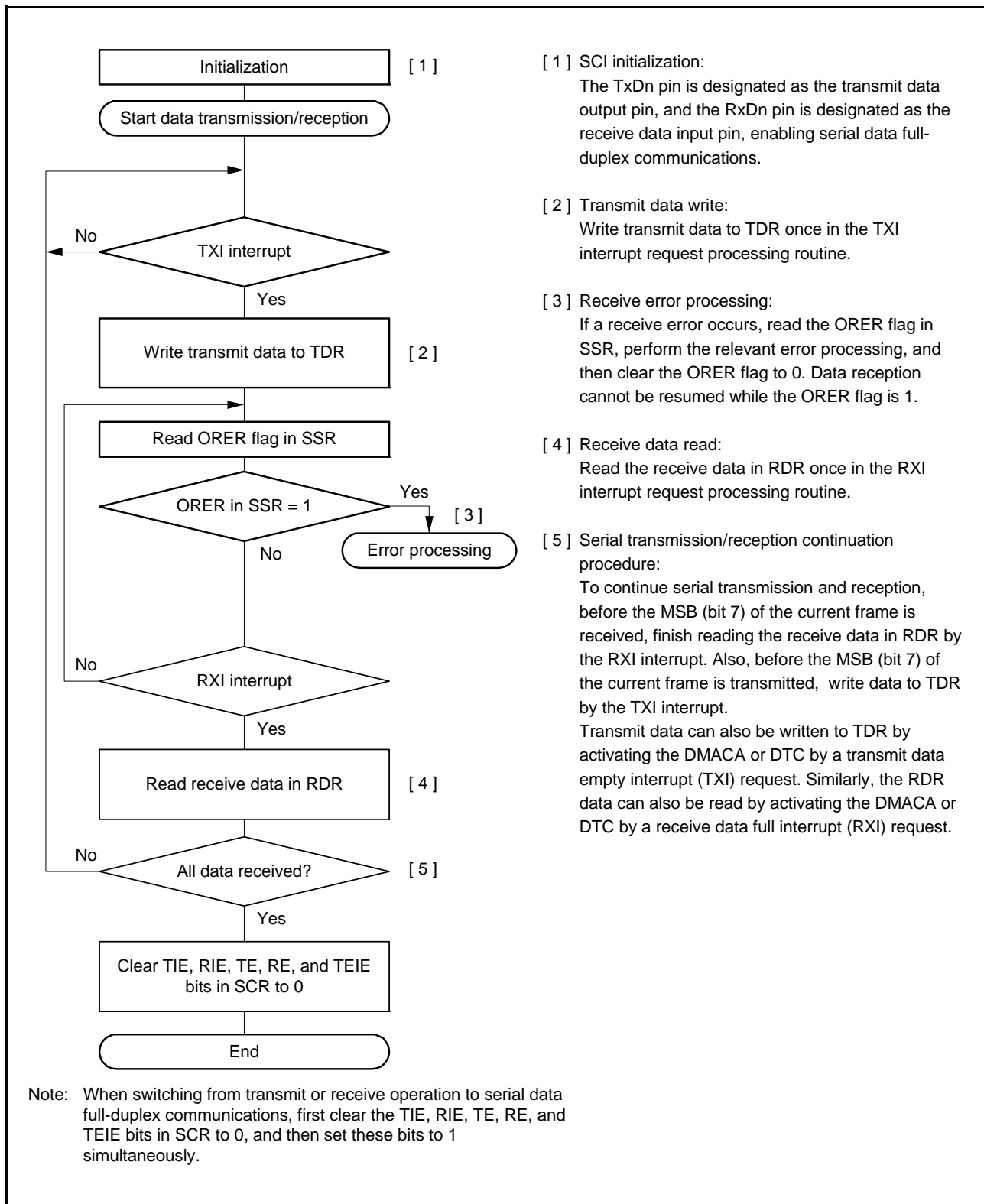


Figure 29.23 Example of Simultaneous Serial Transmission and Reception Flowchart (Clock Synchronous Mode)

29.3 Smart-Card Interface Mode

As an extended function of the SCI, it is capable of operating in compliance with ISO/IEC 7816-3 (Identification Cards) as an interface with smart cards (IC cards).

The SCI runs in smart-card interface mode when the setting of the SCMR.SMIF bit is "1".

29.3.1 Register Descriptions

Table 29.13 lists the registers of the SMCI. Some registers (TDR, RDR, and SCMR) have same functions in serial communications interface mode and smart card interface mode. Therefore, for the descriptions on the TDR, RDR, and SCMR registers, see section 29.2.1, Register Descriptions.

Table 29.13 Registers of SMCI

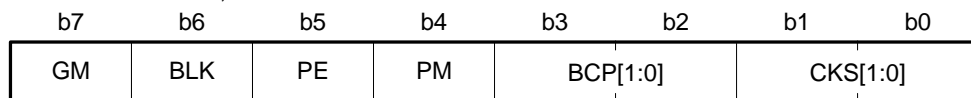
Channel	Register Name	Symbol	Value after Reset	Address	Access Size
SCI0	Serial mode register	SMR	00h	0008 8240h	8
	Bit rate register	BRR	FFh	0008 8241h	8
	Serial control register	SCR	00h	0008 8242h	8
	Transmit data register	TDR	FFh	0008 8243h	8
	Serial status register	SSR	84h	0008 8244h	8
	Receive data register	RDR	00h	0008 8245h	8
	Smart card mode register	SCMR	F2h	0008 8246h	8
SCI1	Serial mode register	SMR	00h	0008 8248h	8
	Bit rate register	BRR	FFh	0008 8249h	8
	Serial control register	SCR	00h	0008 824Ah	8
	Transmit data register	TDR	FFh	0008 824Bh	8
	Serial status register	SSR	84h	0008 824Ch	8
	Receive data register	RDR	00h	0008 824Dh	8
	Smart card mode register	SCMR	F2h	0008 8246h	8
SCI2	Serial mode register	SMR	00h	0008 8250h	8
	Bit rate register	BRR	FFh	0008 8251h	8
	Serial control register	SCR	00h	0008 8252h	8
	Transmit data register	TDR	FFh	0008 8253h	8
	Serial status register	SSR	84h	0008 8254h	8
	Receive data register	RDR	00h	0008 8255h	8
	Smart card mode register	SCMR	F2h	0008 825Eh	8
SCI3	Serial mode register	SMR	00h	0008 8258h	8
	Bit rate register	BRR	FFh	0008 8259h	8
	Serial control register	SCR	00h	0008 825Ah	8
	Transmit data register	TDR	FFh	0008 825Bh	8
	Serial status register	SSR	84h	0008 825Ch	8
	Receive data register	RDR	00h	0008 825Dh	8
	Smart card mode register	SCMR	F2h	0008 825Eh	8
SCI5	Serial mode register	SMR	00h	0008 8268h	8
	Bit rate register	BRR	FFh	0008 8269h	8
	Serial control register	SCR	00h	0008 826Ah	8
	Transmit data register	TDR	FFh	0008 826Bh	8
	Serial status register	SSR	84h	0008 826Ch	8
	Receive data register	RDR	00h	0008 826Dh	8
	Smart card mode register	SCMR	F2h	0008 826Eh	8

Table 29.13 Registers of SMCI

Channel	Register Name	Symbol	Value after Reset	Address	Access Size
SCI6	Serial mode register	SMR	00h	0008 8270h	8
	Bit rate register	BRR	FFh	0008 8271h	8
	Serial control register	SCR	00h	0008 8272h	8
	Transmit data register	TDR	FFh	0008 8273h	8
	Serial status register	SSR	84h	0008 8274h	8
	Receive data register	RDR	00h	0008 8275h	8
	Smart card mode register	SCMR	F2h	0008 8276h	8

29.3.1.1 Serial Mode Register (SMR)

Addresses: SCI0.SMR 0008 8240h, SCI1.SMR 0008 8248h, SCI2.SMR 0008 8250h, SCI3.SMR 0008 8258h
SCI5.SMR 0008 8268h, SCI6.SMR 0008 8270h



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

Bit	Symbol	Bit Name	Function	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 00: PCLK clock (n = 0)*1 01: PCLK/4 clock (n = 1)*1 10: PCLK/16 clock (n = 2)*1 11: PCLK/64 clock (n = 3)*1	R/W*3
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the BCP2 bit in SCMR. Setting values in BCP2 bit in SCMR and BCP[1:0] bits in SMR: BCP2 b3 b2 0 0 0: 93 clock cycles (S = 93)*2 0 0 1: 128 clock cycles (S = 128)*2 0 1 0: 186 clock cycles (S = 186)*2 0 1 1: 512 clock cycles (S = 512)*2 1 0 0: 32 clock cycles (S = 32)*2 (Initial value) 1 0 1: 64 clock cycles (S = 64)*2 1 1 0: 372 clock cycles (S = 372)*2 1 1 1: 256 clock cycles (S = 256)*2	R/W*3
b4	PM	Parity Mode	(Valid only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity	R/W*3
b5	PE	Parity Enable	(Valid only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. Set this bit to 1 in smart card interface mode.	R/W*3
b6	BLK	Block Transfer Mode	0: Normal mode operation 1: Block transfer mode operation	R/W*3
b7	GM	GSM Mode	0: Normal mode operation 1: GSM mode operation	R/W*3

Note 1. n is the decimal notation of the value of n in BRR (see section 29.2.1.9, Bit Rate Register (BRR)).

Note 2. S indicates the value of S in BRR (see section 29.2.1.9, Bit Rate Register (BRR)).

Note 3. Writing to these bits is only possible when the TE and RE bits in the SCR are set to "0" (disabling both serial transmission and reception operations).

SMR is used to set the SCI's serial transfer format and select the clock source for the on-chip baud rate generator.

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relation between the settings of these bits and the baud rate, see section 29.2.1.9, Bit Rate Register (BRR).

BCP[1:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

Set these bits in combination with the BCP2 bit in SCMR.

For details, see section 29.3.3.2, Receive Data Sampling Timing and Reception Margin.

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd).

For details on the usage of this bit in smart card interface mode, section 29.3.3, Data Format (Except in Block Transfer Mode).

PE Bit (Parity Enable)

Set the PE bit to 1.

The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.

For details, see section 29.3.3.1, Block Transfer Mode.

GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.

In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, see section 29.3.3.4, Serial Data Transmission (Except in Block Transfer Mode) and section 29.3.3.6, Clock Output Control.

29.3.1.2 Serial Control Register (SCR)

Addresses: SCI0.SCR 0008 8242h, SCI1.SCR 0008 824Ah, SCI2.SCR 0008 8295h, SCI3.SCR 0008 25Ah
SCI5.SCR 0008 826Ah, SCI6.SCR 0008 8272h

b7	b6	b5	b4	b3	b2	b1	b0
TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	

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

Bit	Symbol	Bit Name	Function	R/W
b1, b0	CKE[1:0]	Clock Enable	<ul style="list-style-type: none"> When GM in SMR = 0 <ul style="list-style-type: none"> b1 b0 0 0: Output disabled (SCKn pin functions as I/O port.) 0 1: Clock output 1 0: (Setting prohibited) 1 1: (Setting prohibited) When GM in SMR = 1 <ul style="list-style-type: none"> 0 0: Output fixed low 0 1: Clock output 1 0: Output fixed high 1 1: Clock output 	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

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. A 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.

SCR is a register that enables/disables the SCI transfer operations and the interrupt requests and selects the transfer clock source. For details on interrupt requests, see section 29.4, Interrupt Sources.

CKE[1:0] Bits (Clock Enable)

These bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, see section 29.3.3.6, Clock Output Control.

TEIE Bit (Transmit End Interrupt Enable)

This bit should be 0 in smart card interface mode.

MPIE Bit (Multi-Processor Interrupt Enable)

This bit should be 0 in smart card interface mode.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit. Note that SMR should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by clearing the RE bit to 0, the ORER, FER, and PER flags in SSR are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to TDR. Note that SMR should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by clearing the RIE bit to 0.

An ERI interrupt request can be cancelled by reading 1 from the ORER, FER, or PER flag in SSR and then clearing the flag to 0, or clearing the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables notification of a TXI interrupt request.

Notification of a TXI interrupt request is disabled by clearing the TIE bit to 0.

29.3.1.3 Serial Status Register (SSR)

Addresses: SCI0.SSR 0008 8244h, SCI1.SSR 0008 824Ch, SCI2.SSR 0008 8254h, SCI3.SSR 000 825Ch
SCI5.SSR 0008 826Ch, SCI6.SSR 0008 8274h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	MPBT	Multi-Processor Bit Transfer	This bit should be set to 0 in smart card interface mode.	R/W
b1	MPB	Multi-Processor	This bit is not used in smart card interface mode. It should be set to 0.	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	ERS	Error Signal Status Flag	0: Low error signal not responded 1: Low error signal responded	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: When data is transferred from RDR 1: When data has been received normally, and transferred from RSR to RDR	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: When data is transferred to TDR 1: When data is transferred from TDR to TSR	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag.

Note 2. Write 1 when writing is necessary.

SSR is a register containing status flags of the SCI.

MPBT Bit (Multi-Processor Bit Transfer)

This bit should be set to 0 in smart card interface mode.

MPB Bit (Multi-Processor)

This bit is not used in smart card interface mode. It should be set to 0.

TEND Flag (Transmission End Flag)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When SCR.TE bit = 0 (disabling serial transmission operations)
- When a specified period has elapsed after the latest transmission of one byte, the ERS flag is 0, and the TDR register is not updated

The set timing is determined by register settings as listed below.

When SMR.GM = 0 and SMR.BLK = 0, 12.5 etu after the start of transmission

When SMR.GM = 0 and SMR.BLK = 1, 11.5 etu after the start of transmission

When SMR.GM = 1 and SMR.BLK = 0, 11.0 etu after the start of transmission

When SMR.GM = 1 and SMR.BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- Writing further data for transmission to the TDR register

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 serial reception cannot be transferred to RDR.

[Clearing condition]

- When a 0 is written to PER after reading PER = 1 (After writing a 0 to it, read the PER bit to check that it has actually been cleared to 0.)
Even when the RE bit in SCR is cleared to 0 (which indicates that serial reception is disabled), the PER flag is not affected and retains its previous value.

ERS Flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled

[Clearing condition]

- When a 0 is written to ERS after reading ERS = 1

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR
In RDR, the receive data prior to an overrun error occurrence is retained, but data received following the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed.

[Clearing condition]

- When a 0 is written to ORER after reading ORER = 1 (After writing a 0 to it, read the ORER bit to check that it has actually been cleared to 0.)
Even when the RE bit in SCR is cleared to 0, the ORER flag is not affected and retains its previous value.

RDRF Bit (Receive Data Full Flag)

Indicates whether RDR has received data.

[Setting condition]

- When data has been received normally, and transferred from RSR to RDR

[Clearing condition]

- When data is transferred from RDR

TDRE Bit (Transmit Data Empty Flag)

Indicates whether TDR has data to be transmitted.

[Setting condition]

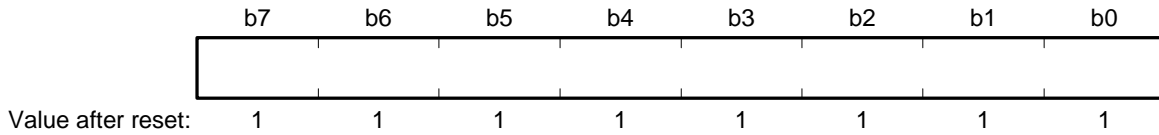
- When data is transferred from TDR to TSR

[Clearing condition]

- When data is transferred to TDR

29.3.1.4 Bit Rate Register (BRR)

Addresses: SCI0.BRR 0008 8241h, SCI1.BRR 0008 8249h, SCI2.BRR 0008 8251h,
 SCI3.BRR 0008 8259h, SCI5.BRR 0008 8269h, SCI6.BRR 0008 8271h



BRR is an 8-bit register that adjusts the bit rate.

As the SCI performs baud rate generator control independently for each channel, different bit rates can be set for each channel. Table 29.14 shows the relationships between the N setting in BRR and bit rate B for smart card interface mode. The initial value of BRR is FFh.

BRR can be read from by the CPU at all times, but it can be written to only when the TE and RE bits in SCR are 0.

Table 29.14 Relationships between N Setting in BRR and Bit Rate B

Mode	BRR Setting	Error
Smart card interface	$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$

[Legend]

- B: Bit rate (bps)
- N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
- PCLK: Operating frequency (MHz)
- n and S: Determined by the SMR setting shown in the following table.

Table 29.15 shows sample N settings in BRR in smart card interface mode. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 29.3.3.2, Receive Data Sampling Timing and Reception Margin. Table 29.16 shows the maximum bit rate for each operating frequency.

Table 29.15 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	7.1424			10.00			10.7136			13.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	30	0	1	25	0	1	8.99

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	14.2848			16.00			18.00			20.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	15.99	0	2	6.66

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	25.00			30.00			33.00			50.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	3	12.49	0	3	5.01	0	4	7.59	0	27	0.00

Table 29.16 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 372)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N	PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	13441	0	0	20.00	26882	0	0
10.7136	14400	0	0	25.00	33602	0	0
13.00	17473	0	0	30.00	40323	0	0
16.00	21505	0	0	33.00	44355	0	0
18.00	24194	0	0	50.00	67205	0	0

29.3.2 Operation in Smart Card Interface Mode

The SCI supports the smart card (IC card) interface conforming to the ISO/IEC 7816-3 (Identification Card) standard, as an extended function of the SCI.

Smart card interface mode can be selected using the appropriate register.

29.3.2.1 Sample Connection

Figure 29.24 shows a sample connection between a smart card (IC card) and this LSI.

As in the figure, since this LSI communicates with an IC card using a single transmission line, interconnect the TxDn and RxDn pins and pull up the data transmission line to Vcc using a resistor.

Setting the TE and RE bits in SCR to 1 with an IC card disconnected enables closed transmission/reception allowing self diagnosis.

To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card. The output port of the RX62N/RX621 can be used to output a reset signal.

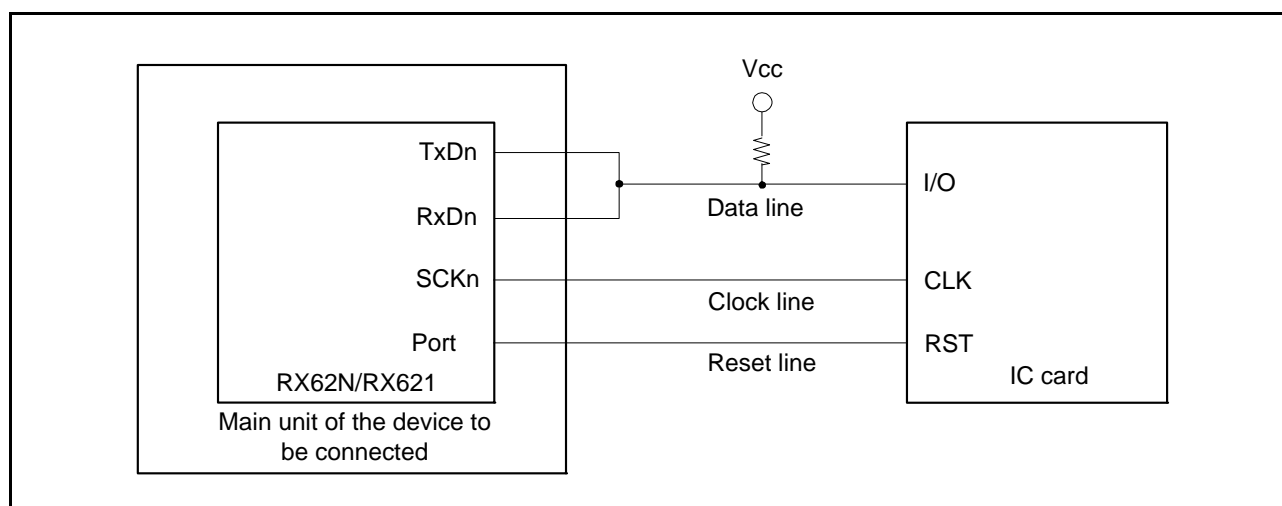


Figure 29.24 Sample Connection with a Smart Card (IC Card)

29.3.3 Data Format (Except in Block Transfer Mode)

Figure 29.25 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring one bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically re-transmitted after at least 2 etu.

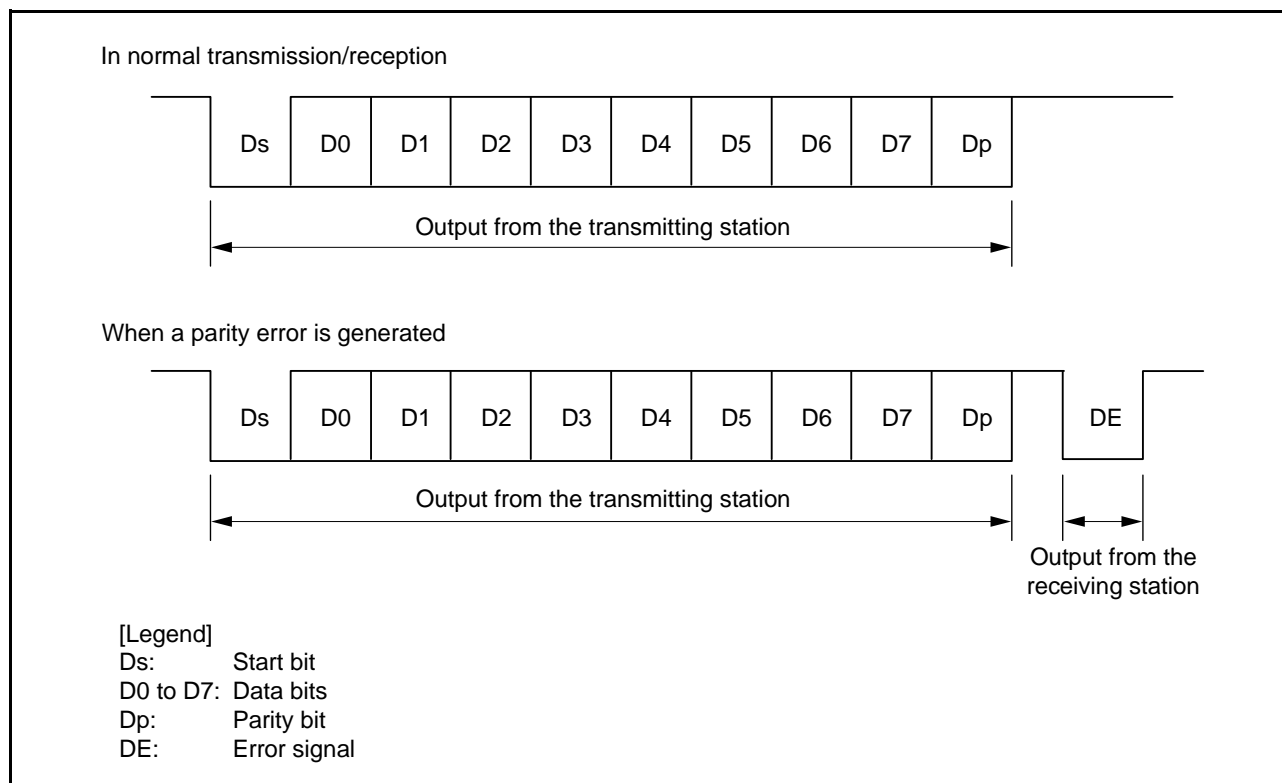


Figure 29.25 Data Formats in Smart Card Interface Mode

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 29.26. 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.

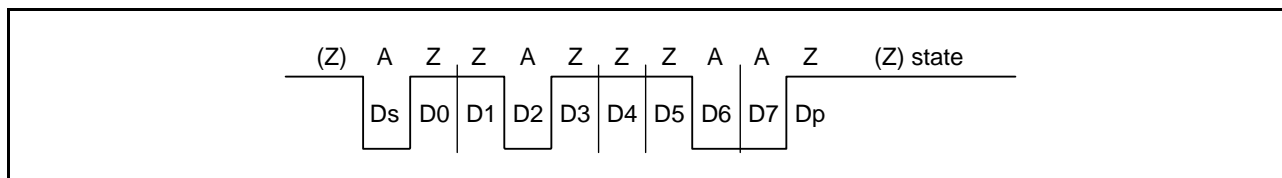


Figure 29.26 Direct Convention (SDIR in SCMR = 0, SINV in SCMR =0, PM in SMR = 0)

(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 29.27. 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 SNIV bit of the RX62N/RX621 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.

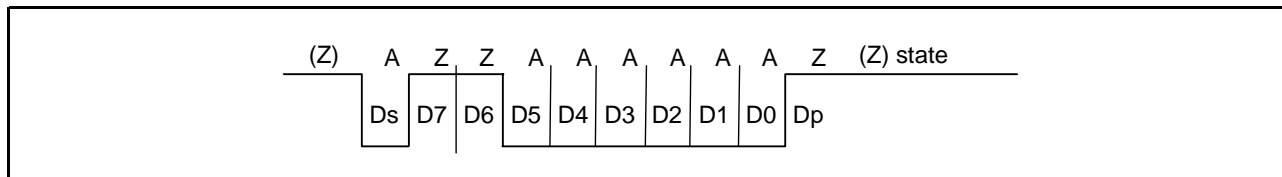


Figure 29.27 Inverse Convention (SDIR in SCMR = 1, SINV in SCMR =1, PM in SMR = 1)

29.3.3.1 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the PER bit in SSR is set by error detection, clear the PER bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag in SSR is set 11.5 etu after transmission start.
- In block transfer mode, the ERS flag in SSR indicates the error signal status as in normal smart card interface mode, but the flag is always read as 0 because no error signal is transferred.

29.3.3.2 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. 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 29.28. The reception margin here is determined by the following formula.

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1+F) \right| \times 100 \text{ [%]}$$

[Legend]

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.

$$M = \left\{ 0.5 - \frac{1}{2 \times 372} \right\} \times 100 \text{ (%) = 49.866\%}$$

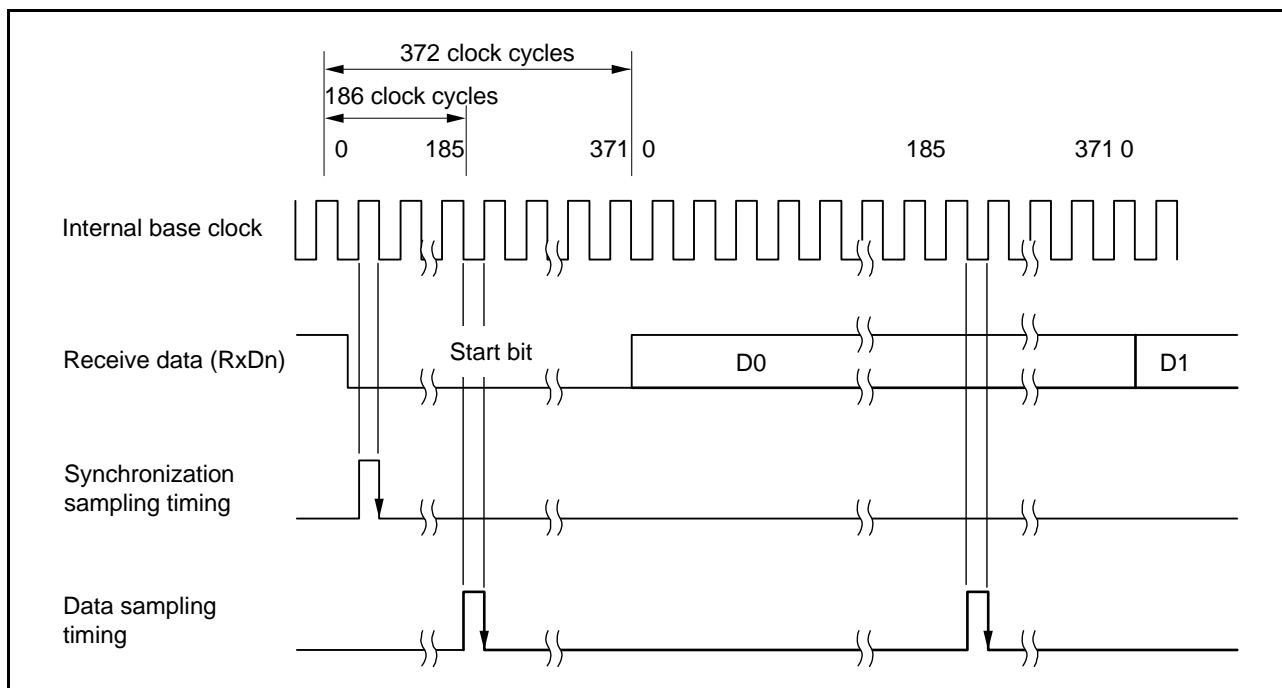


Figure 29.28 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

29.3.3.3 Initialization of the Smart Card Interface

Before transmitting and receiving data, initialize the SCI using the following procedure. Initialization is also necessary before switching from transmission to reception and vice versa.

1. Write the initial value "00h" to the SCR.
2. Set the B_j bit in ICR of PORT_n (n = 0 to 9 and A to G, j = 0 to 7) of the corresponding pin to 1.
3. Set the error flags ORER, ERS, and PER in SSR to 0.
4. Set bits GM, BLK, OE, BCP[1:0], and CKS[1:0] in SMR and the BCP2 bit in SCMR appropriately. Also set the PE bit in SMR to 1.
5. Set bits SDIR, SINV, and SMIF in SCMR appropriately. Also set the Bi bit in DDR of PORT_n corresponding to the Tx_{Dn} pin to 0. Then the Tx_{Dn} and Rx_{Dn} pins are changed from port pins to SCI pins and placed in the high impedance state.
6. Set the value corresponding to the bit rate in BRR.
7. Set the CKE[1:0] bits in SCR appropriately, and set bits TIE, RIE, TE, RE, and TEIE in SCR to 0 at the same time. When the CKE0 bit is set to 1, the SCK_n pin is allowed to output clock pulses.
8. Wait for at least a 1-bit interval, and then set the TIE, RIE, TE, and RE bits in SCR to 1. Setting the TE and RE bits to 1 simultaneously is prohibited except for self diagnosis.

To change reception mode to transmission mode, first check that reception has completed, and then initialize the SCI. At the end of initialization, set TE = 1 and RE = 0. Reception completion can be verified by reading the RXI request, ORER, or PER flag in SSR.

To change transmission mode to reception mode, first check that transmission has completed, and then initialize the SCI. At the end of initialization, set TE = 0 and RE = 1. Transmission completion can be verified by reading the TEND flag in SSR.

29.3.3.4 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode) is different from that in normal serial communications interface mode in that an error signal is sampled and data can be re-transmitted. Figure 29.29 shows the data re-transfer operation during transmission.

1. When an error signal from the receiving 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 re-transferred from TDR to TSR allowing automatic data retransmission.
3. If no error signal is returned from the receiving end, 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 29.31 shows a sample flowchart of serial transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC or DMACA.

When the TEND flag in SSR is set to 1 in transmission, if the TIE bit in SCR is 1, a TXI interrupt request is generated. The DTC or DMACA is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC or DMACA activation beforehand, allowing transfer of transmit data. The TEND flag is automatically cleared to 0 when the DTC or DMACA transfers the data.

If an error occurs, the SCI automatically re-transmits the same data. During this retransmission, the TEND flag is kept to 0 and the DTC or DMACA is not activated. Therefore, the SCI and DTC or DMACA automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared; set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag to 0.

When transmitting/receiving data using the DTC or DMACA, be sure to make settings to enable the DTC or DMACA before making SCI settings.

For DTC or DMACA settings, see section 14, DMA Controller (DMACA) and section 16, Data Transfer Controller (DTCa).

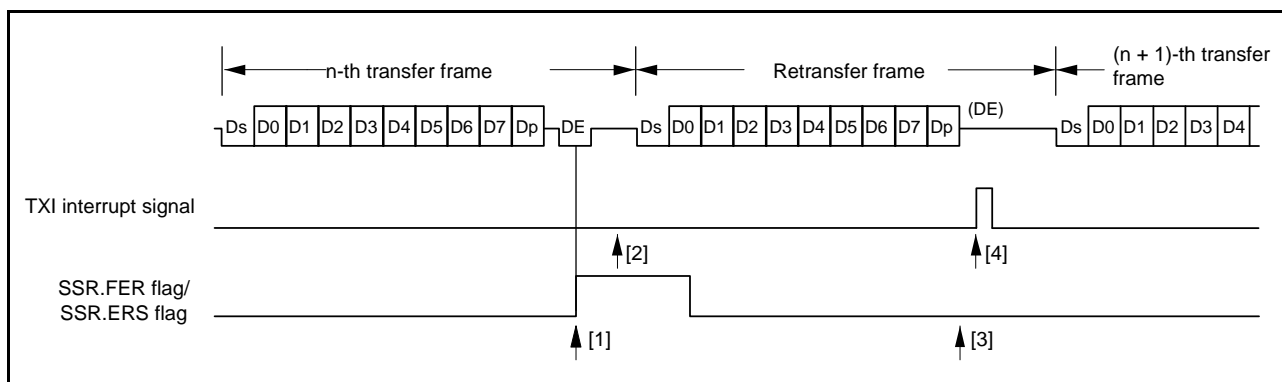


Figure 29.29 Data Retransfer Operation in SMCI Transmission Mode

Note that the SSR.TEND flag is set in different timings depending on the GM bit setting in SMR. Figure 29.30 shows the TEND flag generation timing.

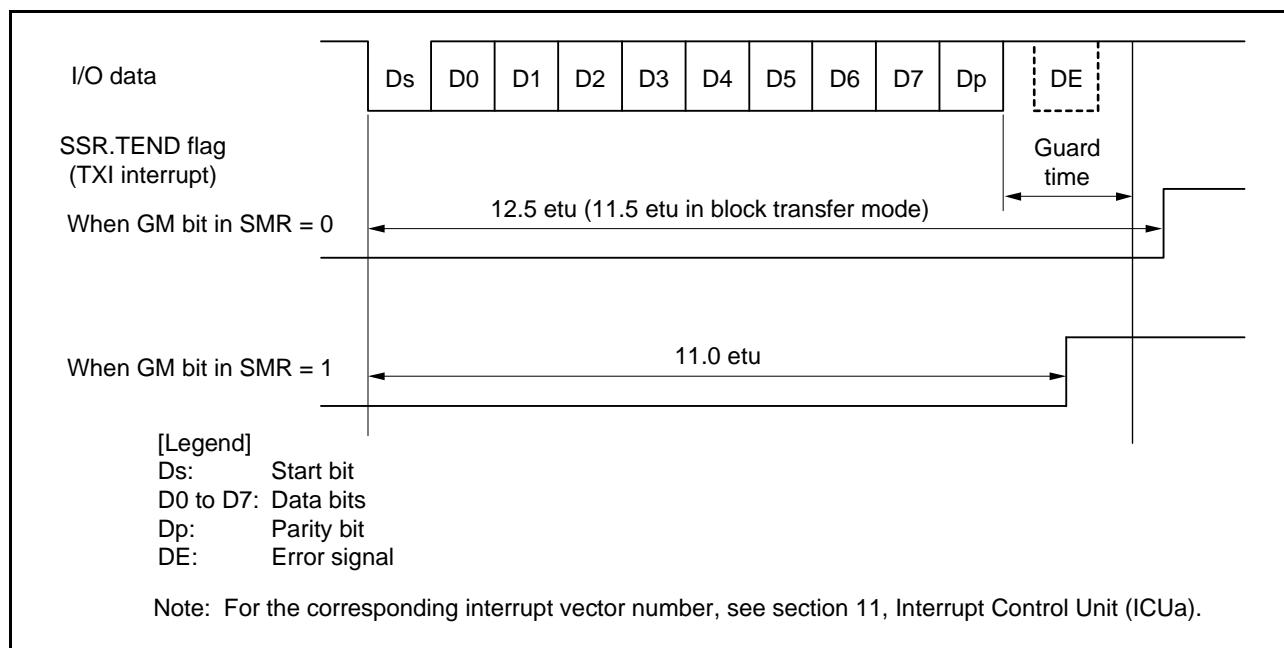


Figure 29.30 SSR.TEND Flag Generation Timing during Transmission

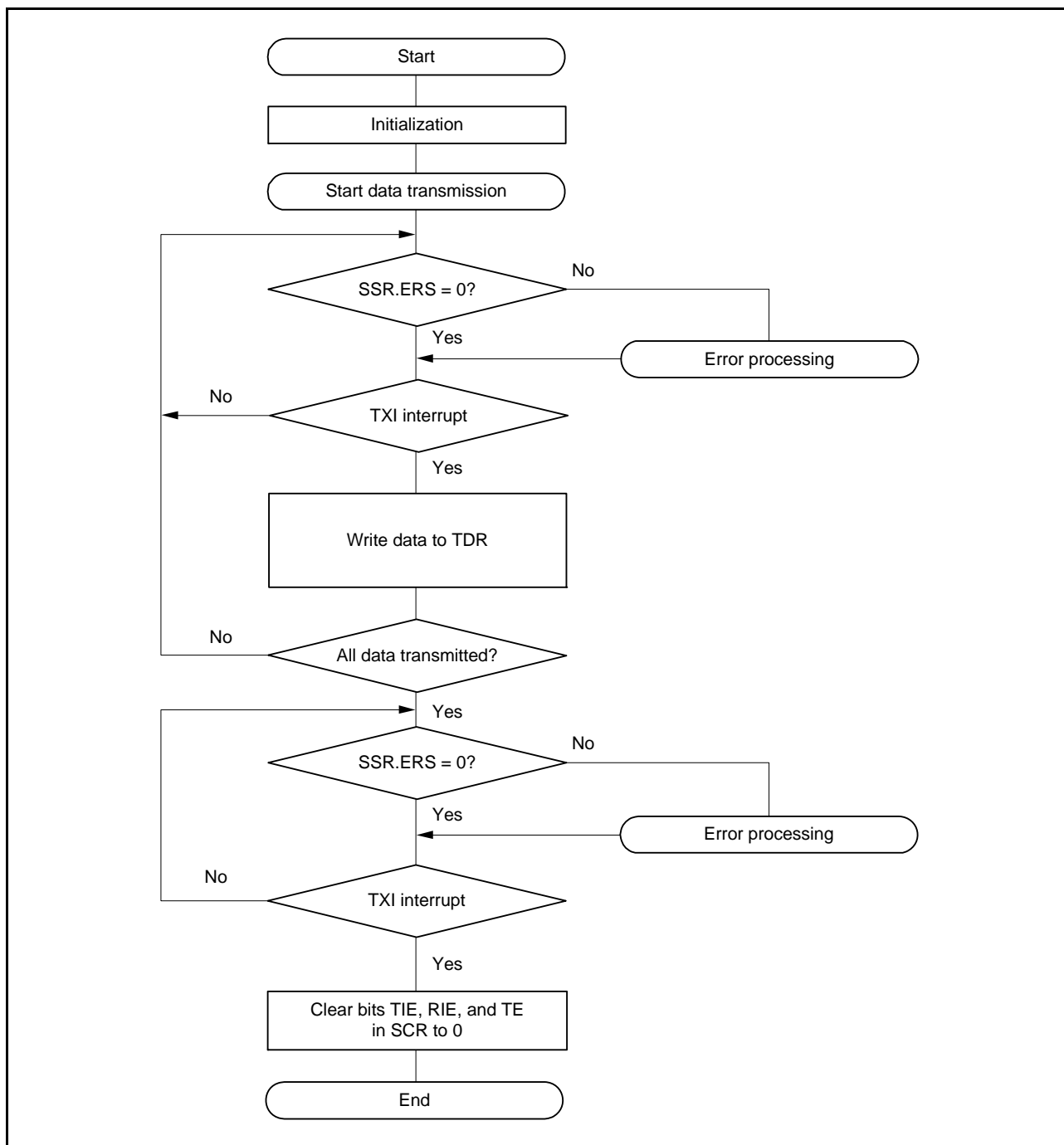


Figure 29.31 Sample Serial Transmission Flowchart

29.3.3.5 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in serial communications interface mode. Figure 29.32 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 29.33 shows a sample flowchart for serial data reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DTC or DMACA.

In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DTC or DMACA is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DTC or DMACA 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. If an error occurs, the DTC or DMACA is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMACA 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: For operations in block transfer mode, see section 29.2.2, Operation in Asynchronous Mode.

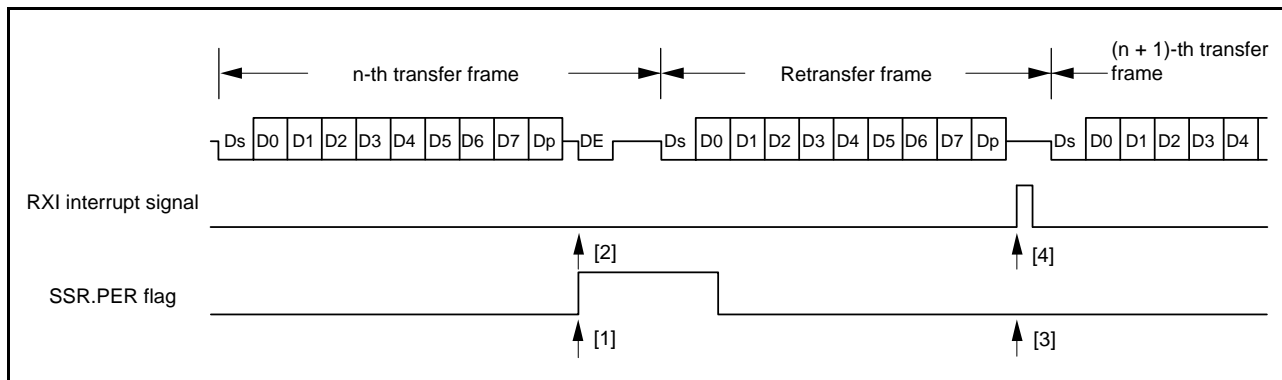


Figure 29.32 Data Retransfer Operation in SMCI Reception Mode

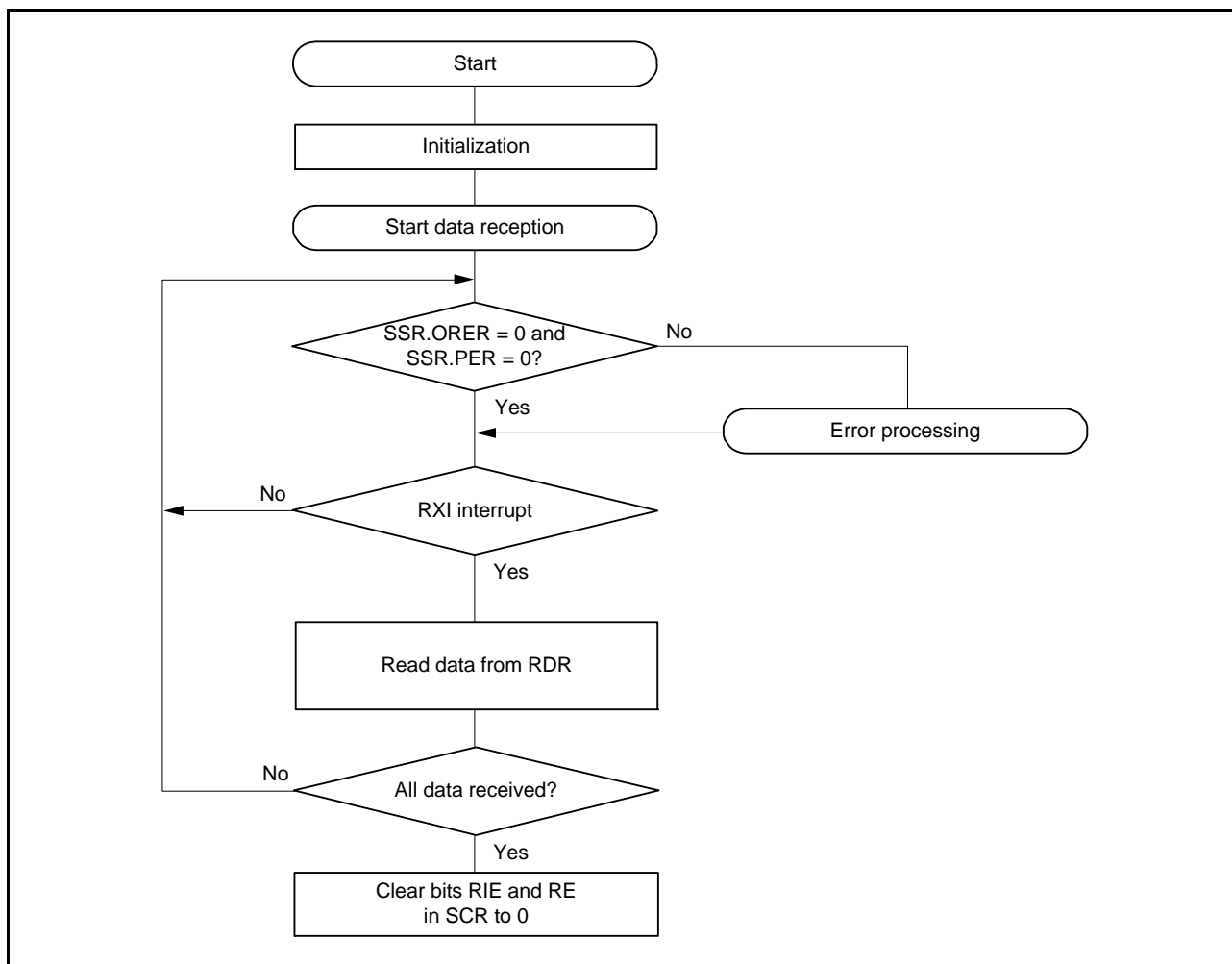


Figure 29.33 Sample Serial Reception Flowchart

29.3.3.6 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 29.34 shows an example of clock output fixing timing when the CKE0 bit is controlled with GM = 1 and CKE1 = 0.

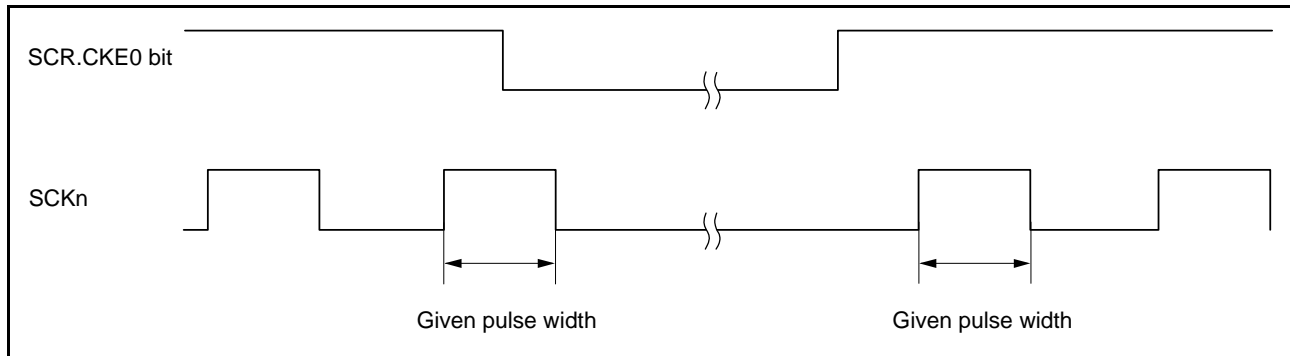


Figure 29.34 Clock Output Fixing Timing

At power-on and transitions to/from software standby mode, use the following procedure to secure the appropriate clock duty cycle.

(1) At Power-On

To secure the appropriate clock duty cycle simultaneously with power-on, use the following procedure.

1. Initially, port input is enabled in the high-impedance state. To fix the potential level, use a pull-up or pull-down resistor.
2. Fix the SCKn pin to the specified output using the CKE1 bit in SCR.
3. Set SMR and SCMR to enable smart card interface mode. Set the CKE0 bit in SCR to 1 to start clock output.

(2) At Mode Switching

(a) At transition from smart card interface mode to software standby mode

1. Set the data register (DR of PORTn) and data direction register (DDR of PORTn) corresponding to the SCKn pin to the values for the output fixed state in software standby mode.
2. Write 0 to the TE and RE bits in SCR to stop transmission/reception.
Simultaneously, set the CKE1 bit in SCR to the value for the output fixed state in software standby mode.
3. Write 0 to the CKE0 bit in SCR to stop the clock.
4. Wait for one cycle of the serial clock. In the mean time, the clock output stops with the low level retained after outputting the specified high width.
5. Make a transition to software standby mode.

(b) At transition from software standby mode to smart card interface mode

1. Cancel software standby mode.
2. Set the SCR.CKE bit to 1. The clock output with the specified clock frequency is then restarted.

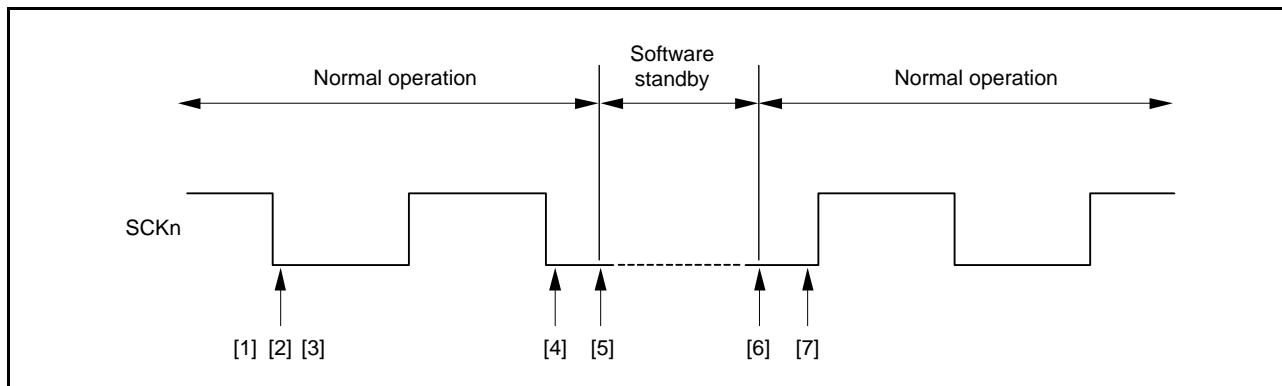


Figure 29.35 Clock Stop and Restart Procedure

29.4 Interrupt Sources

29.4.1 Interrupts in Serial Communications Interface Mode

Table 29.17 lists interrupt sources in normal serial communications interface 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. Transfer of data from the transmit data register (TDR) to TSR while the TIE bit in SCR is 1 leads to the generation of a TXI interrupt request. Moreover, setting the TIE bit in SCR to 1 and then setting the TE bit to 1, or setting the TIE and TE bits in SCR to 1 simultaneously using one instruction, leads to the generation of a TXI interrupt request. A TXI interrupt request can activate the DTC or DMACA for data transfer.

Setting of received data in RDR while the RIE bit in SCR is 1 leads to the generation of an RXI interrupt request. An RXI interrupt can activate the DTC or DMACA for data transfer.

Setting the ORER, FER, or PER flag in SSR to 1 while the RIE bit in SCR is 1 leads to generation of an ERI interrupt request. Here, an RXI interrupt request is not generated.

If the TDR has not been updated by the time of transmission of the tail-end bit of data being transmitted, the TEND flag in SSR is set to 1 and, if the value of the TEIE bit in SCR is 1, a TEI interrupt request is generated. Writing of data to the TDR during TXI interrupt processing leads to clearing of the TEND flag in SSR and clearing of the TEI interrupt at its source. When clearing the TEND flag in SSR by writing transmit data to TDR, read the TEND flag in SSR to confirm that it is 0.

The TXI interrupt request is generated by setting the TIE bit in SCR to 1 and then setting the TE bit to 1 or by setting the TIE and TE bits in SCR to 1 simultaneously using one instruction. The TXI interrupt is not generated if the TE bit is set while the TIE bit is 0 or the TIE bit is set to 1 after the TE bit is set to 1. Therefore, to disable the TXI interrupt, perform the transmit end interrupt processing, and then start data transfer again, for example, in the final data transmission, the TXI interrupt should be enabled/disabled using the corresponding ICU.IERm.IENj bit.

Table 29.17 SCI Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMACA Activation	Priority
ERI	Receive error	ORER, FER, or PER	Not possible	Not possible	High
RXI	Receive data full	—	Possible	Possible	↑
TXI	Transmit data empty	—	Possible	Possible	
TEI	Transmit end	TEND	Not possible	Not possible	Low

29.4.2 Interrupts in Smart Card Interface Mode

Table 29.18 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 29.18 SCI Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMACA Activation	Priority
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible	High
RXI	Receive data full	—	Possible	Possible	↑
TXI	Transmit data empty	TEND	Possible	Possible	Low

Data transmission/reception using the DTC or DMACA is also possible in smart card interface mode, similar to in the normal SCI mode. In transmission, when the TEND flag in SSR is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC or DMACA allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC or DMACA activation. The TEND flag is automatically cleared to 0 when the DTC or DMACA transfers the data.

If an error occurs, the SCI automatically re-transmits the same data. During the retransmission, the TEND flag is kept to 0 and the DTC or DMACA is not activated. Therefore, the SCI and DTC or DMACA automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the ERS flag in SSR is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the RIE bit in SCR to 1 to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC or DMACA, be sure to make settings to enable the DTC or DMACA before making SCI settings. For DTC or DMACA settings, see section 14, DMA Controller (DMACA) and 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 or DMACA allowing transfer of receive data if the RXI request is specified beforehand as a source of DTC or DMACA activation. If an error occurs, the error flag is set. Therefore, the DTC or DMACA is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

29.5 Usage Notes

29.5.1 Setting the Module Stop Function

Operation of the SCI can be disabled or enabled using the module stop control register B (MSTPCRB). The initial setting is for operation of the SCI to be halted. Register access is enabled by clearing the module stop state. For details, see section 9, Low Power Consumption.

29.5.2 Break Detection and Processing

When a framing error is detected, a break can be detected by reading the RxDn pin value directly. In a break, the input from the RxDn pin becomes all 0s, and so the FER flag in SSR is set to 1 (framing error), and the PER flag in SSR may also be set to 1 (parity error). The SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is cleared to 0 (no framing error), it will be set to 1 again.

29.5.3 Mark State and Break Detection

When the TE bit in SCR is 0 (serial transmission disabled), the TxDn pin is used as an I/O port whose direction (input or output) and level are determined by the Bj bit in DR of PORTn and Bj bit in DDR of PORTn. This can be used to set the TxDn pin to mark state or send a break during serial data transmission. To maintain the communications line in mark state (the state of 1) until the TE bit is set to 1 (to enable serial transmission), set both Bj bit in DR of PORTn and Bj bit in DDR of PORTn to 1. Since the TE bit is cleared to 0 at this time, the TxDn pin becomes an I/O port, and 1 is output from the TxDn pin. To send a break during serial data transmission, first set Bj bit in DDR of PORTn = 1 and Bj bit in DR of PORTn = 0, and then clear the TE bit to 0. When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TxDn pin becomes an I/O port, and a 0 is output from the TxDn pin.

29.5.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only)

Transmission cannot be started when a receive error flag (ORER, FER, or RER) in SSR is set to 1, even if data is written to TDR. Be sure to clear the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be cleared to 0 even if the RE bit in SCR is cleared to 0 (serial reception disabled).

29.5.5 Writing Data to TDR

Data can always be written to TDR. However, if new data is written to TDR when transmit data is remaining in TDR, the previous data in TDR is lost because it has not been transferred to TSR yet. Be sure to write transmit data to TDR in the TXI interrupt request processing routine.

29.5.6 Restrictions on Clock Synchronous Transmission

When the external clock source is used as a synchronization clock, update TDR by the DMACA or DTC and wait for at least five clock cycles before allowing the transmit clock to be input. If the transmit clock is input within four clock cycles after TDR is updated, the SCI may malfunction.

29.5.7 Restrictions on Using DTC or DMACA

When using the DMACA or DTC to read RDR, be sure to set the receive end interrupt (RXI) as the activation source of the relevant SCI.

29.5.8 SCI Operations during Low Power Consumption State

(1) Transmission

Before specifying the module stop state or making a transition to software standby mode, stop the transmit operations ($TIE = TE = TEIE = 0$ in SCR). TSR, TDR, and SSR are reset. Setting the TE bit to 0 modifies the TSR register and the SSR.TEND flag to 0. The states of the output pins in the module stop state or in software standby mode depend on the port settings, and the output pins are held high after cancellation. If the transition is made during data transmission, the data being transmitted will be undefined.

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 29.36 shows a sample flowchart for transition to software standby mode during transmission. Figure 29.37 and 29.38 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. 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 29.39 shows a sample flowchart for mode transition during reception.

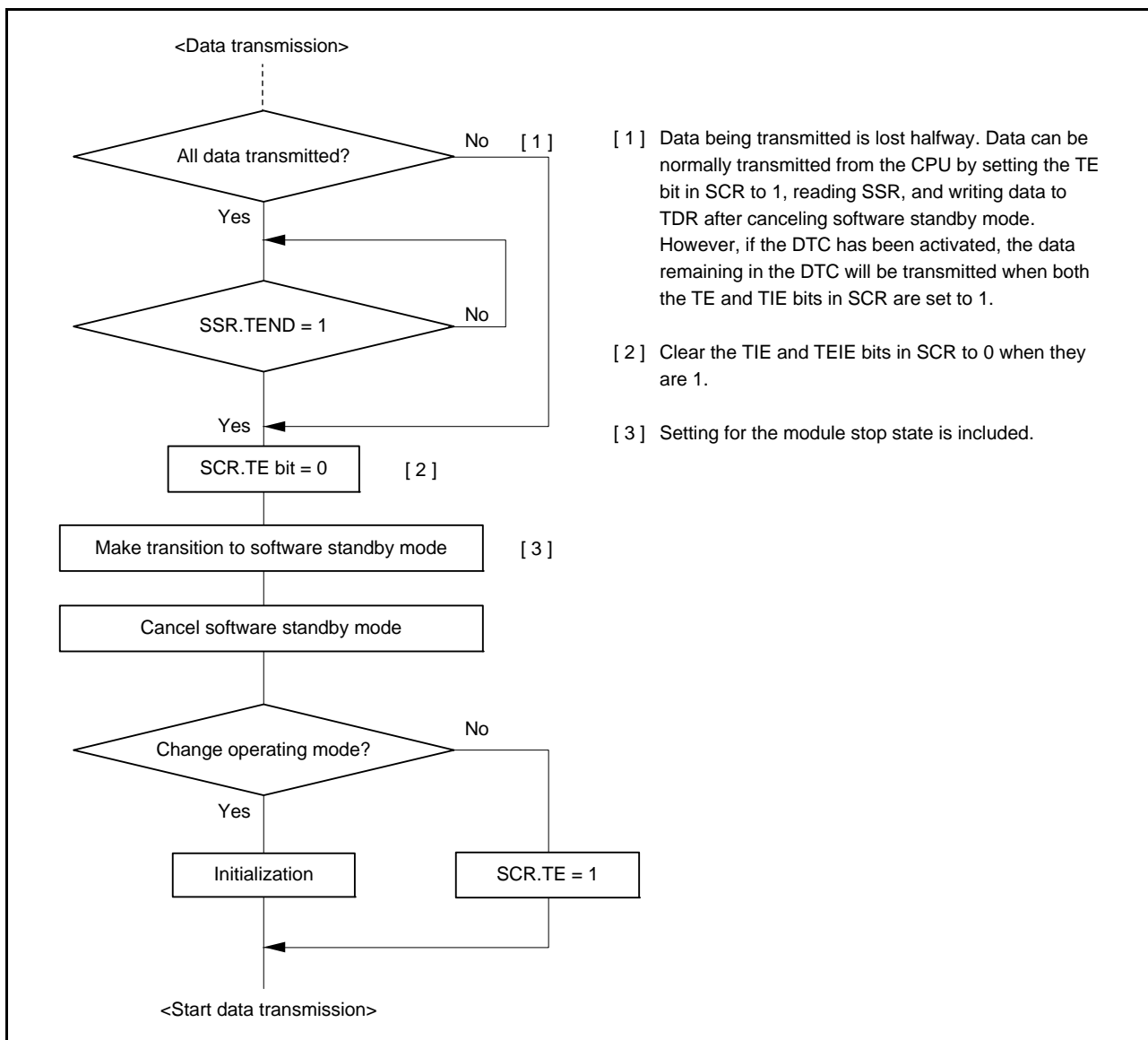


Figure 29.36 Example of Flowchart for Transition to Software Standby Mode during Transmission

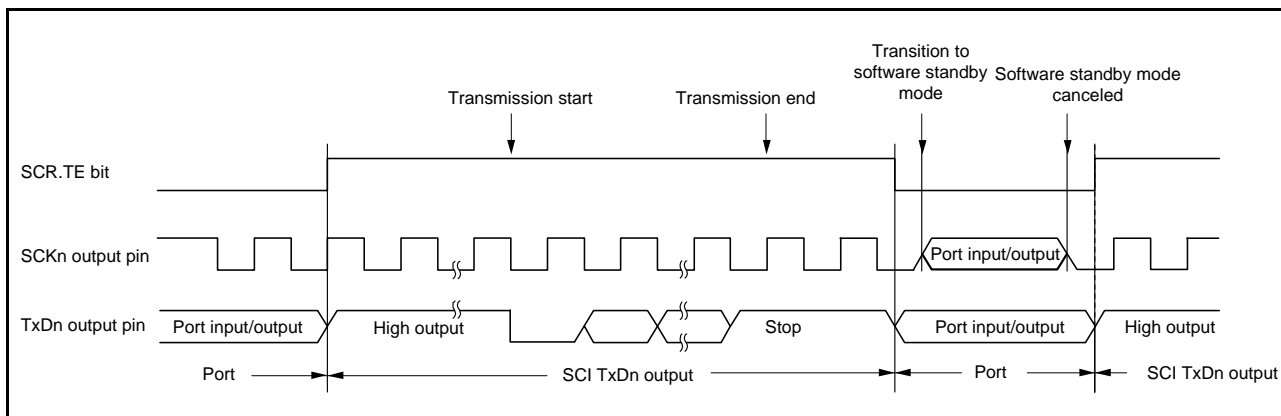


Figure 29.37 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

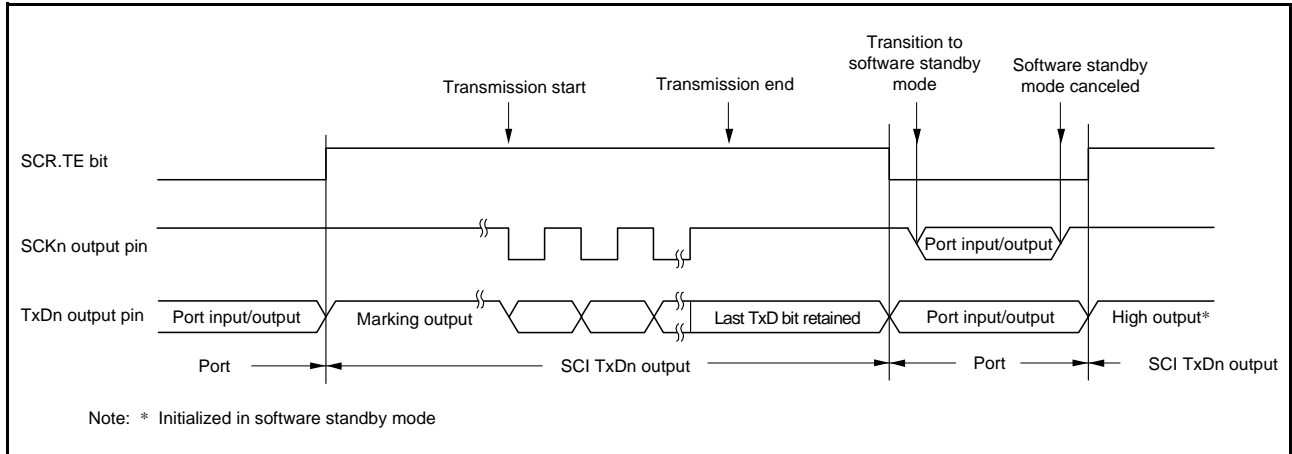


Figure 29.38 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

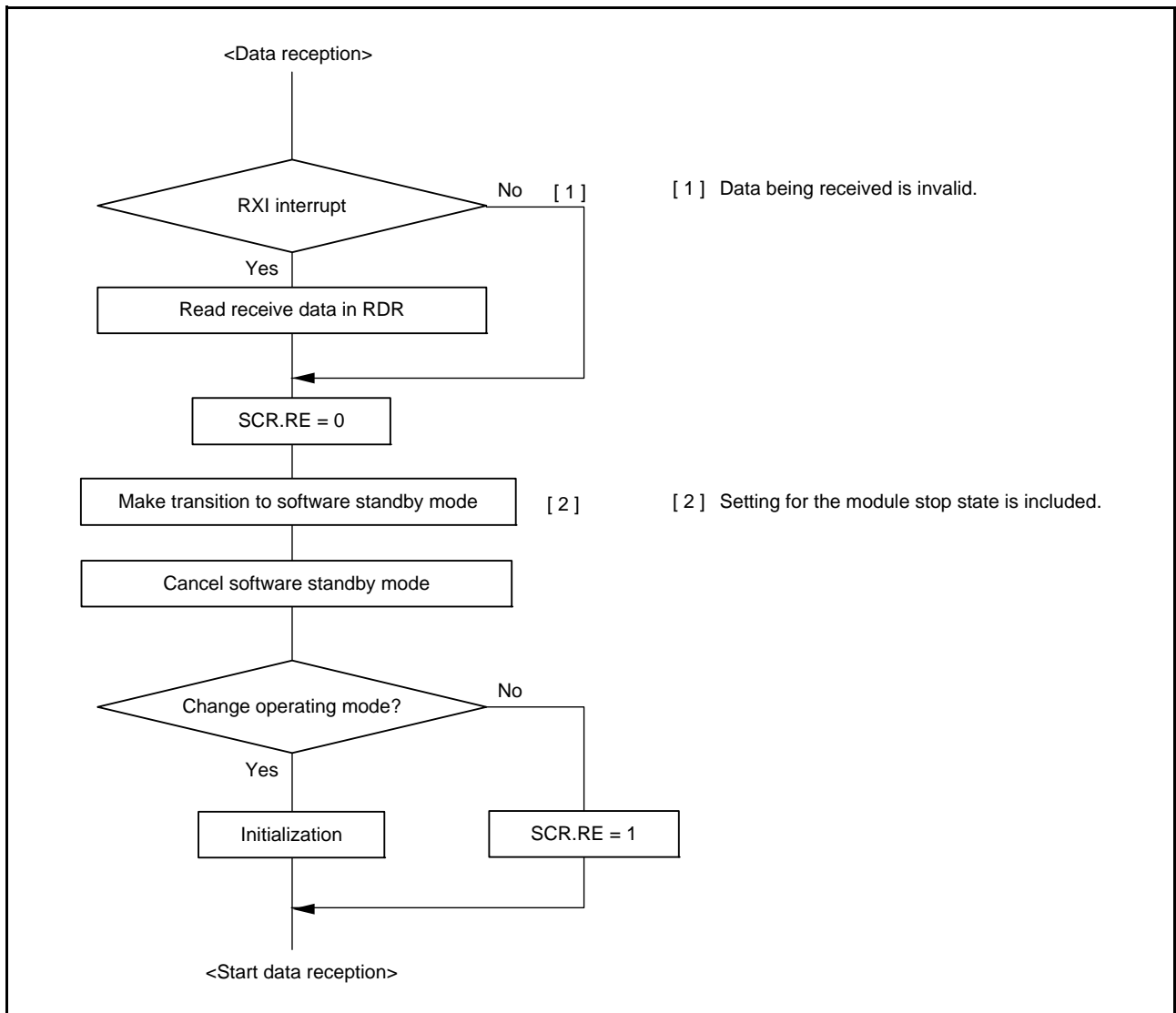


Figure 29.39 Example of Flowchart for Transition to Software Standby Mode during Reception

29.5.9 External Clock Input in Clock Synchronous Mode

In clock synchronous mode, the external clock SCKn must be input as follows:

High-pulse period, low-pulse period = 2 clock cycles or more, period = 6 clock cycles or more

30. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes of data blocks.

30.1 Overview

Table 30.1 lists the specifications of the CRC calculator, and Figure 30.1 shows a block diagram of the CRC calculator.

Table 30.1 Specifications of CRC

Item	Description
Data for CRC calculation*	CRC code generated for any desired data in 8n-bit units (where n is a whole number)
Data block size	8 bits
CRC processor unit	Operation executed on eight bits in parallel
CRC generating polynomial	One of three generating polynomials selectable <ul style="list-style-type: none"> • 8-bit CRC $X^8 + X^2 + X + 1$ • 16-bit CRC $X^{16} + X^{15} + X^2 + 1$ $X^{16} + X^{12} + X^5 + 1$
CRC calculation switching	CRC code generation for LSB-first or MSB-first communication selectable
Power-down function	Module stop state can be set

Note: * The circuit does not have functionality to divide data for calculation into a data-block size. Write data in 8-bit units.

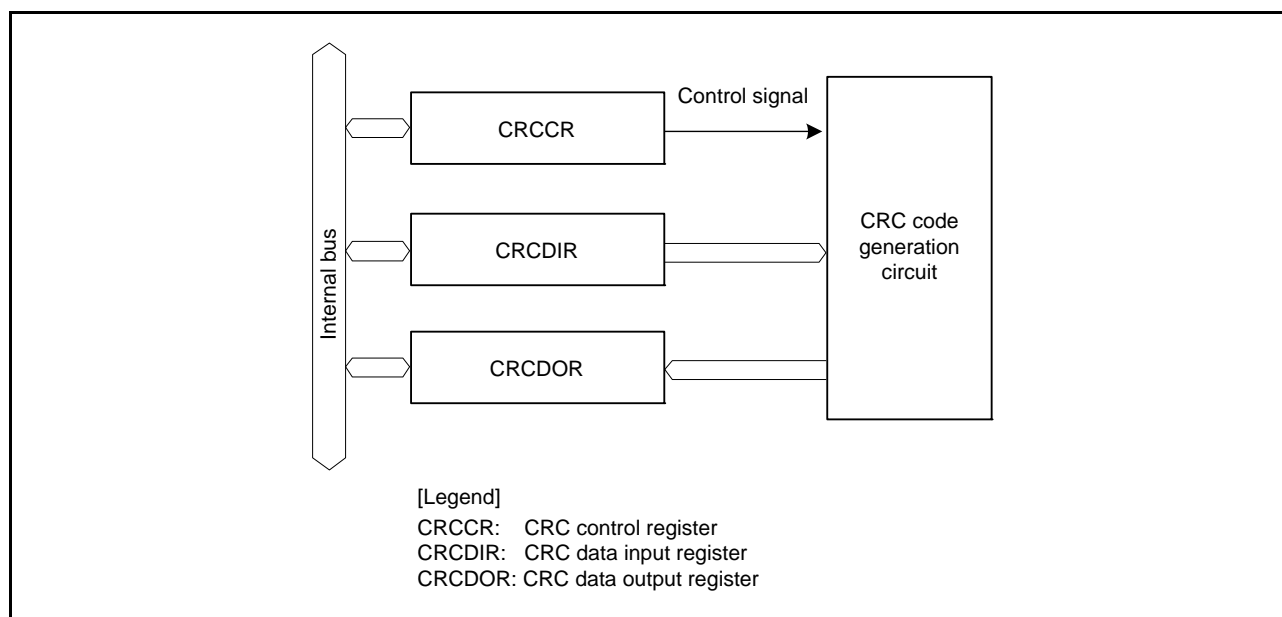


Figure 30.1 Block Diagram of CRC Calculator

30.2 Register Descriptions

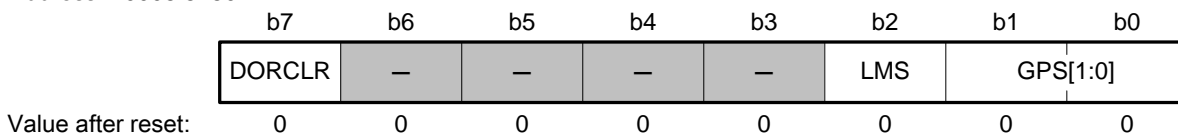
Table 30.2 lists the registers of the CRC calculator.

Table 30.2 Registers of CRC Calculator

Register Name	Symbol	Value after Reset	Address	Access Size
CRC control register	CRCCR	00h	0008 8280h	8
CRC data input register	CRCDIR	00h	0008 8281h	8
CRC data output register	CRCDOR	0000h	0008 8282h	16

30.2.1 CRC Control Register (CRCCR)

Address: 0008 8280h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	GPS[1:0]	CRC Generating Polynomial Switching	b1 b0 0 0: No calculation is executed.* 0 1: $X^8 + X^2 + X + 1$ 1 0: $X^{16} + X^{15} + X^2 + 1$ 1 1: $X^{16} + X^{12} + X^5 + 1$	R/W
b2	LMS	CRC Calculation Switching	0: Performs CRC operation for LSB-first communication. The lower-order byte (bits 7 to 0) is the first to be transmitted when the contents of the CRCDOR (CRC code) are divided into bytes. 1: Performs CRC operation for MSB-first communication. The higher-order byte (bits 15 to 8) is first to be transmitted when the contents of the CRCDOR (CRC code) are divided into bytes.	R/W
b6 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	DORCLR	CRCDOR Register Clear	0: No effect on the operation 1: Clear the CRCDOR register This bit is always read as 0.	R/W

Note : * The CRC data output register (CRCDOR) is always 0000h.

CRCCR initializes the CRC calculator, switches the operation mode, and selects the generating polynomial.

GPS[1:0] Bits (CRC Generating Polynomial Switching)

These bits select the CRC code generating polynomial.

LMS Bit (CRC Calculation Switching)

Selects LSB-first or MSB-first communication for the CRC code generation.

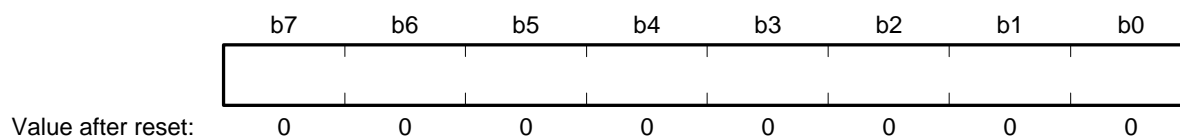
DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is cleared to 0000h.

This bit is always read as 0.

30.2.2 CRC Data Input Register (CRCDIR)

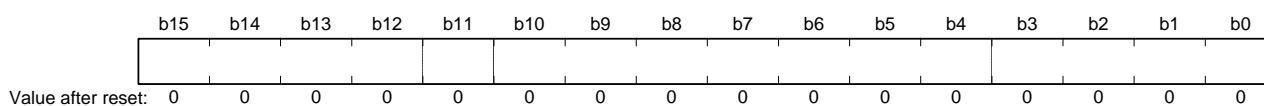
Address: 0008 8281h



CRCDIR is an 8-bit readable/writable register, to which the bytes to be CRC-operated are written.

30.2.3 CRC Data Output Register (CRCDOR)

Address: 0008 8282h



CRCDOR is a 16-bit readable/writable register that contains the result of CRC calculation.

In general, the value will be 0 if there is no CRC error when the calculated CRC code matches the CRC code that continues on, for verification, from the transferred data.

When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in the lower-order byte (b7 to b0). The higher-order byte (b15 to b8) is read as 00h.

30.3 Operation

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer.

The following figures show examples in which the CRCCR.GPS[1:0] bits are set to 11b so the CRC code is calculated by using a 16-bit CRC (with the polynomial $X^{16} + X^{12} + X^5 + 1$), and the CRC code is calculated for the value "F0h".

When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in the lower-order byte of CRCDOR.

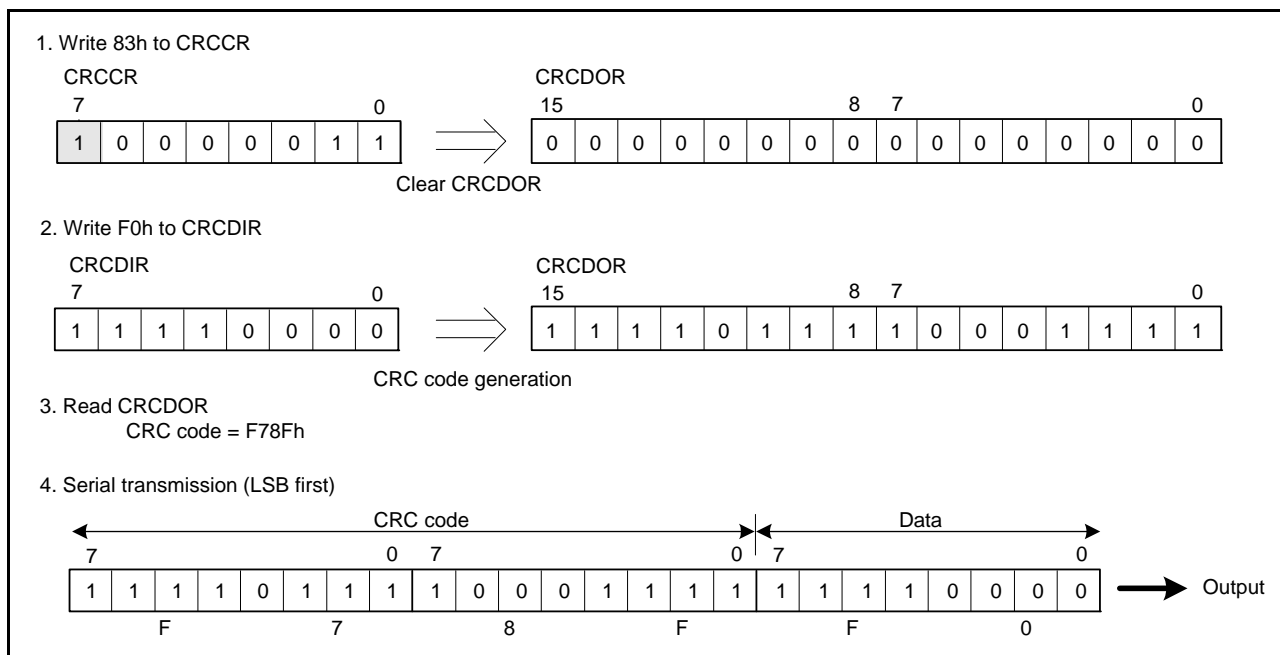


Figure 30.2 LSB-First Data Transmission

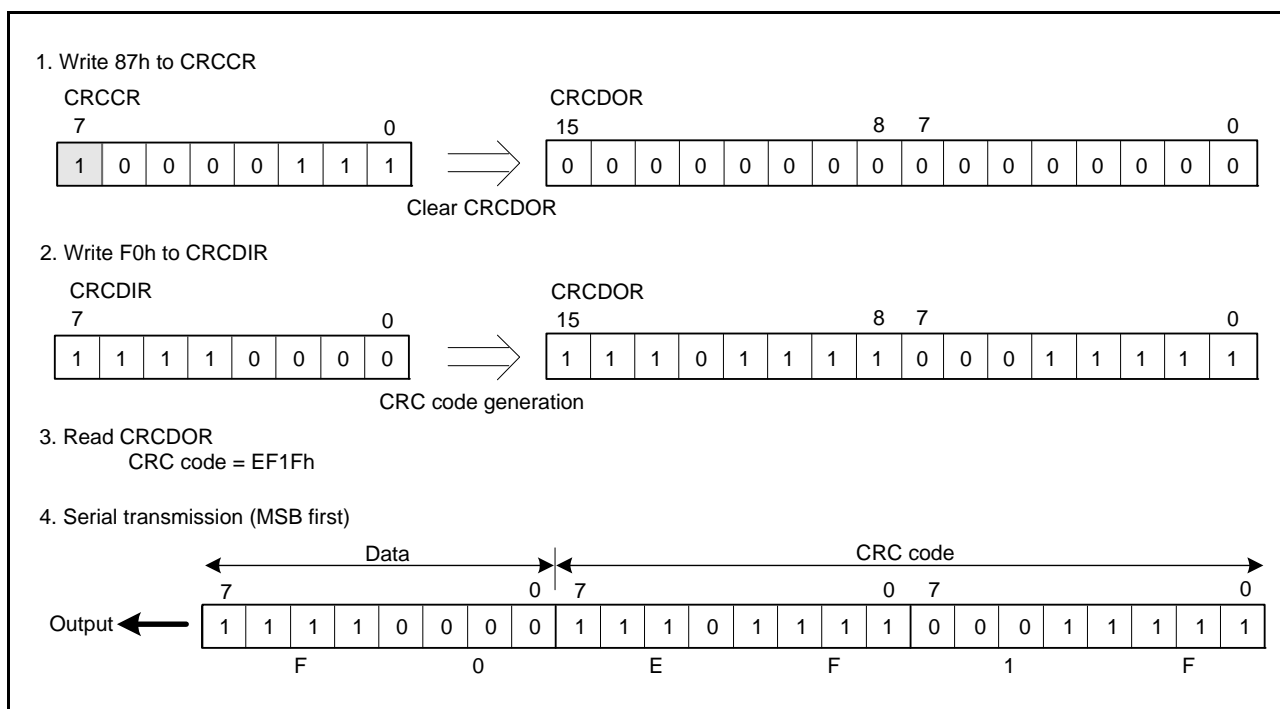


Figure 30.3 MSB-First Data Transmission

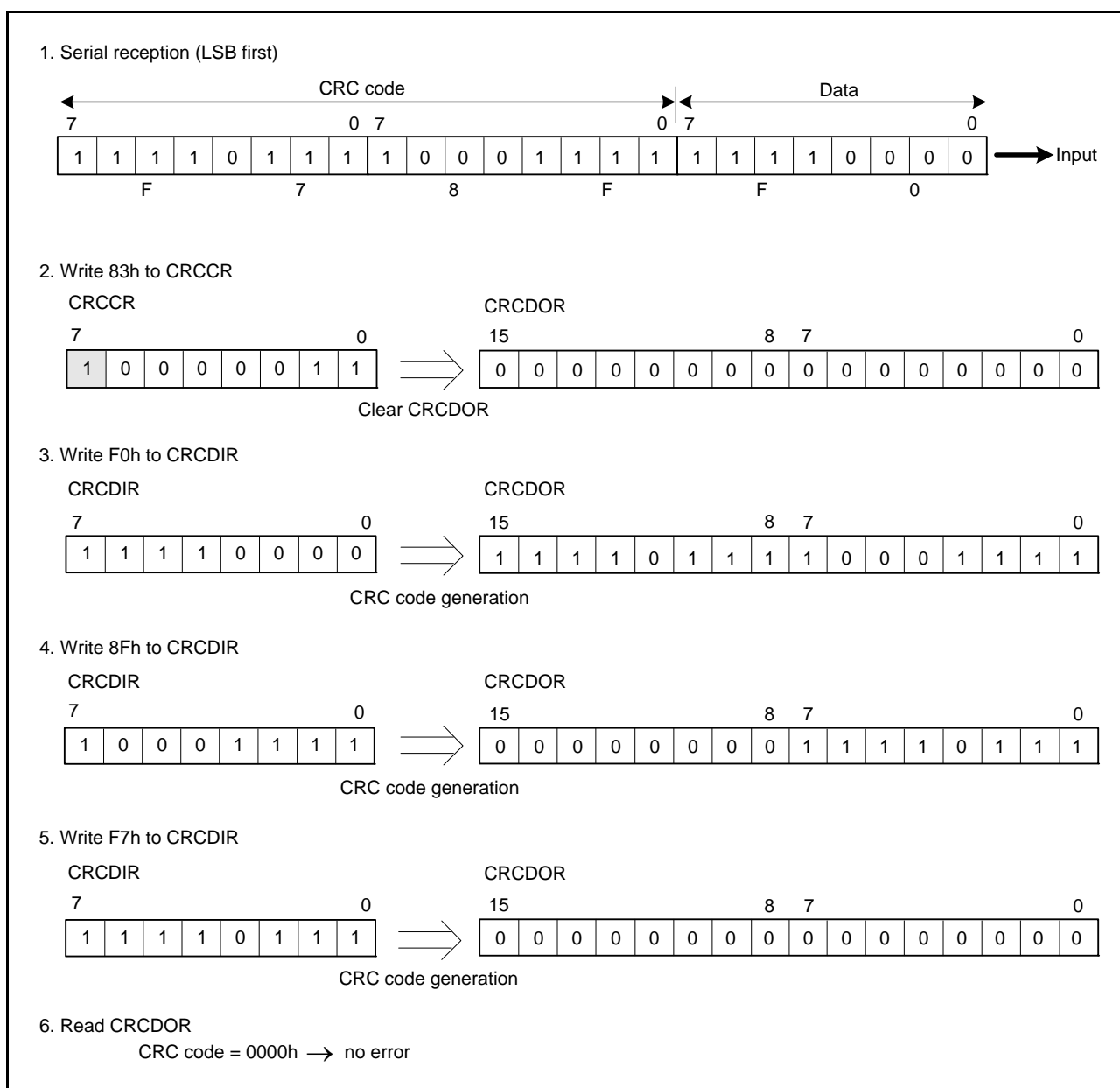


Figure 30.4 LSB-First Data Reception

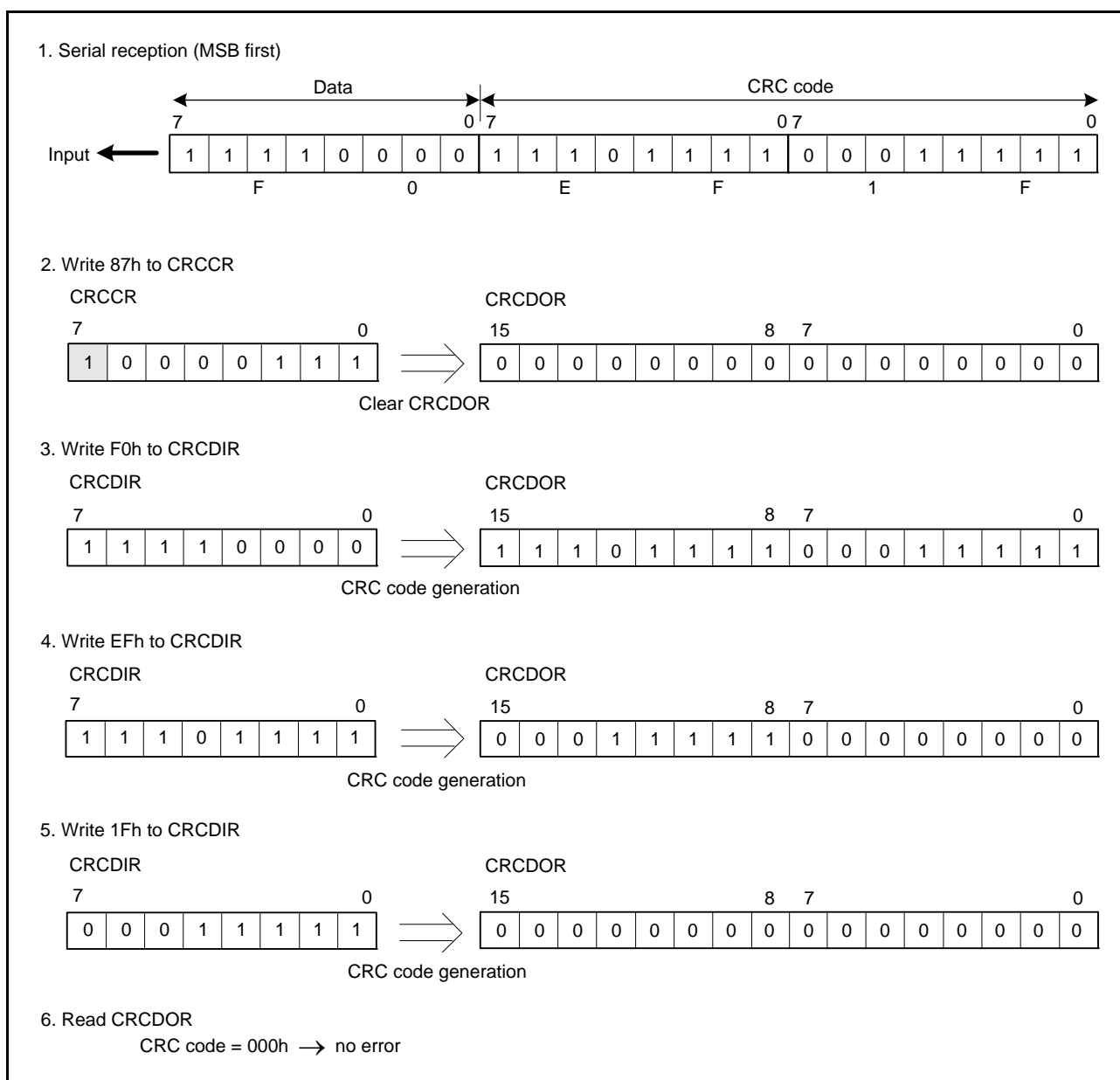


Figure 30.5 MSB-First Data Reception

30.4 Usage Notes

30.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using the module stop control register B (MSTPCRB). The initial setting is for operation of the CRC calculator to be halted. Register access is enabled by clearing the module stop state. For details, see section 9, Low Power Consumption.

30.5 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB-first or MSB-first.

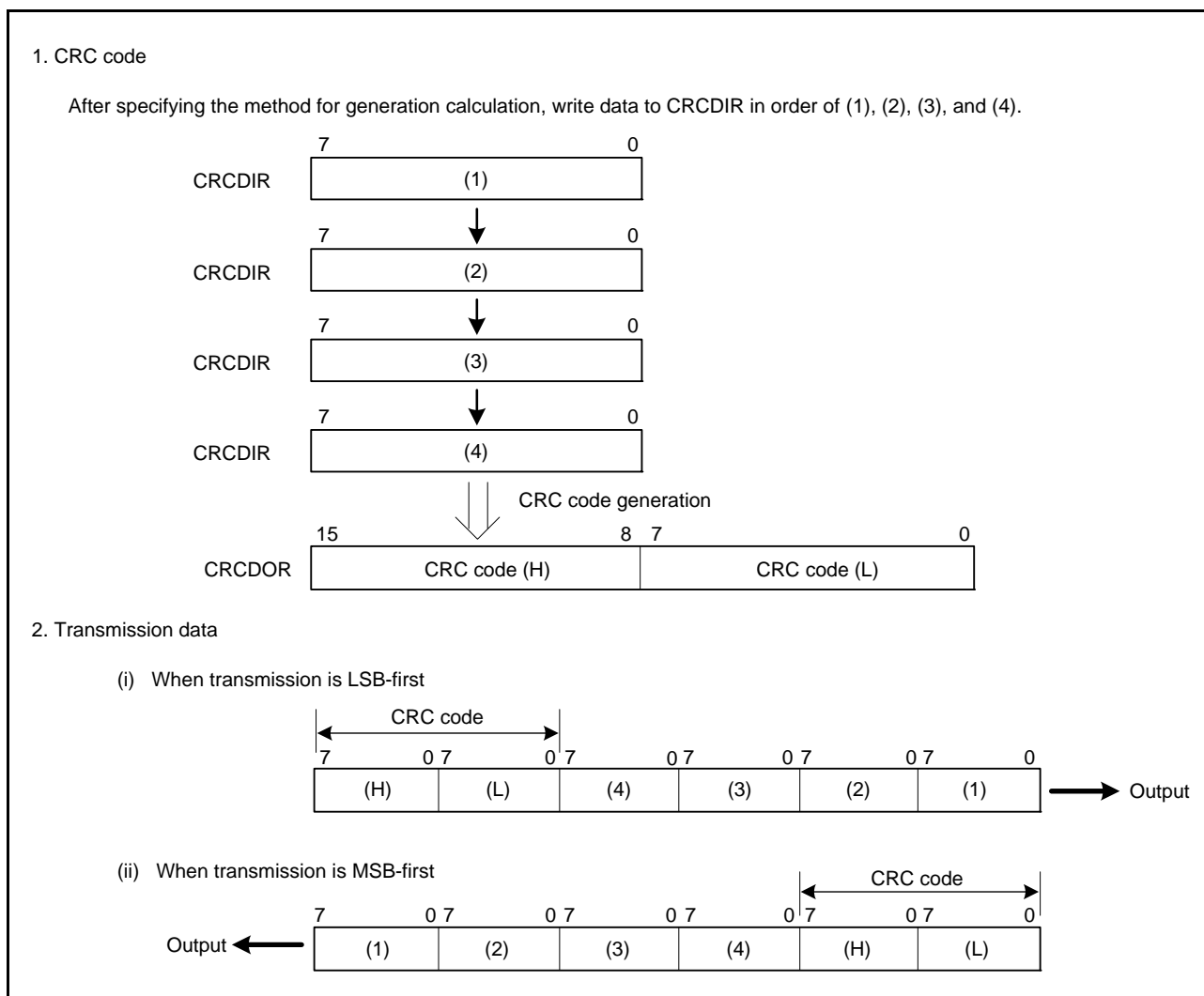


Figure 30.6 LSB-First and MSB-First Data Transmission

31. I²C Bus Interface (RIIC)

The RX62N/RX621 Group has two I²C bus interfaces (RIIC modules).

The RIIC module conforms with and provides a subset of the NXP I²C bus (inter-IC bus) interface functions.

31.1 Overview

Table 31.1 lists the specifications of the RIIC, Figure 31.1 shows a block diagram of the RIIC, and Figure 31.2 shows an example of I/O pin connections to external circuits (I²C bus configuration example). Table 31.2 shows the input/output pins of the RIIC.

Table 31.1 RIIC Specifications

Item	Specifications
Communications format	<ul style="list-style-type: none"> I²C bus format or SMBus format Master mode or slave mode selectable Automatic securing of the various set-up times, hold times, and bus-free times for the transfer rate
Transfer rate	Up to 1M bps
SCL clock	<ul style="list-style-type: none"> For master operation, the duty cycle of the SCL clock is selectable in the range from 4% to 96%.
Issuing and detecting conditions	<ul style="list-style-type: none"> 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"> Up to three slave-address settings can be made. Seven- and ten-bit address formats are supported (along with the use of both at once). General call addresses, device ID addresses, and SMBus host addresses are detectable.
Acknowledgement	<ul style="list-style-type: none"> 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. 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 (i.e. the return of ACK or NACK is selectable).
Wait function	<ul style="list-style-type: none"> In reception, the following periods of waiting can be obtained by holding the clock signal (SCL) at the low level: <ul style="list-style-type: none"> waiting between the eighth and ninth clock cycles (timing of the received data full interrupt can be selected for this); and waiting between the ninth clock cycle and the first clock cycle of the next transfer (WAIT function)
SDA output delay function	<ul style="list-style-type: none"> Timing of the output of transmitted data, including the not-acknowledge bit, can be delayed.
Arbitration	<ul style="list-style-type: none"> For multi-master operation <ul style="list-style-type: none"> Operation to synchronize the SCL (clock) signal in cases of conflict with the SCL signal from another master is possible. 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. 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. 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). 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. Loss of arbitration due to non-matching of internal and line levels for data is detectable in slave transmission.
Timeout detection function	<ul style="list-style-type: none"> The internal timeout detection function is capable of detecting long-interval stoppages of the SCL (clock signal).
Noise removal	<ul style="list-style-type: none"> The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable.
Interrupt sources	<ul style="list-style-type: none"> Four sources: <ul style="list-style-type: none"> Error in transfer or occurrence of events (detection of AL, NACK, time-out, a start condition including a restart condition, or a stop condition) Receive-data-full (including matching with a slave address) Transmit-data-empty (including matching with a slave address) Transmission complete

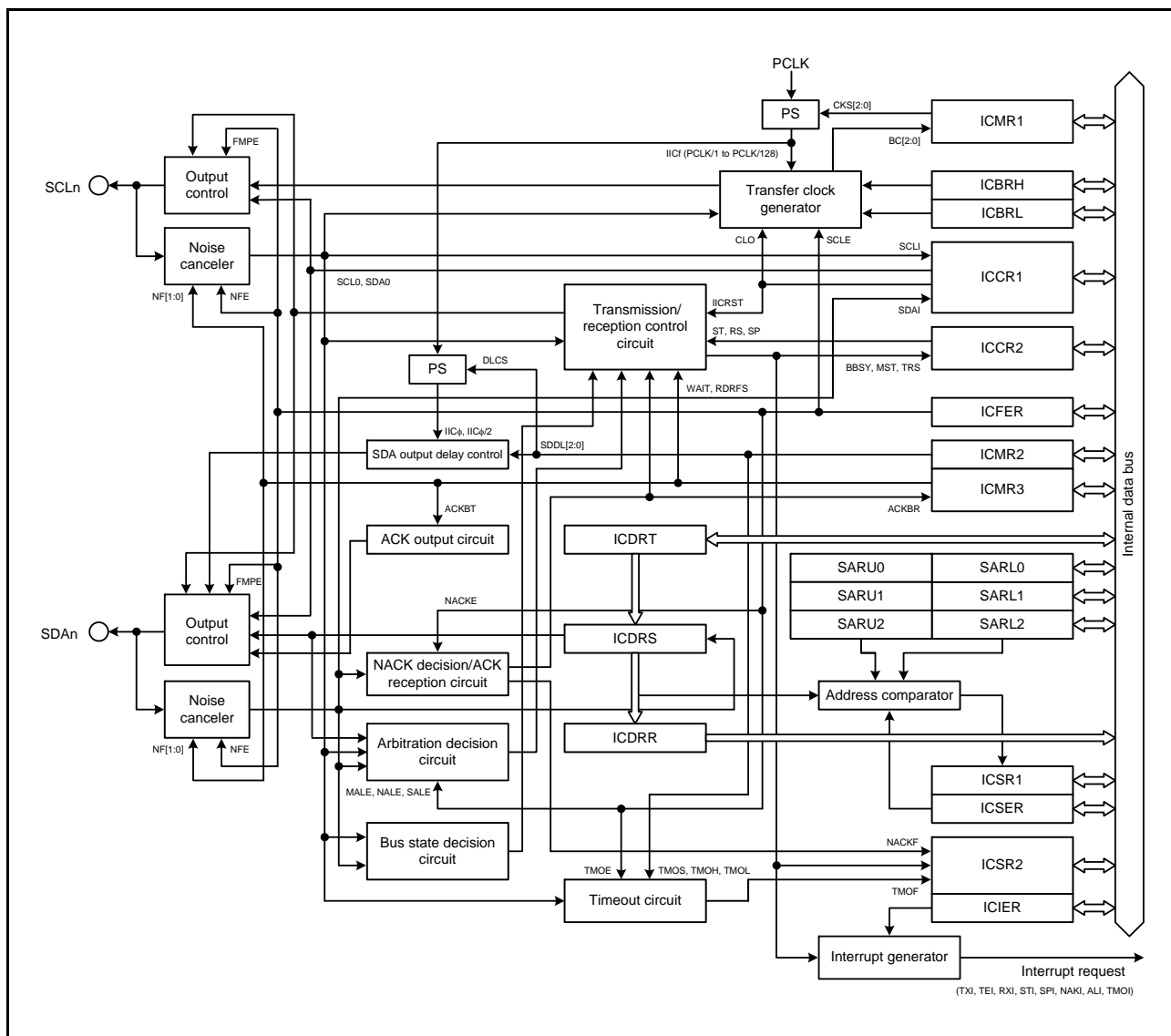


Figure 31.1 Block Diagram of RIIC

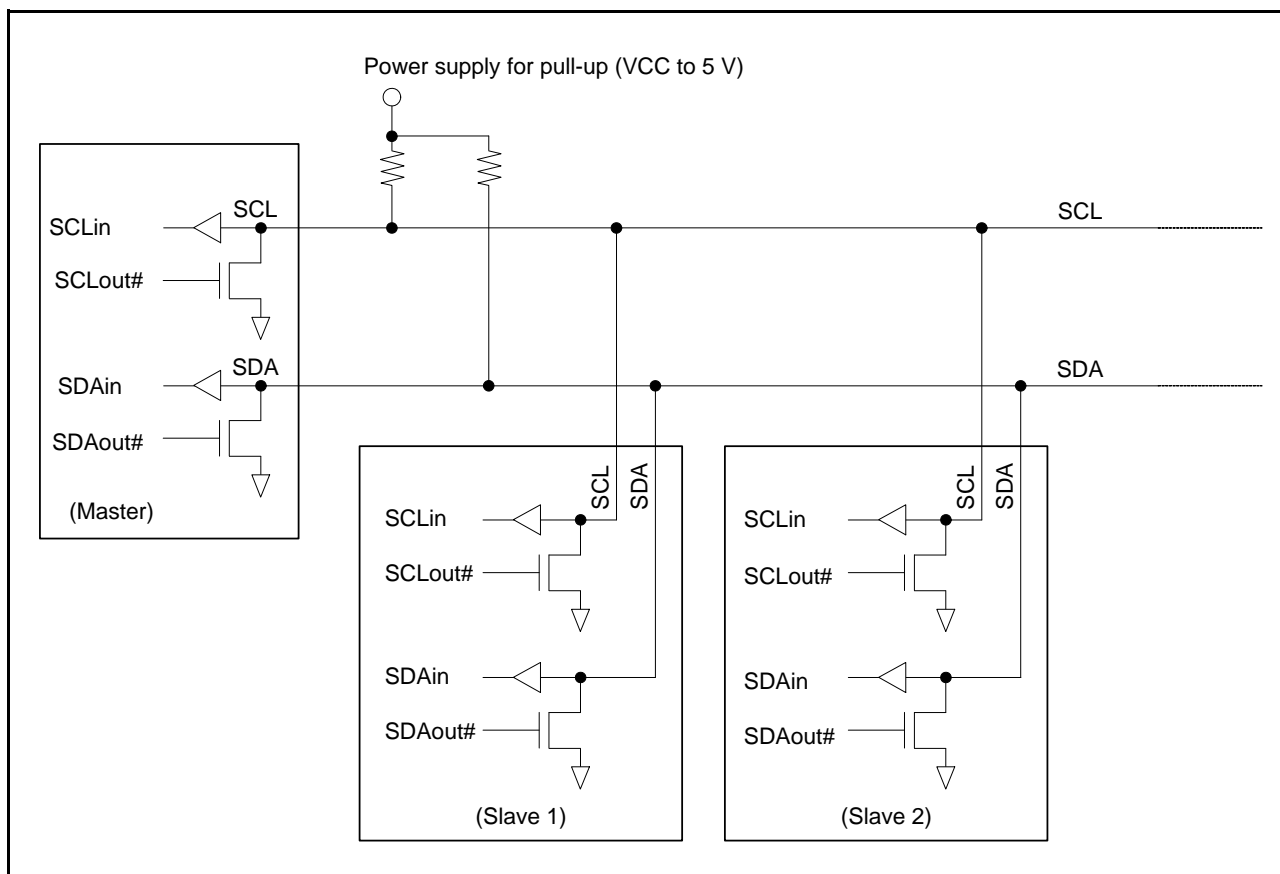


Figure 31.2 Connections to the External Circuit by the I/O Pins (I²C Bus Configuration Example)

Table 31.2 Pin Configuration

Channel	Pin Name	I/O	Function
RIIC0	SCL0	I/O	RIIC0 serial clock I/O pin
	SDA0	I/O	RIIC0 serial data I/O pin
RIIC1	SCL1	I/O	RIIC1 serial clock I/O pin
	SDA1	I/O	RIIC1 serial data I/O pin

31.2 Register Descriptions

Table 31.3 lists the registers of the RIIC.

Table 31.3 Registers of the RIIC (1 / 2)

Channel	Register Name	Symbol	Value after Reset	Address	Access Size
RIIC0	I ² C bus control register 1	ICCR1	1Fh	0008 8300h	8
	I ² C bus control register 2	ICCR2	00h	0008 8301h	8
	I ² C bus mode register 1	ICMR1	08h	0008 8302h	8
	I ² C bus mode register 2	ICMR2	06h	0008 8303h	8
	I ² C bus mode register 3	ICMR3	00h	0008 8304h	8
	I ² C bus function enable register	ICFER	72h	0008 8305h	8
	I ² C bus status enable register	ICSER	09h	0008 8306h	8
	I ² C bus interrupt enable register	ICIER	00h	0008 8307h	8
	I ² C bus status register 1	ICSR1	00h	0008 8308h	8
	I ² C bus status register 2	ICSR2	00h	0008 8309h	8
	Slave address register L0	SARL0	00h	0008 830Ah	8
	Slave address register U0	SARU0	00h	0008 830Bh	8
	Slave address register L1	SARL1	00h	0008 830Ch	8
	Slave address register U1	SARU1	00h	0008 830Dh	8
	Slave address register L2	SARL2	00h	0008 830Eh	8
	Slave address register U2	SARU2	00h	0008 830Fh	8
	I ² C bus bit rate low-level register	ICBRL	FFh	0008 8310h	8
	I ² C bus bit rate high-level register	ICBRH	FFh	0008 8311h	8
	I ² C bus transmit data register	ICDRT	FFh	0008 8312h	8
	I ² C bus receive data register	ICDRR	00h	0008 8313h	8
I ² C bus shift register	ICDRS	—	—	8	

Table 31.3 Registers of the RIIC (2 / 2)

Channel	Register Name	Symbol	Value after Reset	Address	Access Size
RIIC1	I ² C bus control register 1	ICCR1	1Fh	0008 8320h	8
	I ² C bus control register 2	ICCR2	00h	0008 8321h	8
	I ² C bus mode register 1	ICMR1	08h	0008 8322h	8
	I ² C bus mode register 2	ICMR2	06h	0008 8323h	8
	I ² C bus mode register 3	ICMR3	00h	0008 8324h	8
	I ² C bus function enable register	ICFER	72h	0008 8325h	8
	I ² C bus status enable register	ICSER	09h	0008 8326h	8
	I ² C bus interrupt enable register	ICIER	00h	0008 8327h	8
	I ² C bus status register 1	ICSR1	00h	0008 8328h	8
	I ² C bus status register 2	ICSR2	00h	0008 8329h	8
	Slave address register L0	SARL0	00h	0008 832Ah	8
	Slave address register U0	SARU0	F0h	0008 832Bh	8
	Slave address register L1	SARL1	00h	0008 832Ch	8
	Slave address register U1	SARU1	F0h	0008 832Dh	8
	Slave address register L2	SARL2	00h	0008 832Eh	8
	Slave address register U2	SARU2	F0h	0008 832Fh	8
	I ² C bus bit rate low-level register	ICBRL	FFh	0008 8330h	8
	I ² C bus bit rate high-level register	ICBRH	FFh	0008 8331h	8
	I ² C bus transmit data register	ICDRT	FFh	0008 8332h	8
	I ² C bus receive data register	ICDRR	00h	0008 8333h	8
I ² C bus shift register	ICDRS	—	—	8	

31.2.1 I²C Bus Control Register 1 (ICCR1)

Addresses: RIIC0.ICCR1 0008 8300h, RIIC1.ICCR1 0008 8320h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAI	SDA Bus Input Monitor	0: SDA _n pin input is at a low level 1: SDA _n pin input is at a high level	R
b1	SCLI	SCL Bus Input Monitor	0: SCL _n pin input is at a low level 1: SCL _n pin input is at a high level	R
b2	SDAO	SDA Output Control*1*2	Read: 0: SDA _n pin output is at a low level 1: SDA _n pin is in a high-impedance state Write: 0: Changes the SDA _n pin output to a low level 1: Changes the SDA _n pin in a high-impedance state (High level output is achieved through an external pull-up resistor.)	R/W *1, *2
b3	SCLO	SCL Output Control*1*2	Read: 0: SCL _n pin output is at a low level 1: SCL _n pin is in a high-impedance state Write: 0: Changes the SCL _n pin output to a low level 1: Changes the SCL _n pin in a high-impedance state (High level output is achieved through an external pull-up resistor.)	R/W *1, *2
b4	SOWP	SCLO/SDAO Write Protect*2	0: Allows the SCLO and SDAO bits to be rewritten (This bit is always read as 1.)	R/W *2
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 ² C Bus Interface Internal Reset	0: Clears the RIIC reset or internal reset 1: Initiates the RIIC reset or internal reset (Clears the bit counter and the SCL _n /SDA _n output latch)	R/W
b7	ICE	I ² C Bus Interface Enable	0: Disables the RIIC (the SCL _n pin and SDA _n pin function as ports) 1: Enables the RIIC transfer function (the SCL _n pin and SDA _n pin drive the bus)	R/W

Note 1. Do not write to these bits during communication. Changing a value during communication may cause a transmission or reception failure or an AL error.

Note 2. To change the SDAO and SCLO bits, set the SOWP bit to 0 at the same timing to set the SDAO and SCLO bits to 0.

ICCR1 enables or disables the RIIC, resets the internal state of the RIIC, outputs an extra SCL clock cycle, and manipulates and monitors the SCLn pin and SDAn pin.

SDAI Bit (SDA Bus Input Monitor)

This bit monitors the input level of the SDAn pin.

SCLI Bit (SCL Bus Input Monitor)

This bit monitors the input level of the SCLn pin.

SDAO Bit (SDA Output Control)

This bit controls the output level of the SDAn pin. This bit also monitors the output state of the SDAn pin.

SCLO Bit (SCL Output Control)

This bit controls the output level of the SCLn pin. This bit also monitors the output state of the SCLn pin.

SOWP Bit (SCLO/SDAO Write Protect)

This bit controls the modification of the SCLO and SDAO bits.

CLO Bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.

Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.

For details on this function, see section 31.11.2, Extra SCL Clock Cycle Output Function.

IICRST Bit (I²C Bus Interface Internal Reset)

This bit is used to reset the internal states of the RIIC.

Setting this bit to 1 initiates an RIIC reset or internal reset.

Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. Table 31.4 lists the resets of the RIIC.

The RIIC reset resets all registers including the BBSY flag in ICCR2 and internal states of the RIIC, and the internal reset resets the bit counter (BC[2:0] bits in ICMR1), the I²C bus shift register (ICDRS), and the I²C bus status registers (ICSR1 and ICSR2) as well as the internal states of the RIIC. For the reset conditions for each register, see section 31.14, Reset States.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error.

If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the SCLn pin and SDAn pin at a high impedance.

Note: If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If an internal reset is necessary because the RIIC hangs up with the SCLn line in a low level output state in slave mode, initiate an internal reset and then issue a restart condition from the master device or resume communication from the start condition issuance after issuing a stop condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 31.4 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers and internal states of the RIIC.
	1	Internal reset	Reset the BC[2:0] bits in ICMR1, and the ICSR1, ICSR2, ICDRS registers and the internal states of the RIIC.

ICE Bit (I²C Bus Interface Enable)

This bit is used to enable or disable the transfer operation of the RIIC.

When this bit is set to 0 to disable the RIIC, the SCLn pin and SDAn pin function as ports. An RIIC reset is initiated by setting the IICRST bit to 1 with the ICE bit set to 0, and an internal reset is initiated by setting the IICRST bit to 1 with the ICE bit set to 1.

To prevent unexpected communications, set the RIIC registers with the ICE bit set to 0 (to disable the RIIC), and set the ICE bit to 1 (to enable the transfer operation) after finishing all register settings.

Note: In addition to the I²C bus pin functions, other functions are also multiplexed onto the pins of the RX62N/RX621 Group. To use the pins as I²C bus pins (SCLn pin and SDAn pin), disable the other multiplexed functions. Since both of the SCLn pin and SDAn pin of the I²C bus pins are I/O pins, set the corresponding PORTm.DDR register to 0 (input) and set the PORTm.ICR register to 1 (input buffer enabled).

31.2.2 I²C Bus Control Register 2 (ICCR2)

Addresses: RIIC0.ICCR2 0008 8301h, RIIC1.ICCR2 0008 8321h

	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 always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W
b5	TRS	Transmit/Receive Mode	0: Receive mode 1: Transmit mode	R/W*
b6	MST	Master/Slave Mode	0: Slave mode 1: Transmit mode	R/W*
b7	BBSY	Bus Busy Detection Flag	0: The I ² C bus is released (bus free state) 1: The I ² C bus is occupied (bus busy state or in the bus free state)	R

Note : * When the MTWP bit in ICMR1 is set to 1, the MST and TRS bits can be written to.

ICCR2 has a flag function that indicates whether or not the I²C bus is occupied and whether the RIIC is in transmit/receive or master/slave mode as well as a function to issue a start or stop condition.

ST Bit (Start Condition Issuance Request)

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free).

For details on the start condition issuance, see section 31.10, Start Condition/Restart Condition/Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been issued
- When the AL (arbitration lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free).

Note that arbitration may be lost if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy).

RS Bit (Restart Condition Issuance Request)

This bit is used to request that a restart condition be issued in master mode.

When this bit is set to 1 to request to issue a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the restart condition issuance, see section 31.10, Start Condition/Restart Condition/Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued or a start condition is detected
- When the AL (arbitration lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note 1. This bit cannot be written to while the BBSY flag is 0 (bus free state).

Note 2. Do not set the RS bit to 1 while issuing a stop condition.

Note 3. If the RS bit is set to 1 (restart condition issuance request) in mode other than master mode, the restart condition is not issued in this mode but the restart condition issuance request bit remains set. If the operating mode changes to master mode with the bit not being cleared, the restart condition may be issued: this may hinder communications or cause an unexpected action.

SP Bit (Stop Condition Issuance Request)

This bit is used to request that a stop condition be issued in master mode.

When this bit is set to 1 to request to issue a stop condition, a stop condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the stop condition issuance, see section 31.10, 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 after reading SP = 1
- When a stop condition has been issued or a stop condition is detected
- When the AL (arbitration lost) flag in ICSR2 is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note 1. Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free).

Note 2. Do not set the SP bit to 1 while a restart condition is being issued.

TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.

The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.

The value of the TRS bit is automatically changed to the value for transmission mode or reception mode by detection or issuing of a start condition, setting or clearing of the R/W# bit, etc. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When the R/W# bit added to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in ICSEER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1

[Clearing conditions]

- When a stop condition is detected
- The AL (arbitration lost) flag in ICSR2 being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave mode, a match between the received address and the address enabled in ICSEER when the value of the received R/W# bit is 0 (including cases where the received address is the general call address)
- In slave mode, a restart condition is detected (a start condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0)
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.

The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.

The value of the MST bit is automatically changed to the value for master mode or slave mode by detection or issuing of a start condition, etc. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration lost) flag in ICSR2 is set to 1
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

BBSY Flag (Bus Busy Detection)

The BBSY flag indicates whether the I²C bus is occupied (bus busy) or released (bus free).

This bit is set to 1 when the SDAn line changes from high to low under the condition of SCLn = high, assuming that a start condition has been issued.

When the SDAn line changes from low to high under the condition of SCLn = high, this bit is cleared to 0 after the bus free time (specified in ICBRL) start condition is not detected, assuming that a stop condition has been issued.

[Setting condition]

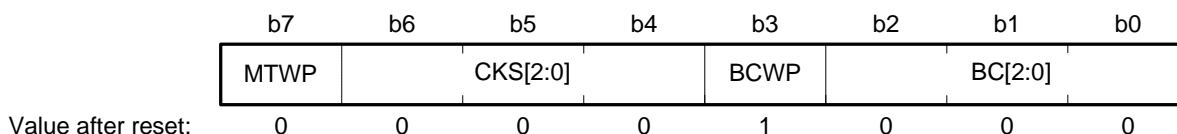
- When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in ICBRL) start condition is not detected after detecting a stop condition
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (RIIC reset)

31.2.3 I²C Bus Mode Register 1 (ICMR1)

Addresses: RIIC0.ICMR1 0008 8302h, RIIC1.ICMR1 0008 8322h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BC[2:0]	Bit Counter*	b2 b1 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*
b3	BCWP	BC Write Protect*	0: Enables a value to be written in the BC[2:0] bits (This bit is always read as 1.)	R/W*
b6 to b4	CKS[2:0]	Internal Reference Clock Selection	b6 b5 b4 0 0 0: PCLK/1 clock 0 0 1: PCLK/2 clock 0 1 0: PCLK/4 clock 0 1 1: PCLK/8 clock 1 0 0: PCLK/16 clock 1 0 1: PCLK/32 clock 1 1 0: PCLK/64 clock 1 1 1: PCLK/128 clock	R/W
b7	MTWP	MST/TRS Write Protect	0: Disables writing to the MST and TRS bits in ICCR2 1: Enables writing to the MST and TRS bits in ICCR2	R/W

Note : * Set the BCWP bit to 0 to rewrite the BC[2:0] bits. The BC[2:0] bits must be rewritten by using the MOV instruction.

ICMR1 specifies the internal reference clock source within the RIIC, indicates the number of bits to be transferred, and protects the MST and TRS bits in ICCR2 from being written.

BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCLn line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred frames when the SCLn line is at a low level.

The values of the BC[2:0] bits return to 000b at the end of a data transfer including the acknowledge bit or when a start condition including a restart condition is detected.

BCWP Bit (BC Write Protect)

This bit enables a value to be written in the BC[2:0] bits.

CKS[2:0] Bits (Internal Reference Clock Selection)

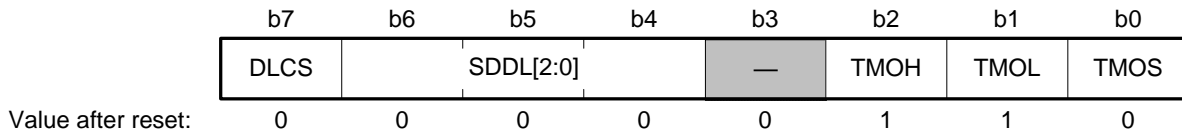
These bits select a reference clock source (IIC ϕ) inside the RIIC.

MTWP Bit (MST/TRS Write Protect)

This bit controls the modification of the MST and TRS bits in ICCR2.

31.2.4 I²C Bus Mode Register 2 (ICMR2)

Addresses: RIIC0.ICMR2 0008 8303h, RIIC1.ICMR2 0008 8323h



Bit	Symbol	Bit Name	Description	R/W
b0	TMOS	Timeout Detection Time Selection	0: Long mode is selected 1: Short mode is selected	R/W
b1	TMOL	Timeout L Count Control	0: Count is disabled while the SCLn line is at a low level 1: Count is enabled while the SCLn line is at a low level	R/W
b2	TMOH	Timeout H Count Control	0: Count is disabled while the SCLn line is at a high level 1: Count is enabled while the SCLn line is at a high level	R/W
b3	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b6 to b4	SDDL[2:0]	SDA Output Delay Counter	<ul style="list-style-type: none"> • When ICMR2.DLCS = 0 (IICϕ) <li style="margin-left: 20px;">b6 b4 0 0 0: No output delay 0 0 1: 1 IICϕ cycle 0 1 0: 2 IICϕ cycles 0 1 1: 3 IICϕ cycles 1 0 0: 4 IICϕ cycles 1 0 1: 5 IICϕ cycles 1 1 0: 6 IICϕ cycles 1 1 1: 7 IICϕ cycles • When ICMR2.DLCS = 1 (IICϕ/2) <li style="margin-left: 20px;">b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IICϕ cycles 0 1 0: 3 or 4 IICϕ cycles 0 1 1: 5 or 6 IICϕ cycles 1 0 0: 7 or 8 IICϕ cycles 1 0 1: 9 or 10 IICϕ cycles 1 1 0: 11 or 12 IICϕ cycles 1 1 1: 13 or 14 IICϕ cycles 	R/W
b7	DLCS	SDA Output Delay Clock Source Selection	0: The internal reference clock (IIC ϕ) is selected as the clock source of the SDA output delay counter 1: The internal reference clock divided by 2 (IIC ϕ /2) is selected as the clock source of the SDA output delay counter	R/W

ICMR2 has a timeout detection function and an SDA output delay function.

TMOS Bit (Timeout Detection Time Selection)

This bit is used to select long mode or short mode for the timeout detection time when the timeout detection function is enabled (TMOE bit = 1 in ICFER). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit-counter. While the SCLn line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IIC ϕ) as a count source.

For details on the timeout detection function, see section 31.11.1, Timeout Detection Function.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout detection function to count up while the SCLn line is held low when the timeout detection function is enabled (TMOE bit = 1 in ICFER).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout detection function to count up while the SCLn line is held high when the timeout detection function is enabled (TMOE bit = 1 in ICFER).

SDDL[2:0] Bits (SDA Output Delay Setup Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledge bit.

For details on this function, see section 31.5, Facility for Delaying SDA Output.

Note 1. Set the SDA output delay time to meet the I²C bus standard (within the data enable time/acknowledge enable time*2) or the SMBus standard (within the data hold time: 300 ns or more, and SCL-clock low-level period – the data setup time: 250 ns). Note that, if a value outside the standard is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

Note 2. Data enable time/acknowledge enable time
3,450 ns (up to 100 kbps: standard mode [Sm])
900 ns (up to 400 kbps: fast mode [fm])
450 ns (up to 1 Mbps: fast mode plus [fm+])

DLCS bit (SDA Output Delay Clock Source Selection)

This bit is used to select the internal reference clock (IIC ϕ) or the internal reference clock divided by 2 (IIC ϕ /2) as the clock source of the SDA output delay time.

31.2.5 I²C Bus Mode Register 3 (ICMR3)

Addresses: RIIC0.ICMR3 0008 8304h, RIIC1.ICMR3 0008 8324h

b7	b6	b5	b4	b3	b2	b1	b0
SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:							
0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Selection	b1 b0 0 0: Noise of up to 1-IIC ϕ range is filtered out (single-stage filter) 0 1: Noise of up to 2- IIC ϕ range is filtered out (2-stage filter) 1 0: Noise of up to 3- IIC ϕ range is filtered out (3-stage filter) 1 1: Noise of up to 4- IIC ϕ range 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*1	W*1
b5	RDRFS	RDRF Flag Set Timing Selection	0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle (The SCLn line is not held low at the falling edge of the eighth clock cycle.) 1: The RDRF flag is set at the rising edge of the eighth SCL clock cycle (The SCLn line is held low at the falling edge of the eighth clock cycle.) • Low-hold is released by writing a value to the ACKBT bit	R/W *2
b6	WAIT	WAIT	0: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) • Low-hold is released by reading ICDRR	R/W *2
b7	SMBS	SMBus/I ² C Bus Selection	0: The I ² C bus is selected 1: The SMBus is selected	R/W

Note 1. If it is attempted to write 1 to both ACKWP and ACKBT bits, the ACKBT bit cannot be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

ICMR3 has functions to send/receive acknowledge and to select the RDRF set timing in RIIC receive operation, WAIT operation, and the SCLn pin and SDAn pin input level of the RIIC.

NF[1:0] Bits (Noise Filter Stage Selection)

These bits are used to select the number of stages of the digital noise filter.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCLn line high-level period or low-level period. If the noise range is set to a value of (SCL clock width: high-level period or low-level period, whichever is shorter) - [1.5 internal reference clock sync (IIC ϕ) + analog noise filter: 120ns (reference value)] 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 a 1 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1

[Clearing conditions]

- When a 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledge timing in receive mode.

[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition issuance is detected (when a stop condition is detected with the SP bit in ICCR2 set to 1)
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (RIIC reset)

Note: The ACKBT bit must be modified while the ACKWP bit is 1. If the ACKBT bit is modified with the ACKWP bit cleared to 0, writing to the ACKBT bit is disabled.

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Selection)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCLn line low at the falling edge of the eighth SCL clock cycle.

When the RDRFS bit is 0, the SCLn line is not held low at the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCLn line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCLn line is released by writing a value to the ACKBT bit.

After data is received with this setting, the SCLn line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1) according to receive data.

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (ICDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCLn line is held low from the falling edge of the ninth clock cycle until the ICDRR value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, be sure to read the ICDRR beforehand.

SMBS Bit (SMBus/I²C Bus Selection)

Setting this bit to 1 selects the SMBus and enables the HOAE bit in IC SER.

31.2.6 I²C Bus Function Enable Register (ICFER)

Addresses: RIIC0.ICFER 0008 8305h, RIIC1.ICFER 0008 8325h

	b7	b6	b5	b4	b3	b2	b1	b0
	FMPE	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOE	Timeout Detection Function Enable	0: The timeout detection function is disabled 1: The timeout detection function is enabled	R/W
b1	MALE	Master Arbitration Lost Detection Enable	0: Master arbitration lost detection is disabled (Disables the arbitration lost detection function and does not clear the MST and TRS bits in ICCR2 automatically when arbitration is lost.) 1: Master arbitration lost detection is enabled (Enables the arbitration lost detection function and clears the MST and TRS bits in ICCR2 automatically when arbitration is lost.)	R/W
b2	NALE	NACK Transmission Arbitration Lost Detection Enable	0: NACK transmission arbitration lost detection is disabled 1: NACK transmission arbitration lost detection is enabled	R/W
b3	SALE	Slave Arbitration Lost Detection Enable	0: Slave arbitration lost detection is disabled 1: Slave arbitration lost detection is enabled	R/W
b4	NACKE	NACK Reception Transfer Suspension Enable	0: Transfer operation is not suspended during NACK reception (transfer suspension disabled) 1: Transfer operation is suspended during NACK reception (transfer suspension enabled)	R/W
b5	NFE	Digital Noise Filter Circuit Enable	0: No digital noise filter circuit is used 1: A digital noise filter circuit is used	R/W
b6	SCLE	SCL Synchronous Circuit Enable	0: No SCL synchronous circuit is used 1: An SCL synchronous circuit is used	R/W
b7	FMPE	Fast-mode Plus Enable	0: No fm+ slope control circuit is used for the SCLn pin and SDAn pin 1: An fm+ slope control circuit is used for the SCLn pin and SDAn pin	R/W

ICFER enables or disables the timeout detection function, the arbitration lost detection function, and the receive operation suspension function during NACK reception, and selects the use of a digital noise filter circuit and SCL synchronous circuit.

TMOE Bit (Timeout Detection Function Enable)

This bit is used to enable or disable the timeout detection function.

For details on the timeout detection function, see section 31.11.1, Timeout Detection Function.

MALE Bit (Master Arbitration Lost Detection Enable)

This bit is used to specify whether to use the arbitration lost detection function in master mode. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

SALE Bit (Slave Arbitration Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received from the slave device in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended.

When the NACKE bit is 0, the next transfer operation is continued regardless of the received acknowledge content.

For details on the NACK reception transfer suspension function, see section 31.8.2, NACK Reception Transfer Suspension Function.

NFE Bit (Digital Noise Filter Circuit Enable)

This bit is used to specify whether to use a digital noise filter circuit.

SCLE Bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCLE bit is cleared to 0 (SCL synchronous circuit not used), the RIIC does not synchronize the SCL clock with the SCL input clock (by detecting the SCLn line level) for the SCL clock output operation in master mode, and the RIIC outputs the SCL clock with the transfer rate set in ICBRH and ICBRL regardless of the SCLn line state. For this reason, if the bus load of the I²C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.

This bit must not be cleared to 0 except for checking the output of the transfer rate that was set during debugging.

FMPE Bit (Fast-mode Plus Enable)

This bit is used to specify whether to use a slope control circuit for Fast-mode Plus[fm+].

When this bit is set to 1, a slope control circuit conforming to the Fast-mode Plus[fm+] slope control standard (tof) of the I²C bus is selected. When this bit is cleared to 0, a slope control circuit conforming to the Standard-mode[Sm] and Fast-mode[fm] slope control standard (tof) of the I²C bus is selected.

Set this bit to 1 when using the transmission rate within a range up to 1 Mbps (Fast-mode Plus[fm+]) of the I²C bus standard. Clear this bit to 0 when using the transmission rate at other rates (up to 100 kbps[Sm], up to 400 kbps[fm]) or for SMBus (10 to 100 kbps).

31.2.7 I²C Bus Status Enable Register (ICSER)

Addresses: RIIC0.ICSER 0008 8306h, RIIC1.ICSER 0008 8326h

	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 always read as 0. The write value should always 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 always read as 0. The write value should always be 0.	R/W
b7	HOAE	Host Address Enable	0: Host address detection is disabled 1: Host address detection is enabled	R/W

ICSER enables or disables comparison of slave addresses, general call address detection, device-ID command detection, and host address detection.

SAR_mE Bit (Slave Address Register m Enable) (m = 0 to 2)

This bit is used to enable or disable the received slave address and the slave address set in SARL_m and SARU_m.

When this bit is set to 1, the slave address set in SARL_m and SARU_m is enabled and is compared with the received slave address.

When this bit is cleared to 0, the slave address set in SARL_m and SARU_m 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 SARL_m and SARU_m (m = 0 to 2) and performs data receive operation.

When this bit is cleared to 0, the received slave address is ignored even if it matches the general call address.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the Device-ID address when a device ID (1111 100b) is received in the first frame after a start condition or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the RIIC recognizes that the Device-ID address has been received. When the following R/W# bit is 0 [W], the RIIC recognizes the second and the following frames as slave addresses and continues the receive operation.

When this bit is cleared to 0, the RIIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device-ID address detection, see section 31.7.3, 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 SARLm and SARUm (m = 0 to 2) and performs the receive operation.

When the SMBS bit in ICMR3 or the HOAE bit is cleared to 0, the received slave address is ignored even if it matches the host address.

31.2.8 I²C Bus Interrupt Enable Register (ICIER)

Addresses: RIIC0.ICIER 0008 8307h, RIIC1.ICIER 0008 8327h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOIE	Timeout Interrupt Enable	0: Timeout interrupt request (TMOI) is disabled 1: Timeout interrupt request (TMOI) is enabled	R/W
b1	ALIE	Arbitration Lost Interrupt Enable	0: Arbitration lost interrupt request (ALI) is disabled 1: Arbitration lost interrupt request (ALI) is enabled	R/W
b2	STIE	Start Condition Detection Interrupt Enable	0: Start condition detection interrupt request (STI) is disabled 1: Start condition detection interrupt request (STI) is enabled	R/W
b3	SPIE	Stop Condition Detection Interrupt Enable	0: Stop condition detection interrupt request (SPI) is disabled 1: Stop condition detection interrupt request (SPI) is enabled	R/W
b4	NAKIE	NACK Reception Interrupt Enable	0: NACK reception interrupt request (NAKI) is disabled 1: NACK reception interrupt request (NAKI) is enabled	R/W
b5	RIE	Receive Data Full Interrupt Enable	0: Receive data full interrupt request (ICRXI) is disabled 1: Receive data full interrupt request (ICRXI) is enabled	R/W
b6	TEIE	Transmit End Interrupt Enable	0: Transmit end interrupt request (ICTEI) is disabled 1: Transmit end interrupt request (ICTEI) is enabled	R/W
b7	TIE	Transmit Data Empty Interrupt Enable	0: Transmit data empty interrupt request (ICTXI) is disabled 1: Transmit data empty interrupt request (ICTXI) is enabled	R/W

ICIER enables or disables various interrupt sources.

TMOIE Bit (Timeout Interrupt Enable)

This bit is used to enable or disable timeout interrupt requests (TMOI) when the TMOF flag in ICSR2 is set to 1. A TMOI interrupt request is canceled by clearing the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration Lost Interrupt Enable)

This bit is used to enable or disable arbitration lost interrupt requests (ALI) when the AL flag in ICSR2 is set to 1. An ALI interrupt request is canceled by clearing the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Enable)

This bit is used to enable or disable start condition detection interrupt requests (STI) when the START flag in ICSR2 is set to 1. An STI interrupt request is canceled by clearing the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Enable)

This bit is used to enable or disable stop condition detection interrupt requests (SPI) when the STOP flag in ICSR2 is set to 1. An SPI interrupt request is canceled by clearing the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Enable)

This bit is used to enable or disable NACK reception interrupt requests (NAKI) when the NACKF flag in ICSR2 is set to 1. An NAKI interrupt request is canceled by clearing the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Data Full Interrupt Enable)

This bit is used to enable or disable receive data full interrupt requests (ICRXI) when the RDRF flag in ICSR2 is set to 1.

TEIE Bit (Transmit End Interrupt Enable)

This bit is used to enable or disable transmit end interrupts (ICTEI) when the TEND flag in ICSR2 is set to 1. An ICTEI interrupt request is canceled by clearing the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Enable)

This bit is used to enable or disable transmit data empty interrupts (ICTXI) when the TDRE flag in ICSR2 is set to 1.

31.2.9 I²C Bus Status Register 1 (ICSR1)

Addresses: RIIC0.ICSR1 0008 8308h, RIIC1.ICSR1 0008 8328h

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 <ul style="list-style-type: none"> This bit is set to 1 when the received slave address matches the SVA[6:0] value in SARL0 while the FS bit in SARU0 is 0 (7-bit address format selected). This bit is set to 1 when the received slave address matches a value of (1111 0b + SVA[1:0] in SARU0) and the following address matches the SARL0 value while the FS bit in SARU0 is 1 (10-bit address format selected). (This bit is set at the rising edge of the ninth SCL clock cycle in the SARL0 match determination frame.)	R/(W) *
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected 1: Slave address 1 is detected <ul style="list-style-type: none"> This bit is set to 1 when the received slave address matches the SVA[6:0] value in SARL1 while the FS bit in SARU1 is 0 (7-bit address format selected). This bit is set to 1 when the received slave address matches a value of (1111 0b + SVA[1:0] in SARU1) and the following address matches the SARL1 value while the FS bit in SARU1 is 1 (10-bit address format selected). (This bit is set at the rising edge of the ninth SCL clock cycle in the SARL1 match determination frame.)	R/(W) *
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected 1: Slave address 2 is detected <ul style="list-style-type: none"> This bit is set to 1 when the received slave address matches the SVA[6:0] value in SARL2 while the FS bit in SARU2 is 0 (7-bit address format selected). This bit is set to 1 when the received slave address matches a value of (1111 0b + SVA[1:0] in SARU2) and the following address matches the SARL2 value while the FS bit in SARU2 is 1 (10-bit address format selected). (This bit is set at the rising edge of the ninth SCL clock cycle in the SARL2 match determination frame.)	R/(W) *
b3	GCA	General Call Address Detection Flag	0: General call address is not detected 1: General call address is detected <ul style="list-style-type: none"> This bit is set to 1 when the received slave address matches the general call address (all 0). 	R/(W) *
b4	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b5	DID	Device-ID Command Detection Flag	0: Device-ID command is not detected 1: Device-ID command is detected <ul style="list-style-type: none"> 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) *
b6	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b7	HOA	Host Address Detection Flag	0: Host address is not detected 1: Host address is detected <ul style="list-style-type: none"> This bit is set to 1 when the received slave address matches the host address (0001 000b). 	R/(W) *

Note : * Only 0 can be written to clear the flag.

ICSR1 indicates various address detection statuses.

AASm Flag (Slave Address m Detection) (m = 0 to 2)

[Setting conditions]

<For 7-bit address format: SARUm.FS = 0>

- When the received slave address matches the SVA[6:0] value in SARLm with the SARmE bit in ICSEr set to 1 (slave address m 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: SARUm.FS = 1>

- When the received slave address matches a value of (1111 0b + SVA[1:0] in SARUm) and the following address matches the SARLm value with the SARmE bit in ICSEr set to 1 (slave address m 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 AASm bit after reading AASm = 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: SARUm.FS = 0>

- When the received slave address does not match the SVA[6:0] value in SARLm with the SARmE bit in ICSEr set to 1 (slave address m detection enabled)

This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: SARUm.FS = 1>

- When the received slave address does not match a value of (1111 0b + SVA[1:0] in SARUm) with the SARmE bit in ICSEr set to 1 (slave address m detection enabled)

This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

- When the received slave address matches a value of (1111 0b + SVA[1:0] in SARUm) and the following address does not match the SARLm value with the SARmE bit in ICSEr set to 1 (slave address m detection enabled)

This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

GCA Flag (General Call Address Detection)

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]) with the GCAE bit in ICSEr set to 1 (general call address detection enabled)

This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit after reading GCA = 1
 - When a stop condition is detected
 - When the received slave address does not match the general call address (0000 000b + 0 [W]) with the GCAE bit in ICSEr set to 1 (general call address detection enabled)
- This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection)

[Setting condition]

- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]) with the DIDE bit in IC SER set to 1 (Device-ID address detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100b)) with the DIDE bit in IC SER set to 1 (Device-ID address detection enabled)
This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]) and the second frame does not match any of slave addresses 0 to 2 with the DIDE bit in IC SER set to 1 (Device-ID address detection enabled)
This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

HOA Flag (Host Address Detection)

[Setting condition]

- When the received slave address matches the host address (0001 000b) with the HOAE bit in IC SER set to 1 (host address detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA bit after reading HOA = 1
- When a stop condition is detected
- When 0 is written to the SMBS bit in ICMR3 or the HOAE bit in IC SER
- When the received slave address does not match the host address (0001 000b) with the HOAE bit in IC SER set to 1 (host address detection enabled)
This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

31.2.10 I²C Bus Status Register 2 (ICSR2)

Addresses: RIIC0.ICSR2 0008 8309h, RIIC1.ICSR2 0008 8329h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOF	Timeout Detection Flag	0: Timeout is not detected 1: Timeout is detected	R/(W) *
b1	AL	Arbitration Lost Flag	0: Arbitration is not lost 1: Arbitration is lost	R/(W) *
b2	START	Start Condition Detection Flag	0: Start condition is not detected 1: Start condition is detected	R/(W) *
b3	STOP	Stop Condition Detection Flag	0: Stop condition is not detected 1: Stop condition is detected	R/(W) *
b4	NACKF	NACK Detection Flag	0: NACK is not detected 1: NACK is detected	R/(W) *
b5	RDRF	Receive Data Full Flag	0: ICDRR contains no receive data 1: ICDRR contains receive data	R/(W) *
b6	TEND	Transmit End Flag	0: Data is being transmitted 1: Data has been transmitted	R/(W) *
b7	TDRE	Transmit Data Empty Flag	0: ICDRT contains transmit data 1: ICDRT contains no transmit data	R

Note : * Only 0 can be written to clear the flag.

ICSR2 indicates various interrupt request flags and statuses.

TMOF Flag (Timeout Detection)

This flag is set to 1 when the RIIC recognizes timeout after the SCLn line state remains unchanged for a certain period.

[Setting condition]

- When the SCLn line state remains unchanged for the period specified by bits TMOH, TMOL, and TMOS in ICMR2 with the TMOE bit in ICFER set to 1 (timeout detection enabled) in master mode or in the slave specification state.

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

AL Flag (Arbitration Lost 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 SDA line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL bit to 1 to indicate that the bus is occupied by another device.

The RIIC can also set the flag to indicate the detection of loss of arbitration during NACK transmission in master mode or during data transmission in slave mode.

[Setting conditions]

<When master arbitration lost detection is enabled: ICFER.MALE = 1>

- When the internal SDA output state does not match the SDA 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 SDA line is driven low while the internal SDA output is 1 (the SDA 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 SDA 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.MALE = 1>

- When the internal SDA output state does not match the SDA 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>

- When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL bit after reading AL = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Table 31.5 Relationship between Arbitration Lost Generation Sources and Arbitration Lost Enable Functions

ICFER		ICSR 2			
MAL					
E	NALE	SALE	AL	Error	Arbitration Lost Generation Source
1	*	*	1	Start condition issuance error	When internal SDA output state does not match SDA 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
*	1	*	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master receive mode or slave receive mode
*	*	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

[Legend] *: Don't care

START Flag (Start Condition Detection)

[Setting condition]

- When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection)

[Setting condition]

- When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

NACKF Flag (NACK Detection)

[Setting condition]

- When acknowledge is not received (NACK is received) from the receive device in transmit mode with the NACKEN bit in ICFER set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, clear the NACKF flag to 0.

RDRF Flag (Receive Data Full)

[Setting conditions]

- When receive data has been transferred from ICDRS to ICDRR
- This flag is set to 1 at the rising edge of the eighth or ninth SCL clock cycle (selected by the RDRFS bit in ICMR3)

When the received slave address matches after a start condition (or a restart condition) is detected with the TRS bit in ICCR2 cleared to 0

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

TEND Flag (Transmit End)

[Setting condition]

- At the rising edge of the ninth SCL clock cycle while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to ICDRT
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty)

[Setting conditions]

- When data has been transferred from ICDRT to ICDRS and ICDRT becomes empty
- When the TRS bit in ICCR2 is set to 1
 - a. When the MST bit in ICCR2 is set to 1 after a start condition (or a restart condition) is detected
 - b. When the RIIC enters transmit mode from receive mode
 - c. When 1 is written to while the ICMR1.MTWP bit is 1

When the received slave address matches while the TRS bit is 1

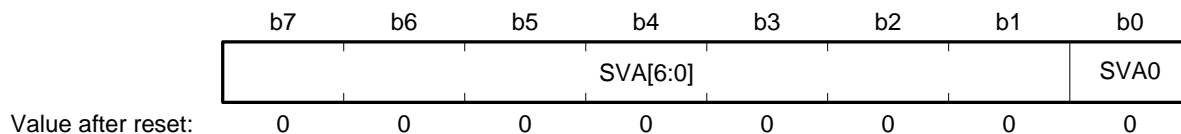
[Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is cleared to 0
 - a. When a stop condition is detected
 - b. When the RIIC enters receive mode from transmit mode
 - c. When 0 is written to while the ICMR1.MTWP bit is 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: When the NACKF flag is set to 1 while the 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.

31.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Addresses: RIIC0.SARL0 0008 830Ah, RIIC1.SARL0 0008 832Ah
 RIIC0.SARL1 0008 830Ch, RIIC1.SARL1 0008 832Ch
 RIIC0.SARL2 0008 830Eh, RIIC1.SARL2 0008 832Eh



Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	The least significant bit (LSB) of a 10-bit slave address is set <ul style="list-style-type: none"> When the FS bit in SARUm is 0 (7-bit address format), this bit is invalid When the FS bit in SARUm is 1 (10-bit address format), this bit is the LSB of the lower 8-bit address (combined with the SVA[6:0] bits) of a 10-bit slave address 	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	A slave address is set <ul style="list-style-type: none"> When the FS bit in SARUm is 0 (7-bit address format), these bits form a 7-bit slave address When the FS bit in SARUm is 1 (10-bit address format), these bits form the lower 8-bit address (combined with the SVA0 bit) of a 10-bit slave address 	R/W

SARLm sets slave address m (7-bit address or lower eight bits of 10-bit address).

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUm.FS = 1), this bit functions as the LSB of a 10-bit address and forms the lower eight bits of a 10-bit address in combination with the SVA[6:0] bits.

When the SARmE bit in ICSEr is set to 1 (SARLm and SARUm enabled) and the SARUm.FS bit is 1, this bit is valid. While the SARUm.FS bit or SARmE bit is 0, the setting of this bit is ignored.

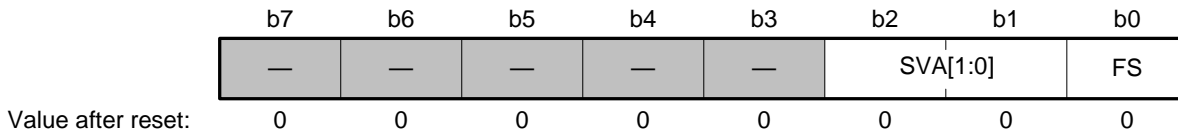
SVA[6:0] Bits (7-Bit Address/10-Bit Address Lower Bits)

When the 7-bit address format is selected (SARUm.FS = 0), these bits function as a 7-bit address. When the 10-bit address format is selected (SARUm.FS = 1), these bits function as the lower eight bits of a 10-bit address in combination with the SVA0 bit.

While the SARmE bit in ICSEr is 0, the setting of these bits is ignored.

31.2.12 Slave Address Register Um (SARUm) (m = 0 to 2)

Addresses: RIIC0.SARU0 0008 830Bh, RIIC1.SARU0 0008 832Bh
 RIIC0.SARU0 0008 830Dh, RIIC1.SARU1 0008 832Dh
 RIIC0.SARU0 0008 830Fh, RIIC1.SARU2 0008 832Fh



Bit	Symbol	Bit Name	Description	R/W
b0	FS	7-Bit/10-Bit Address Format Selection	0: The 7-bit address format is selected 1: The 10-bit address format is selected	R/W
b2, b1	SVA[1:0]	10-Bit Address Upper Bits	A slave address is set <ul style="list-style-type: none"> • When the SARUm.FS bit is 0 (7-bit address format), these bits are invalid • When the SARUm.FS bit is 1 (10-bit address format), these bits form the upper two bits of a 10-bit slave address 	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

SARUm selects 7-bit address format or 10-bit address format and sets the upper bits of a 10-bit slave address.

FS Bit (7-Bit/10-Bit Address Format Selection)

This bit is used to select 7-bit address or 10-bit address for slave address m (in SARLm and SARUm).

When the SARmE bit in ICSEr is set to 1 (SARLm and SARUm enabled) and the SARUm.FS bit is 0, the 7-bit address format is selected for slave address m, the SVA[6:0] setting in SARLm is valid, and the settings of the SVA[1:0] bits and the SVA0 bit in SARLm are ignored.

When the SARmE bit in ICSEr is set to 1 (SARLm and SARUm enabled) and the SARUm.FS bit is 1, the 10-bit address format is selected for slave address m and the settings of the SVA[1:0] bits and SARLm are valid.

While the SARmE bit in ICSEr is 0 (SARLm and SARUm disabled), the setting of the SARUm.FS bit is invalid.

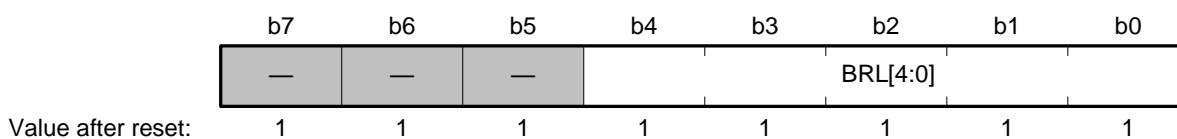
SVA[1:0] Bits (10-Bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), these bits function as the upper two bits of a 10-bit address.

When the SARmE bit in ICSEr is set to 1 (SARLm and SARUm enabled) and the SARUm.FS bit is 1, these bits are valid. While the SARUm.FS bit or SARmE bit is 0, the setting of these bits is ignored.

31.2.13 I²C Bus Bit Rate Low-Level Register (ICBRL)

Addresses: RIIC0.ICBRL 0008 8310h, RIIC1.ICBRL 0008 8330h



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 always read as 1. The write value should always 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 (see section 31.8, Function to Automatically Hold SCLn Clock Low); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time*.

ICBRL counts the low-level period with the internal reference clock source (IIC ϕ) specified by the CKS[2:0] bits in ICMR1.

Note: * Data setup time (tSU: DAT)

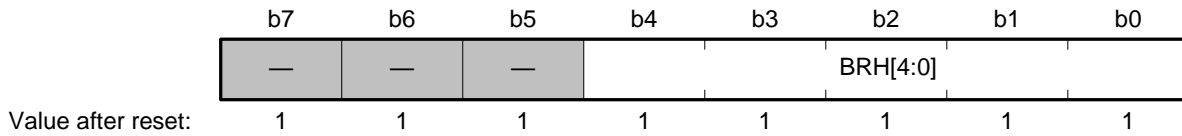
250 ns (up to 100 kbps: standard mode [Sm])

100 ns (up to 400 kbps: fast mode [fm])

50 ns (up to 1 Mbps: fast mode plus [fm+])

31.2.14 I²C Bus Bit Rate High-Level Register (ICBRH)

Addresses: RIIC0.ICBRH 0008 8311h, RIIC1.ICBRH 0008 8331h



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 always read as 1. The write value should always 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 ϕ) specified by the CKS[2:0] bits in ICMR1.

The I²C transfer rate and the SCL clock duty are calculated using the following expression.

$$\text{Transfer rate} = 1 / \{[(ICBRH + 1) + (ICBRL + 1)] / IIC\phi^{*1} + SCLn \text{ line rising time } [tr] + SCLn \text{ line falling time } [tf]\}$$

$$\text{Duty cycle} = \{SCLn \text{ line rising time } [tr]^{*2} + (ICBRH + 1 / IIC\phi)\} / \{SCLn \text{ line falling time } [tf]^{*2} + (ICBRL+1 / IIC\phi)\}$$

Note 1. IIC ϕ = PCLK \times 10⁶ \times Division ratio

Note 2. The SCLn line rising time [tr] and SCLn line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I²C bus standard from NXP Semiconductors.

Table 31.6 shows examples of ICBRH/ICBRL settings.

Table 31.6 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)
1000	000b	2 (E2h)	3 (E3h)	000b	2 (E2h)	4 (E4h)	000b	3 (E3h)	6 (E6h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	16			20			25		
	CKS [2:0]	ICBRH	ICBRL	CKS [2:0]	ICBRH	ICBRL	CKS [2:0]	ICBRH	ICBRL
10	101b	22 (F6h)	25 (F9h)	110b	13 (EDh)	15 (EFh)	110b	16 (F0h)	20 (F4h)
50	011b	16 (F0h)	19 (F3h)	011b	21 (F5h)	24 (F8h)	100b	12 (ECh)	15 (EFh)
100	010b	15 (EFh)	18 (F2h)	010b	19 (F3h)	23 (F7h)	010b	24 (F8h)	29 (FDh)
400	000b	9 (E9h)	20 (F4h)	000b	11 (EBh)	25 (F9h)	001b	7 (E7h)	16 (F0h)
1000	000b	4 (E4h)	7 (E7h)	000b	5 (E5h)	9 (E9h)	000b	6 (E6h)	12 (ECh)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	30			33			50		
	CKS [2:0]	ICBRH	ICBRL	CKS [2:0]	ICBRH	ICBRL	CKS [2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)	110b	22 (F6h)	26 (FAh)	111b	16 (F0h)	20 (F4h)
50	100b	15 (EFh)	18 (F2h)	100b	17 (F1h)	20 (F4h)	100b	26 (FAh)	31 (FFh)
100	010b	2 (E2h)	3 (E3h)	011b	16 (F0h)	19 (F3h)	011b	24 (F8h)	29 (FDh)
400	001b	8 (E8h)	19 (F3h)	001b	9 (E9h)	21 (F5h)	010b	7 (E7h)	16 (F0h)
1000	000b	7 (E7h)	14 (EEh)	000b	8 (E8h)	16 (F0h)	000b	12 (ECh)	24 (F8h)

Note: CBRH/ICBRL settings in these tables are calculated using the following values:

SCLn line rising time (tr): 100 kbps or less, [Sm]: 1000 ns, 400 kbps or less, [Fm]: 300 ns, 1 Mbps or less, [Fm+]: 120 ns

SCLn line falling time (tf): 400 kbps or less, [Sm/Fm]: 300 ns, 1 Mbps or less, [Fm+]: 120 ns

For the specified values of SCLn line rising time (tr) and SCLn line falling time (tf), see the I²C bus standard from NXP Semiconductors.

31.2.15 I2C Bus Transmit Data Register (ICDRT)

Addresses: RIIC0.ICDRT 0008 8312h, RIIC1.ICDRT 0008 8332h



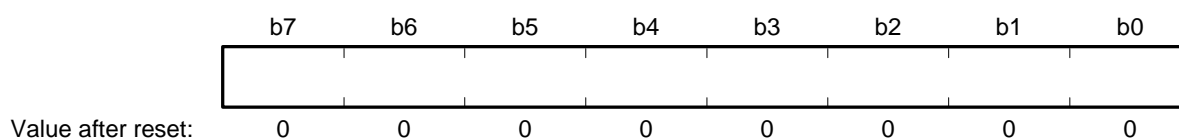
ICDRT is an 8-bit readable/writable register that stores transmit data. When ICDRT detects a space in the I2C bus shift register (ICDRS), it transfers the transmit data that has been written to ICDRT to ICDRS and starts transmitting data in transmit mode.

The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data has been written to ICDRT while the ICDRS data is being transmitted.

ICDRT can always be read and written. Write transmit data to ICDRT once when a transmit data empty interrupt (ICTXI) request is generated.

31.2.16 I2C Bus Receive Data Register (ICDRR)

Addresses: RIIC0.ICDRR 0008 8313h, RIIC1.ICDRR 0008 8333h



ICDRR is an 8-bit read-only register that stores receive data. When one byte of data has been received, the received data is transferred from the I2C bus shift register (ICDRS) to ICDRR to enable the next data to be received.

The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data has been read from ICDRR while ICDRS is receiving data.

ICDRR cannot be written. Read data from ICDRR once when a receive data full interrupt (ICRXI) request is generated.

If ICDRR receives the next receive data before the current data is read from ICDRR (while the RDRF flag in ICSR2 is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

31.2.17 I2C 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 SDA pin. During reception, data is transferred from ICDRS to ICDRR after one byte of data has been received.

ICDRS cannot be accessed directly.

31.3 Operation

31.3.1 Communication Data Format

The I²C 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 31.3 shows the I²C bus format, and Figure 31.4 shows the I²C bus timing.

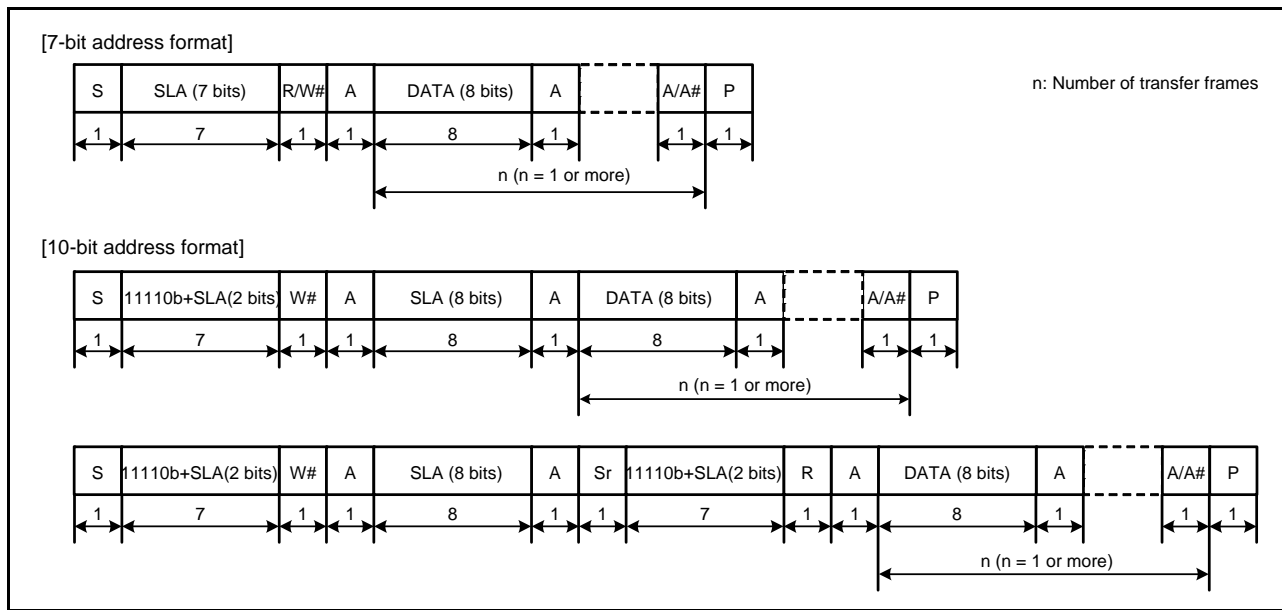


Figure 31.3 I²C Bus Format

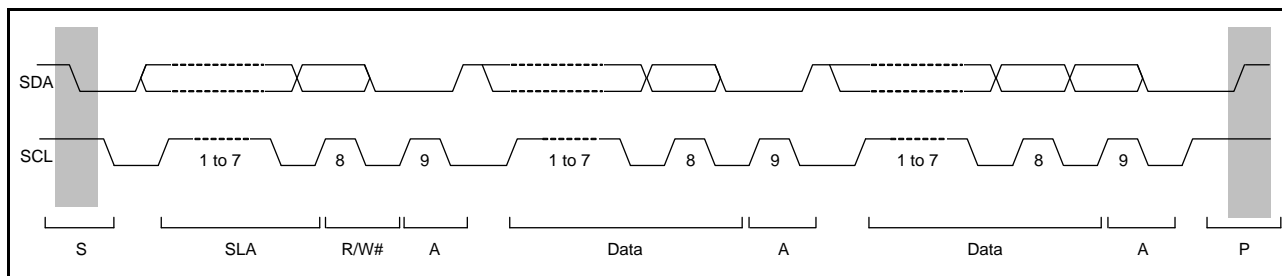


Figure 31.4 I²C Bus Timing (SLA = 7 Bits)

[Legend]

- S: Start condition. The master device drives the SDA_n line low from high level while the SCL_n line is at a high level.
- SLA: Slave address, by which the master device selects a slave device.
- R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.
- A: Acknowledge. The receive device drives the SDA_n line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)
- Sr: Restart condition. The master device drives the SDA_n line low from the high level after the setup time has elapsed with the SCL_n line at the high level.
- DATA: Transmitted or received data
- P: Stop condition. The master device drives the SDA_n line high from low level while the SCL_n line is at a high level.

31.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 31.5.

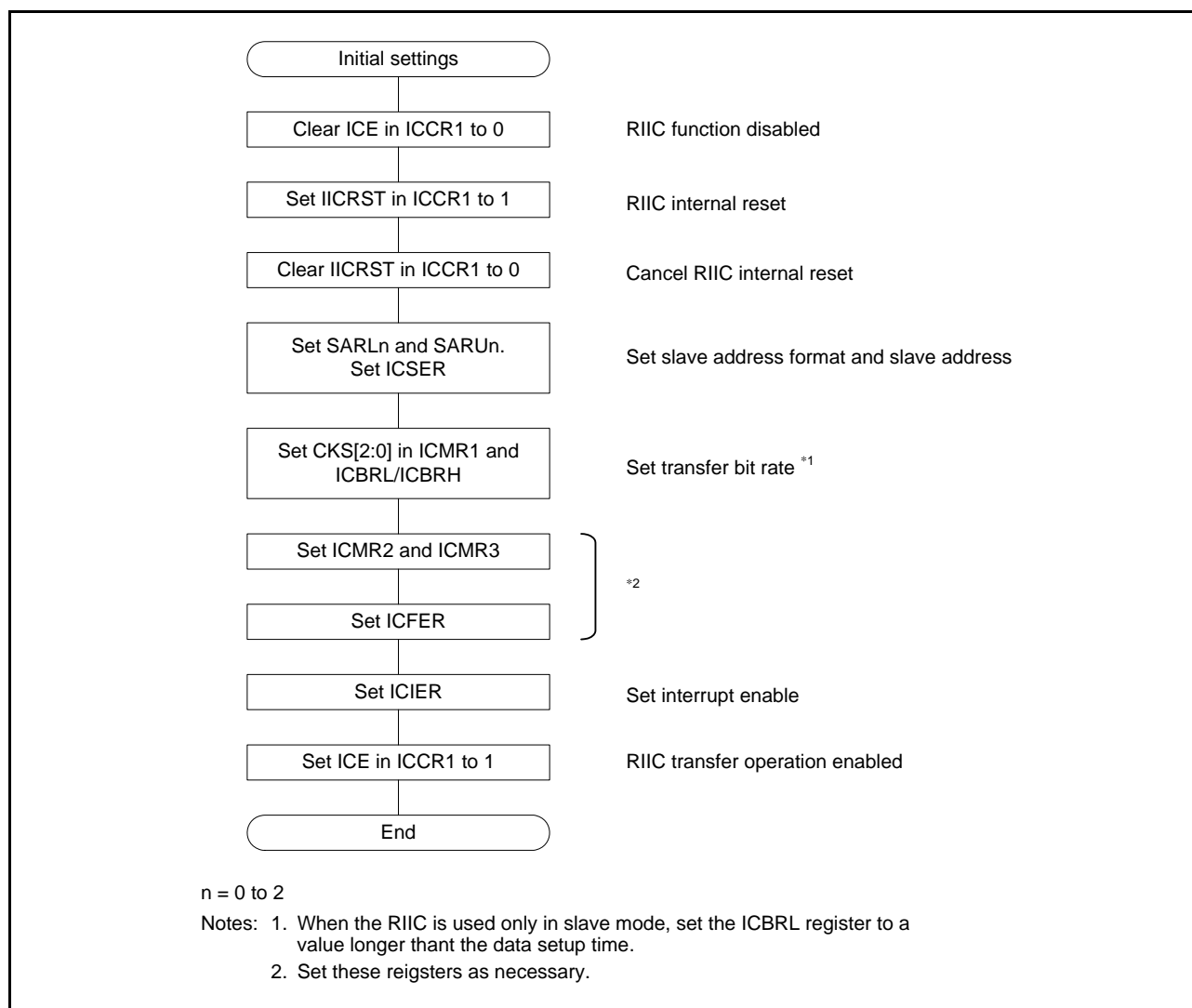


Figure 31.5 Example of RIIC Initialization Flow

31.3.3 Master Transmitter Operation

In master transmitter operation, the RIIC outputs the SCL (clock) and transmitted data signals as the master device, and the slave device returns acknowledgements. Figure 31.6 shows an example of usage of master transmission and Figure 31.7 to Figure 31.9 show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

1. Set the IICRST bit in ICCR1 to 1 (internal reset) and then clear the IICRST bit to 0 (canceling reset) with the ICE bit in ICCR1 cleared to 0 (disabling the interface). This initializes the internal state and the various flags of ICSR1. After that, set registers SARLn, SARUn, ICSEr, ICMR1, ICBRH, and ICBRL (n = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see Figure 31.5). When the necessary register settings have been completed, set the ICE bit to 1 (to enable transfer). This step is not necessary if initialization of the RIIC has already been completed.
2. Read the BBSY flag in ICCR2 to check that the bus is open, and then set the ST bit in ICCR2 to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the START flag in ICSR2 are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA_n 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 transmitter mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. Once the data for transmission are written to ICDRT, the TDRE flag is automatically cleared to 0, the data are transferred from ICDRT to ICDRS, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmitter or master receiver 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 transmitter mode.
 Since the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
 For data transmission with an address in the 10-bit format, start by writing 1111 0b, the two higher-order bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the eight lower-order bits of the slave address to ICDRT.
4. After confirming that the TDRE flag in ICSR2 is 1, write the data for transmission to the ICDRT register. The RIIC automatically holds the SCL_n line low until the data for transmission are ready or a stop condition is issued.
5. After all bytes of data for transmission have been written to the ICDRT register, wait until the value of the TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.
6. Upon detecting the stop condition, the RIIC automatically clears the MST and TRS bits in ICCR2 to 00b and enters slave receiver mode. Furthermore, it automatically clears the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
7. After checking that the ICSR2.STOP flag is 1, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

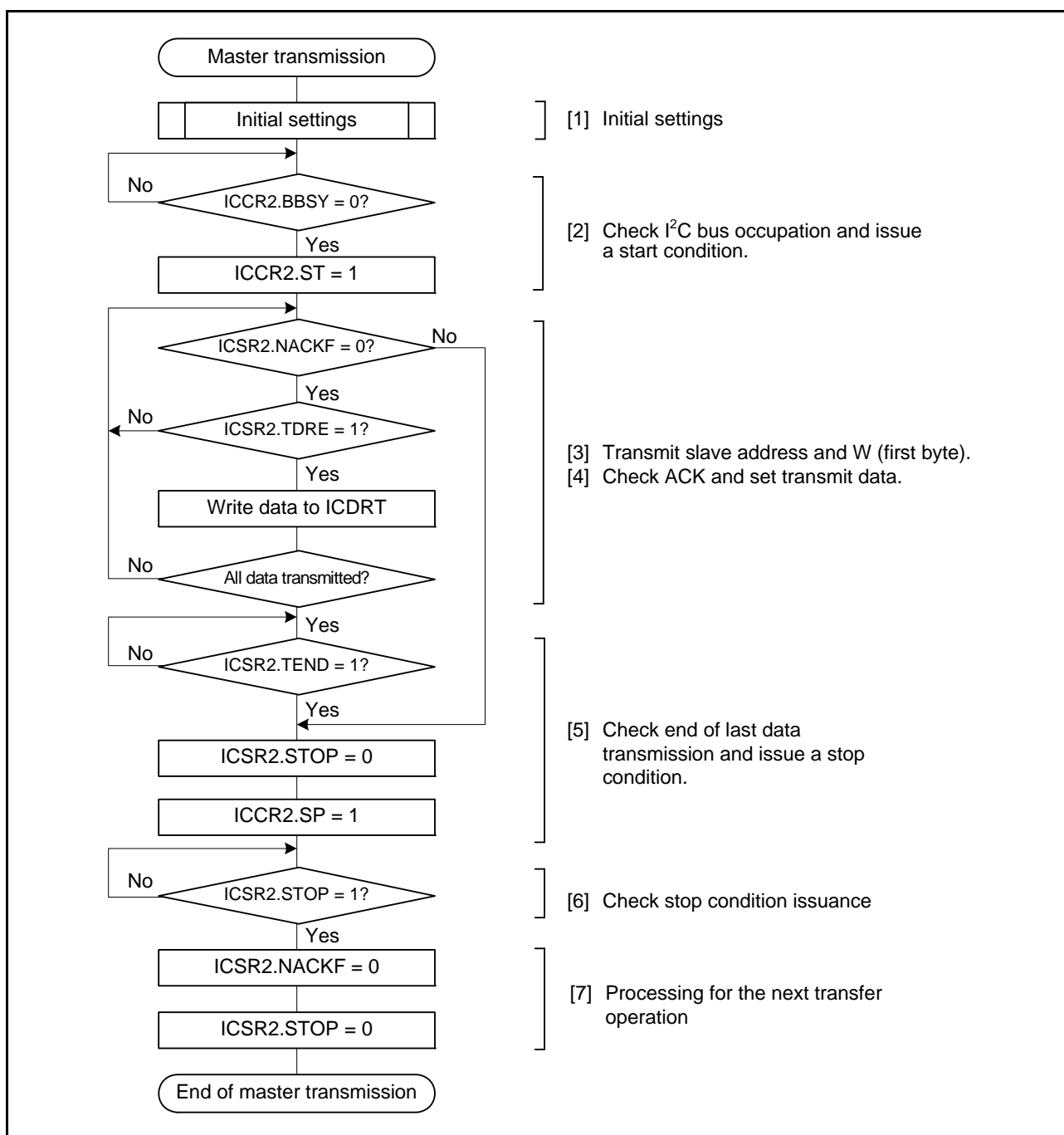


Figure 31.6 Example of Master Transmission Flowchart

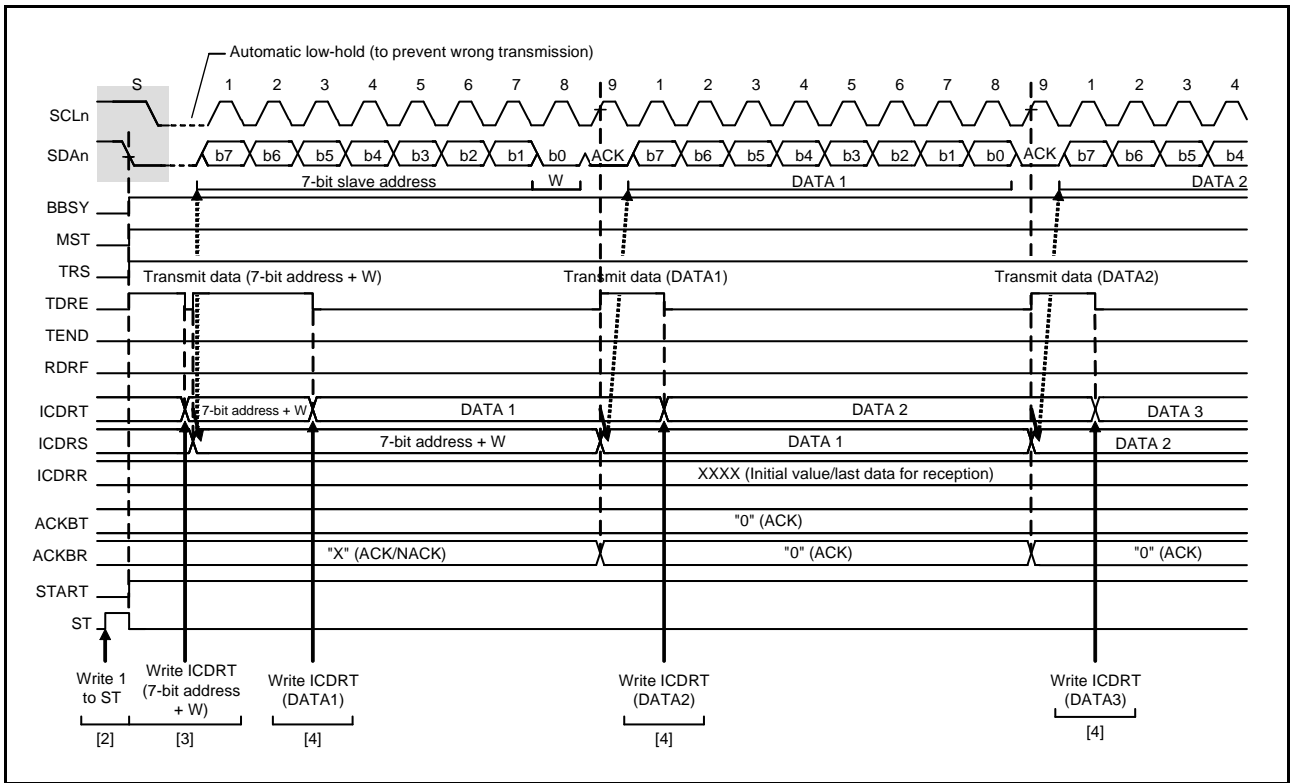


Figure 31.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

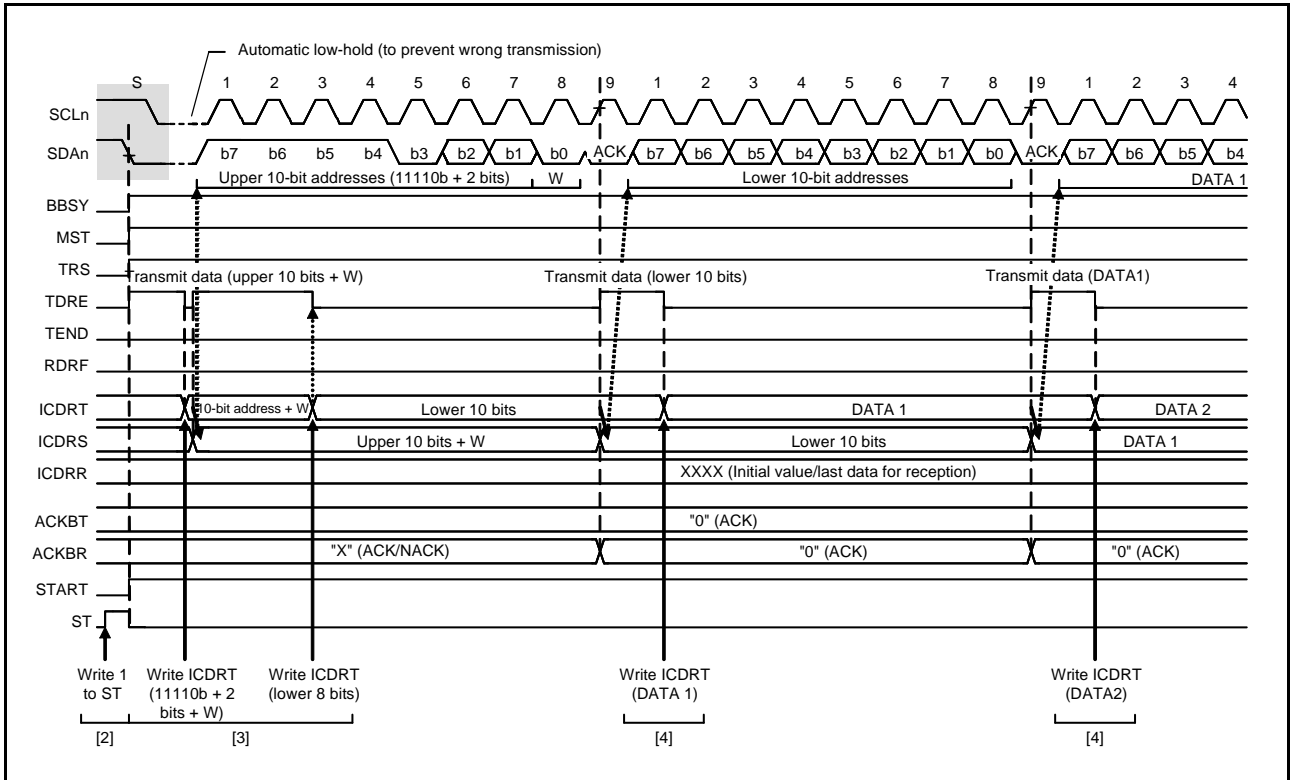


Figure 31.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

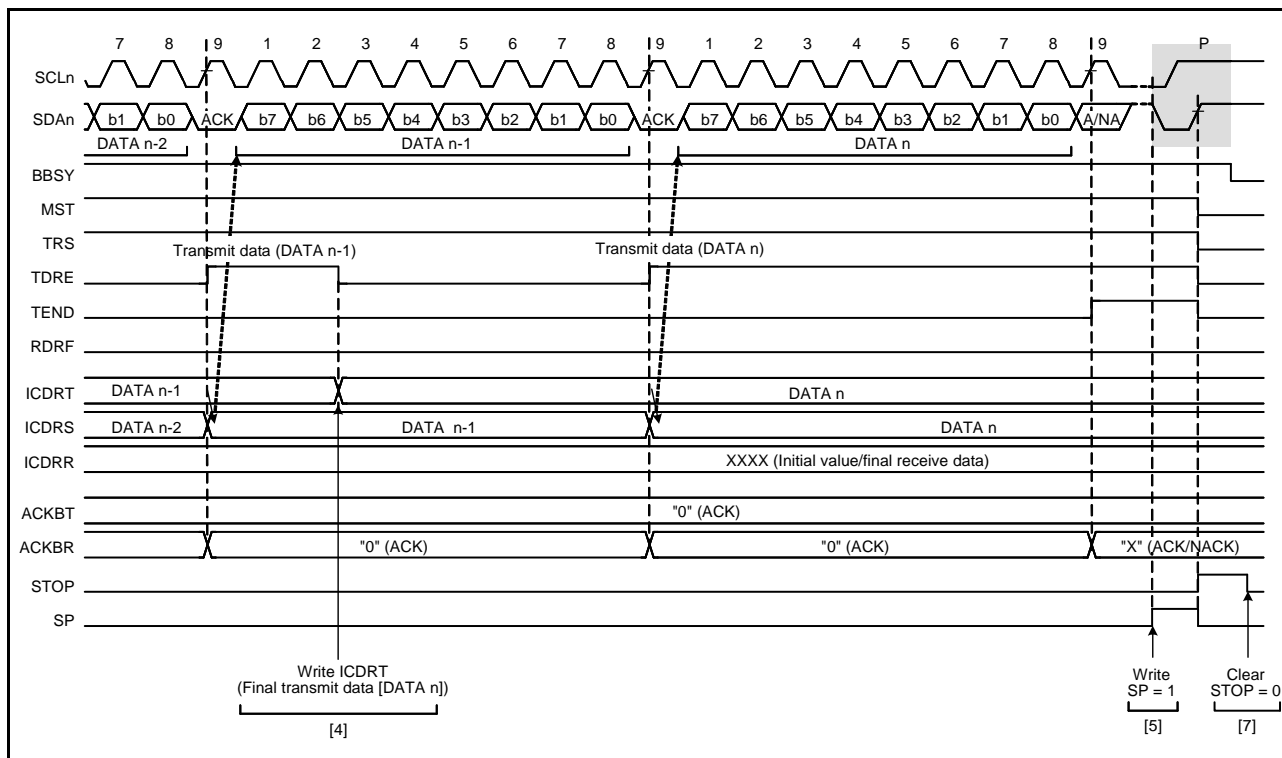


Figure 31.9 Master Transmit Operation Timing (3)

31.3.4 Master Receiver Operation

In master receiver operation, the RIIC as a master device outputs the SCL (clock) signal, receives data from the slave device, and returns acknowledgements. Since the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmitter mode, but the subsequent steps are in master receiver mode.

Figure 31.10 shows an example of usage of master reception and Figure 31.11 and Figure 31.13 show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

1. Set the IICRST bit in ICCR1 to 1 (internal reset) and then clear the IICRST bit to 0 (canceling reset) with the ICE bit in ICCR1 cleared to 0 (disabling the interface). This initializes the internal state and the various flags of ICSR1. After that, set registers SARLn, SARUn, ICSEr, ICMR1, ICBRH, and ICBRL (n = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see Figure 31.5). When the necessary register settings have been completed, set the ICE bit to 1 (to enable transfer). This step is not necessary if initialization of the RIIC has already been completed.
2. Read the BBSY flag in ICCR2 to check that the bus is open, and then set the ST bit in ICCR2 to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. When the RIIC detects the start condition, the BBSY flag and the START flag in ICSR2 are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA_n 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 transmitter mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. Once the data for transmission are written to ICDRT, the TDRE flag is automatically cleared to 0, the data are transferred from ICDRT to ICDRS, and the TDRE flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the ICCR2.TRS bit is automatically updated to select master transmitter or master receiver mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRS bit is cleared to 0 on the rising edge of the ninth cycle of SCL_n (the clock signal), placing the RIIC in master receiver mode. At this time, the TDRE flag is automatically cleared to 0 and the ICSR2.RDRF flag is automatically set to 1. Since the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition. For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the RIIC in master receiver mode.
4. Dummy read ICDRR after confirming that the RDRF flag in ICSR2 is 1; this makes the RIIC start output of the SCL (clock) signal and start data reception.
5. After one byte of data has been received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RDRFS bit in ICMR3. Reading out ICDRR at this time will produce the received data, and the RDRF flag is automatically cleared to 0 at the same time. Furthermore, the value of the acknowledgement field received during the ninth cycle of SCL clock is returned as the value set in the ACKBT bit of ICMR3.

Furthermore, if the next byte to be received is the last but one, set the ICMR3.WAIT bit to 1 (inserting a wait) before reading the ICDRR register (for the second-to-last byte). Doing so ensures that NACK output for the last byte proceeds correctly even if setting of the ICMR3.ACKBT bit to 1 (NACK), described below, is delayed by the reception of another interrupt etc., and that output of the stop condition is possible due to the SCL_n line being fixed to the low level on the falling edge in the ninth clock cycle of reception of the last byte.

6. Set the ICMR3.ACKBT to 1 (NACK) if an indication of clearing of the ICMR3.RDRFS bit to 0 on completion of transfer for the next data reception by the slave device is required.
7. After reading out the byte before last from the ICDRR register, if the value of the ICSR2.RDRF flag is confirmed to be 1, write 1 to the SP bit in ICCR2 (stop condition issuance request) and then read the last byte from ICDRR. When ICDRR is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCL line is released from the low-hold state.
8. Upon detecting the stop condition, the RIIC automatically clears the MST and TRS bits in ICCR2 to "00b" and enters slave receiver mode. Furthermore, detection of the stop condition leads to setting of the STOP flag in ICSR2 to 1.
9. After checking that the ICSR2.STOP flag is 1, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

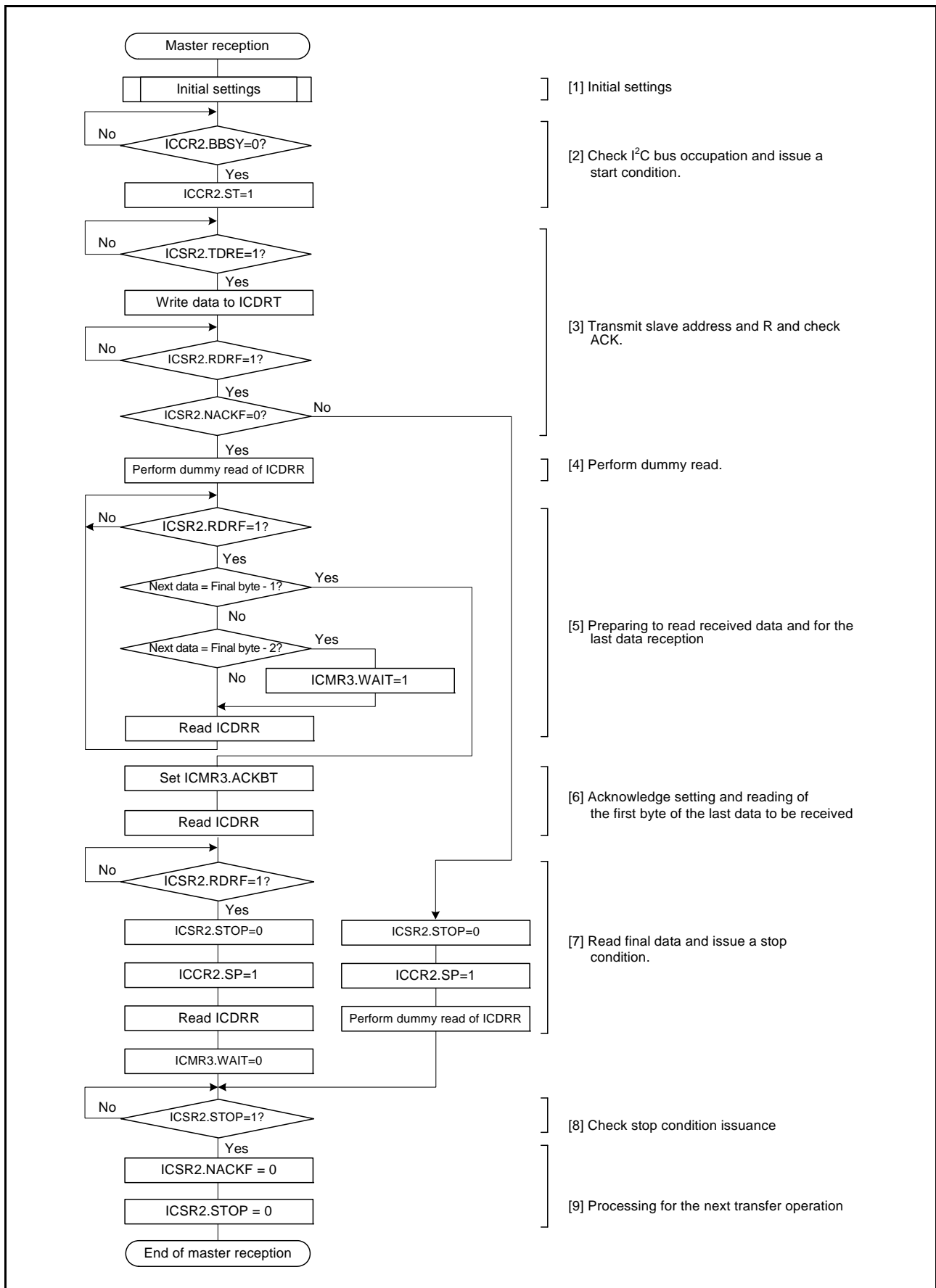


Figure 31.10 Example of Master Reception Flowchart (7-Bit Address Format)

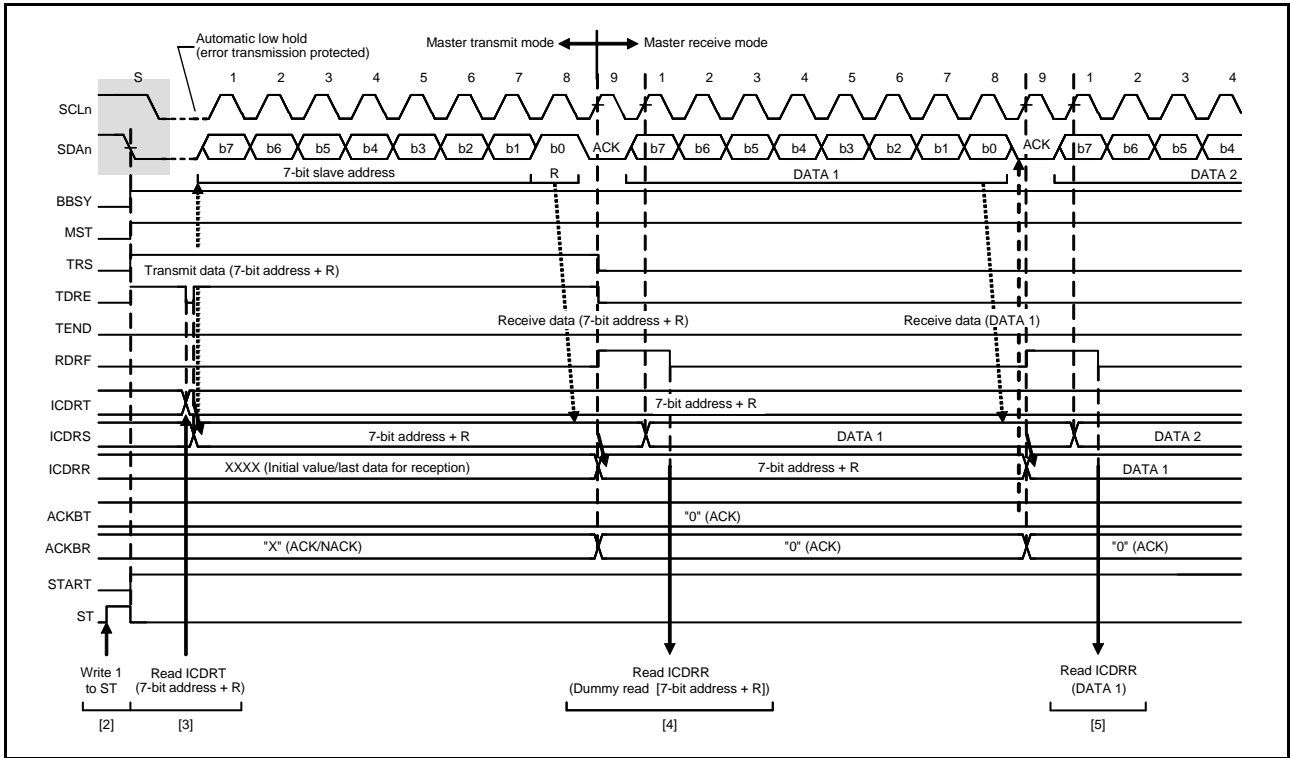


Figure 31.11 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS=0)

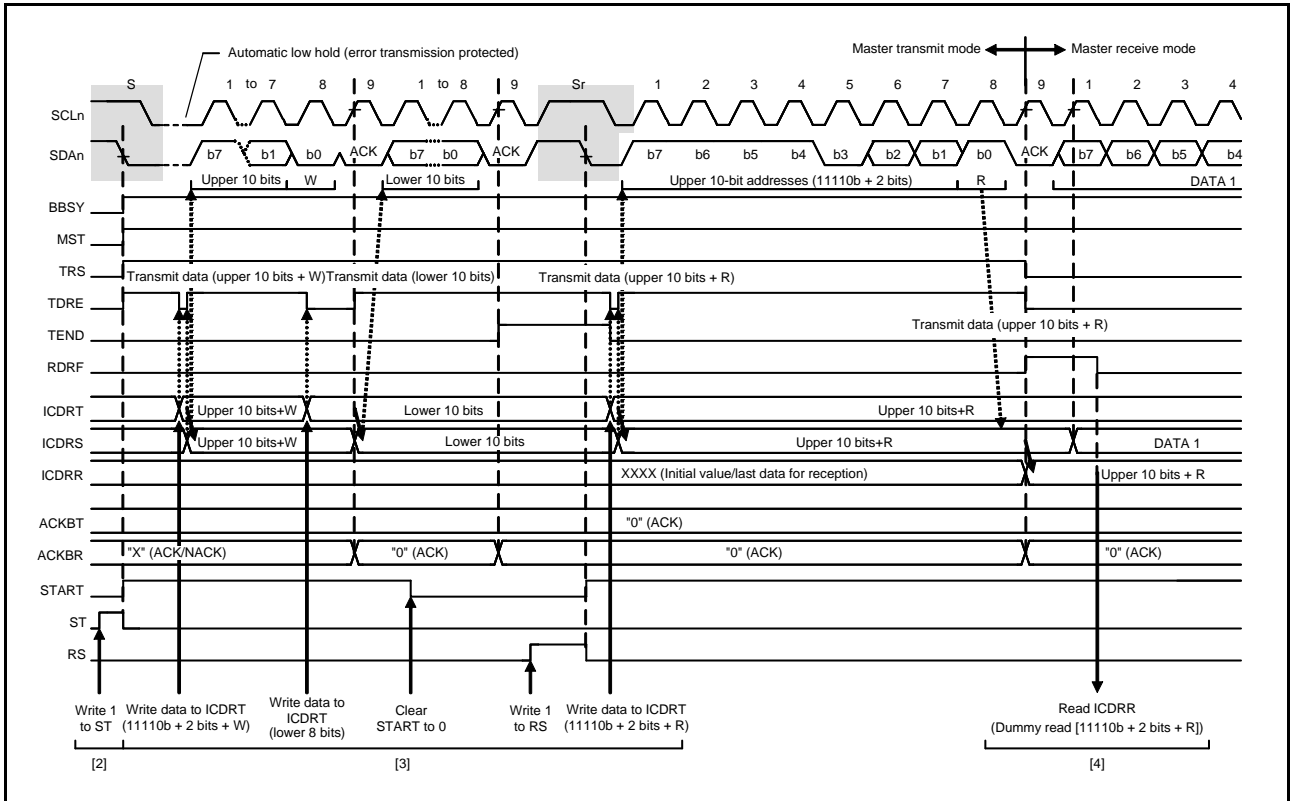


Figure 31.12 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS=0)

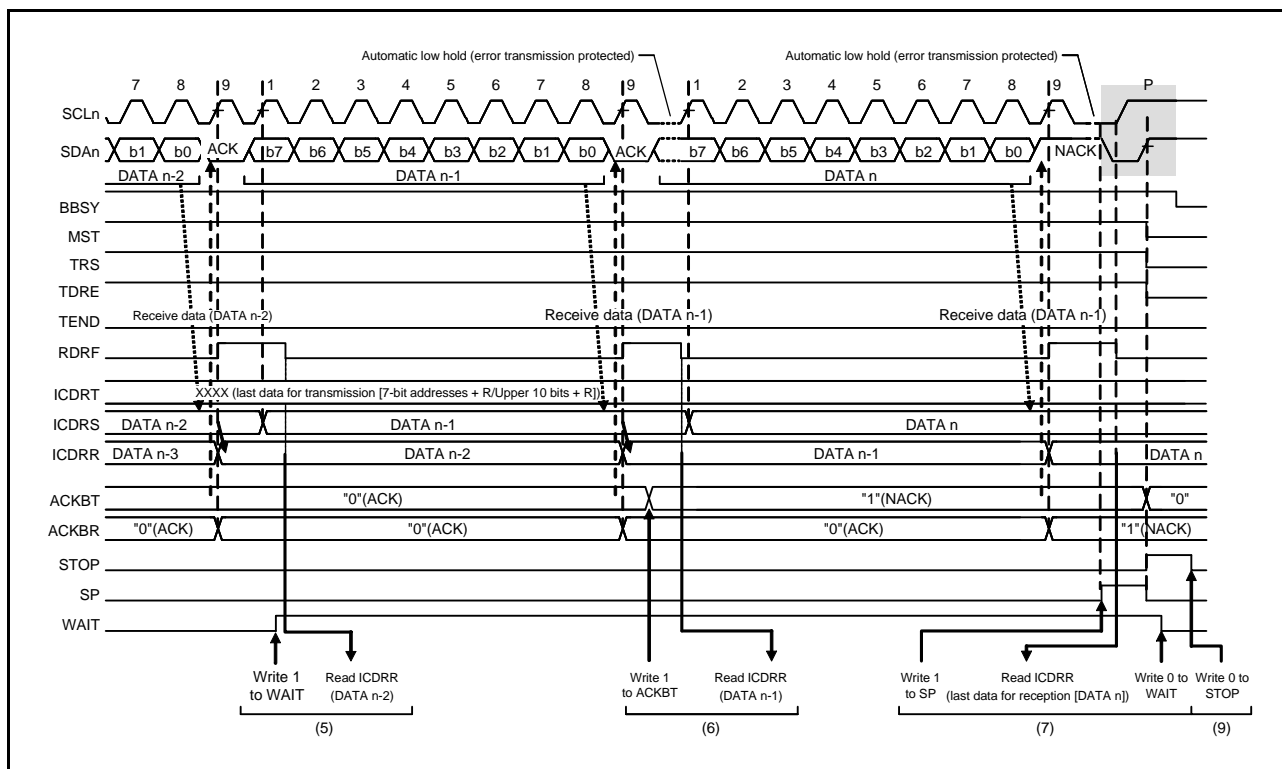


Figure 31.13 Master Receive Operation Timing (3) (when RDRFS=0)

31.3.5 Slave Transmitter Operation

In slave transmitter operation, the master device outputs the SCL (clock) signal, the RIIC transmits data as a slave device, and the master device returns acknowledgements.

Figure 31.14 shows an example of usage of slave transmission and Figure 31.15 and Figure 31.16 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

1. Follow the procedure in Figure 31.5 to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.
2. After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AAS_y (n = 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 transmitter mode by setting both the TRS bit and the TDRE flag in ICSR2 to 1.
3. After the ICSR2.TEND flag is confirmed to be 1, write the data for transmission to the ICDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives a NACK signal) while the ICFER.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 SCL_n 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 SCL_n line.
6. Upon detecting the stop condition, the RIIC automatically clears bits ICSR1.HOA, GCA, and AAS_y (y = 0 to 2), flags ICSR2.TDRE and TEND, and the ICCR2.TRS bit to 0, and enters slave receiver mode.
7. After checking that the ICSR2.STOP flag is 1, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

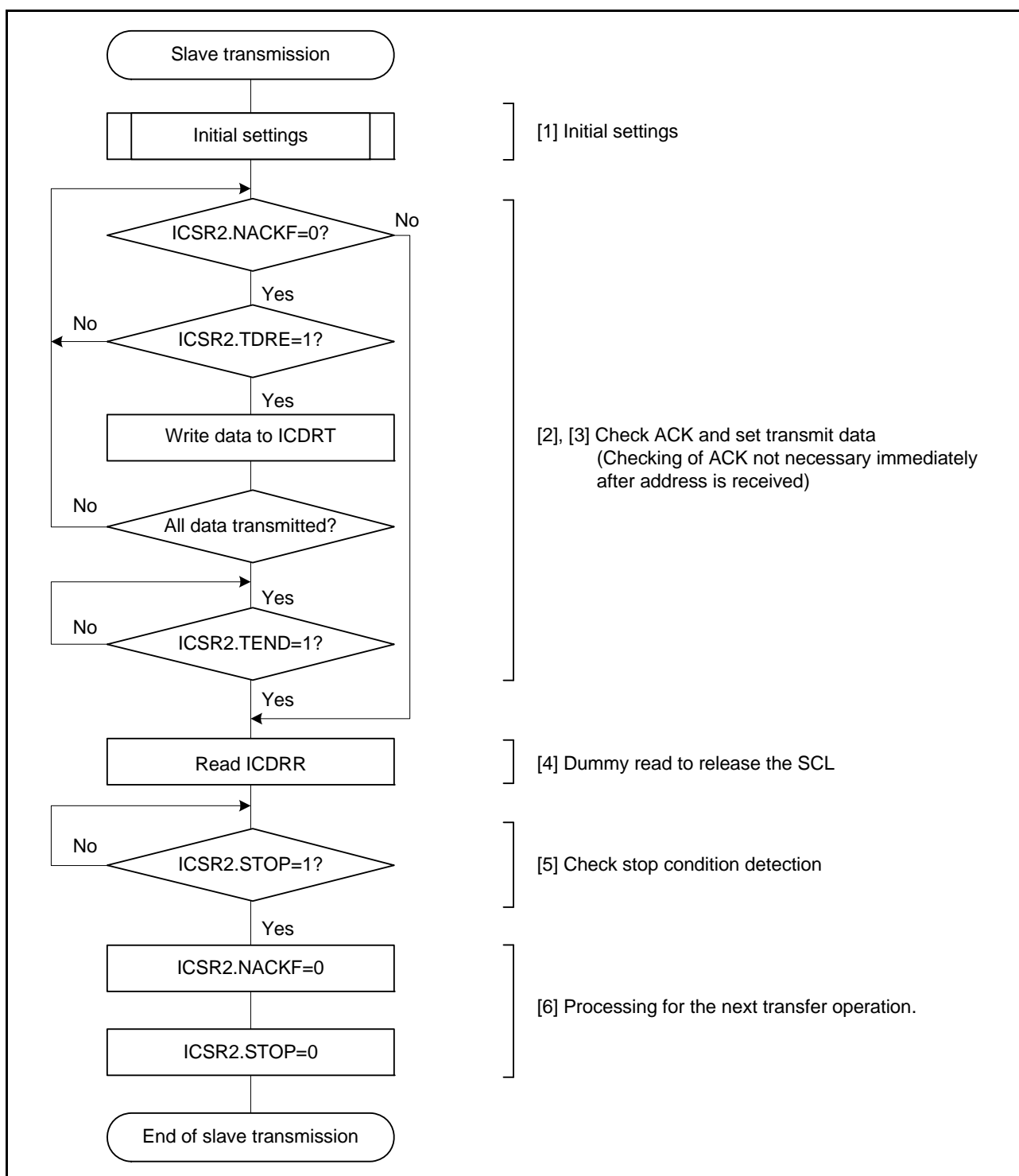


Figure 31.14 Example of Slave Transmission Flowchart

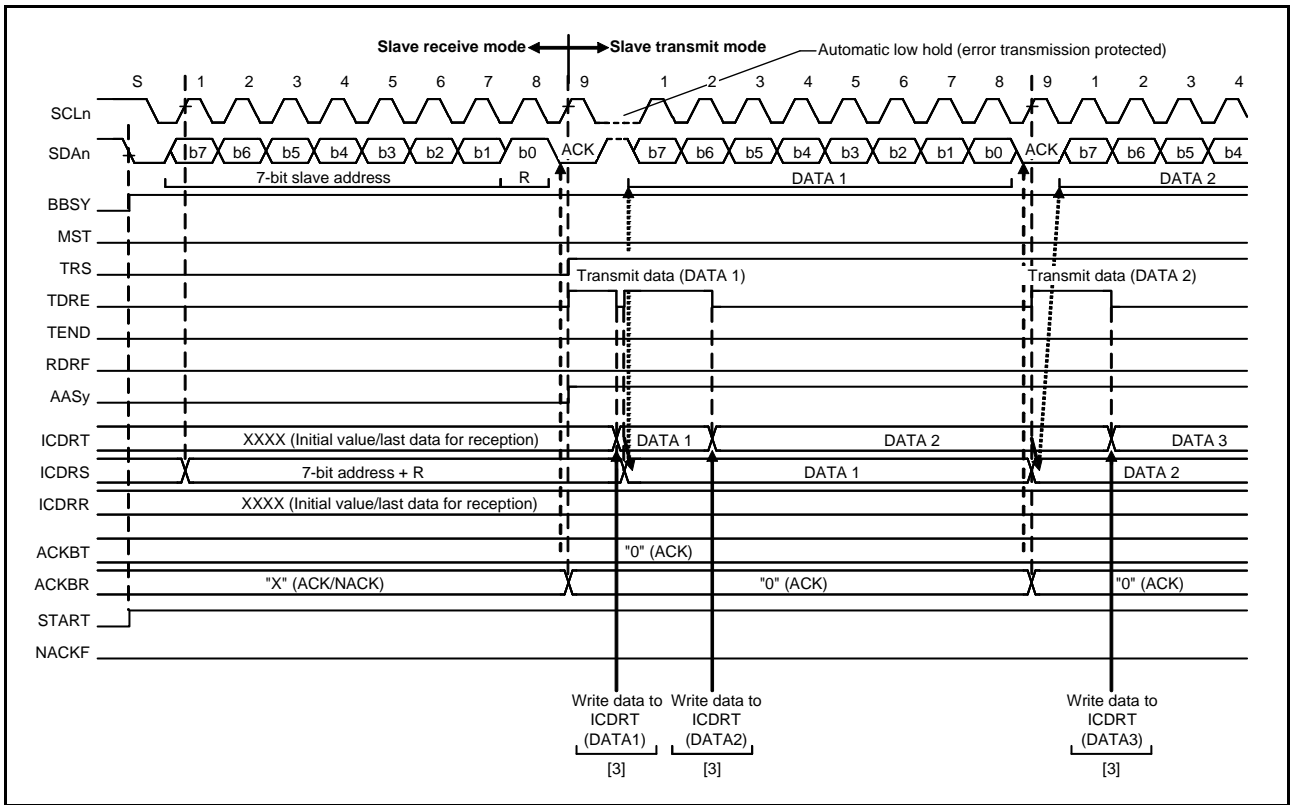


Figure 31.15 Slave Transmit Operation Timing (1) (7-Bit Address Format)

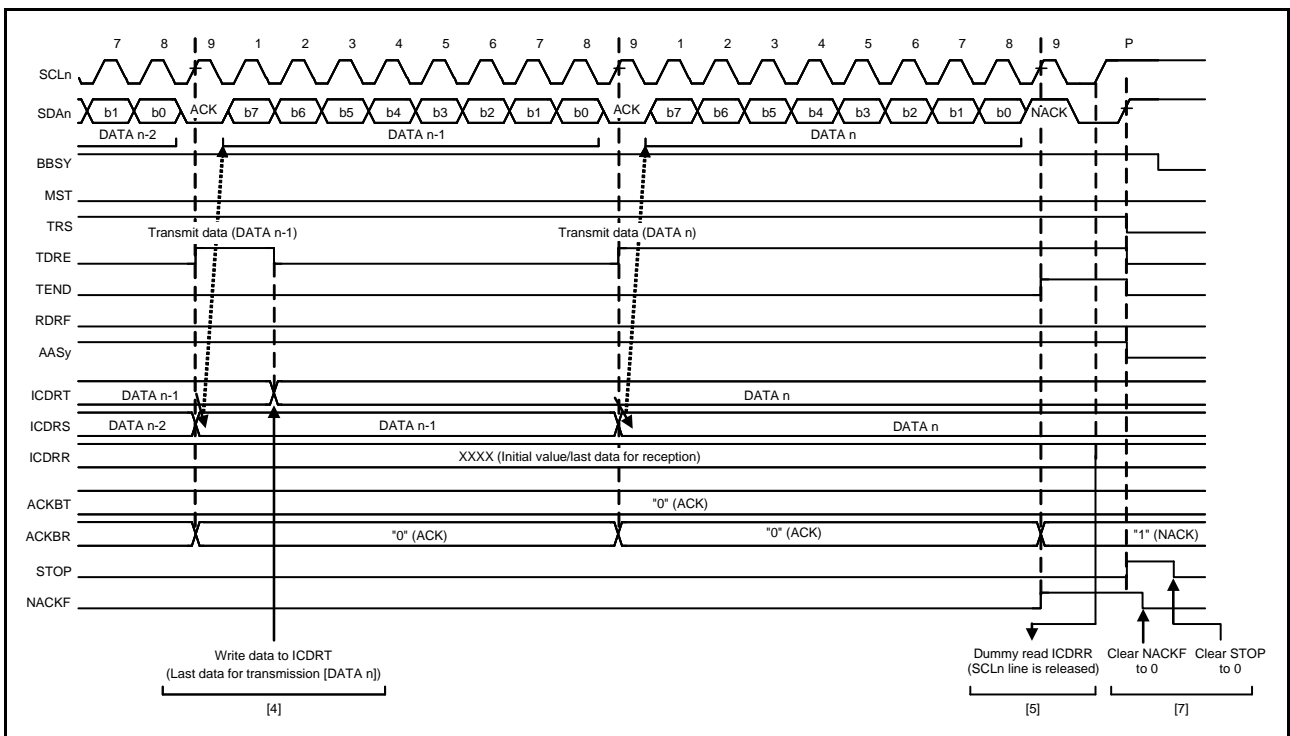


Figure 31.16 Slave Transmit Operation Timing (2)

31.3.6 Slave Receiver Operation

In slave receiver operation, the master device outputs the SCL clock and data, and the RIIC returns acknowledgements as a slave device.

Figure 31.17 shows an example of usage of slave reception and Figure 31.18 and Figure 31.19 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

1. Follow the procedure in Figure 31.5 to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.
2. After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AAS_y (n = 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 continues to place itself in slave receiver mode and sets the RDRF flag in ICSR2 to 1.
3. After the ICSR2.STOP flag is confirmed to be 0 and the ICSR2.RDRF flag to be 1, dummy read ICDRR (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower eight bits when the 10-bit address format is selected).
4. When ICDRR is read, the RIIC automatically clears the ICSR2.RDRF flag to 0. If reading of ICDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL_n line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading ICDRR releases the SCL_n 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_y (y = 0 to 2) to 0.
6. After checking that the ICSR2.STOP flag is 1, clear the ICSR2.STOP flag to 0 for the next transfer operation.

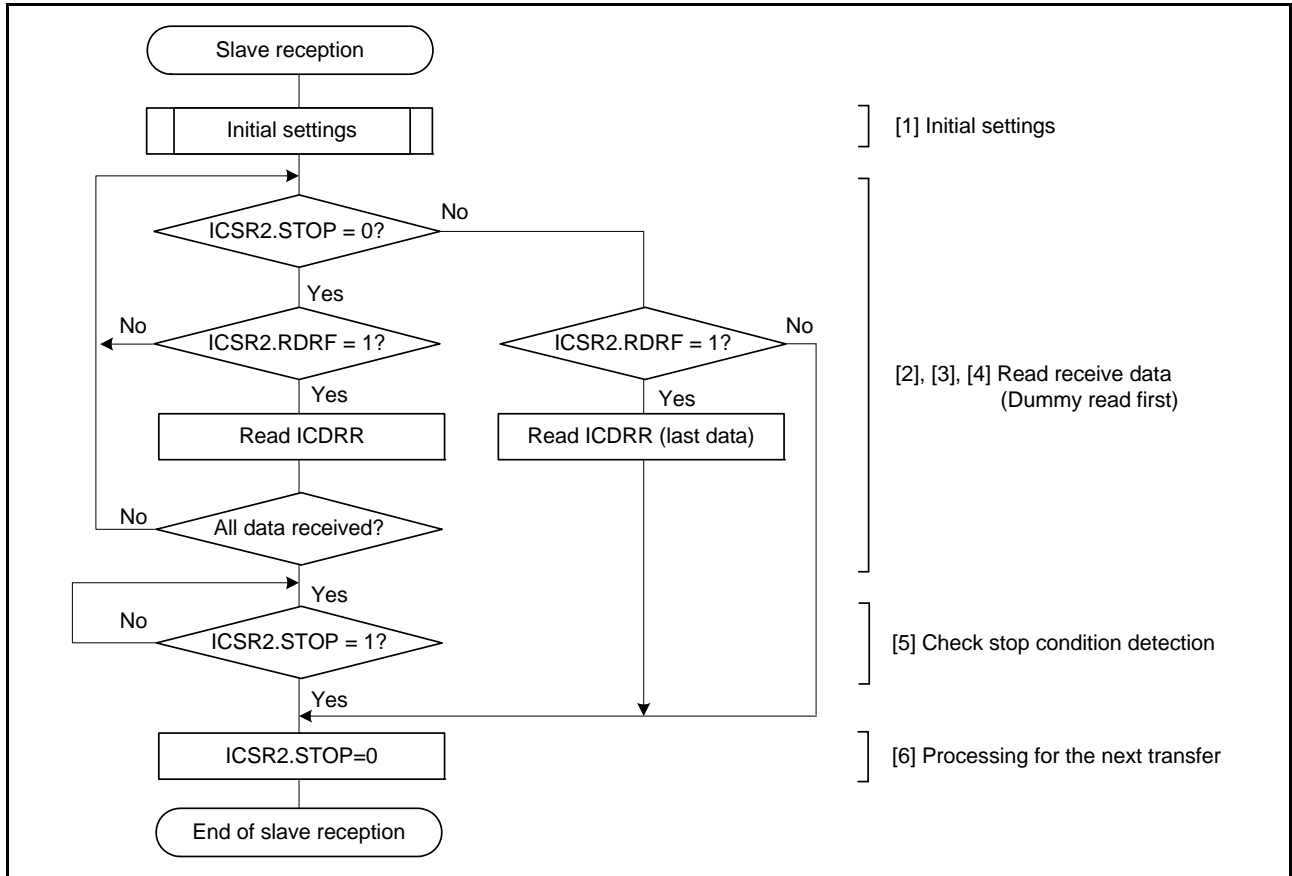


Figure 31.17 Example of Slave Reception Flowchart

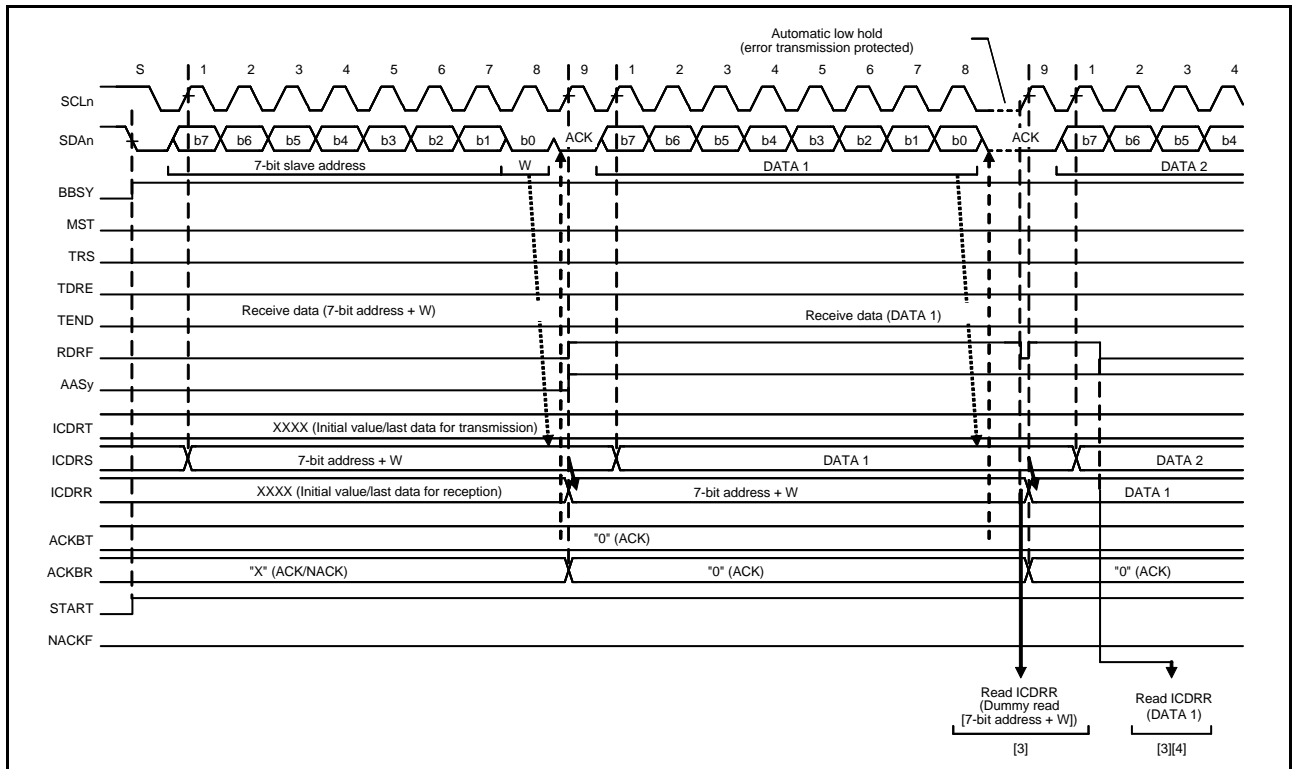


Figure 31.18 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS=0)

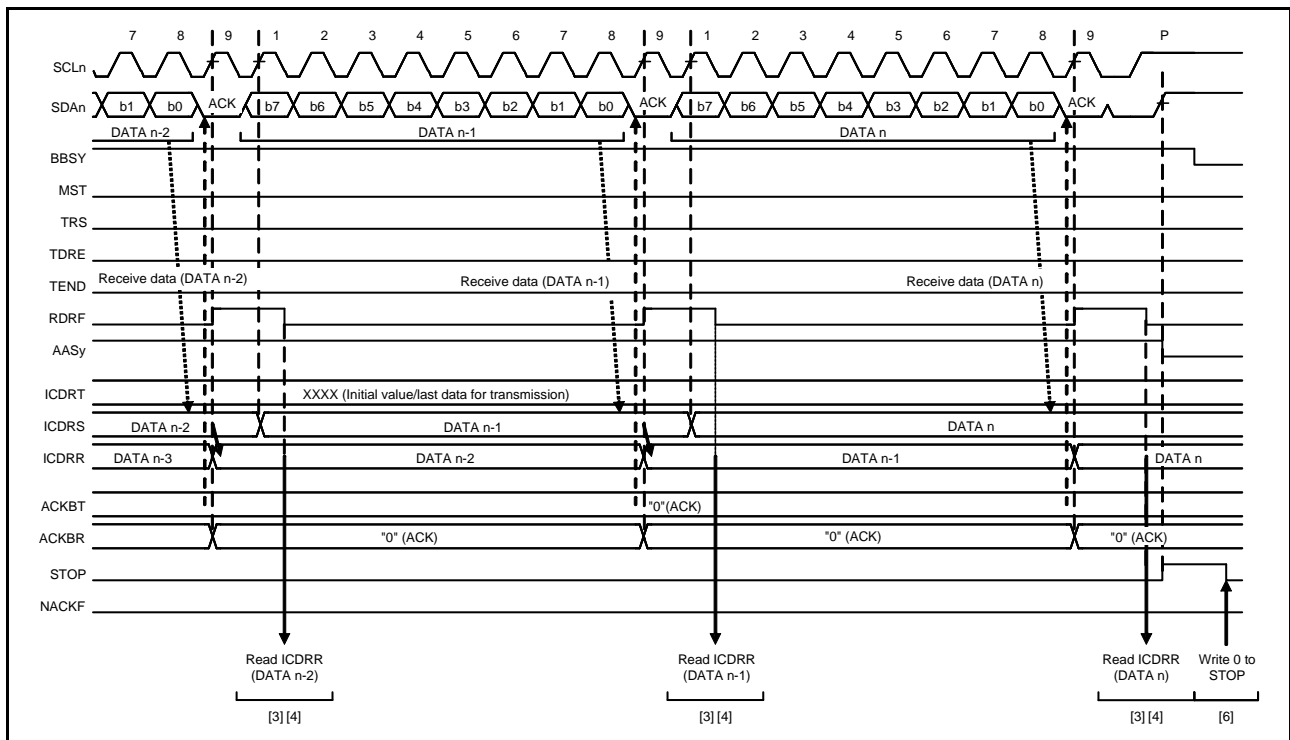


Figure 31.19 Slave Receive Operation Timing (2) (when RDRFS=0)

31.4 SCL Synchronization Circuit

In generation of the SCL (clock) signal, the RIIC starts counting out the value for width at high level specified in ICBRH when it detects a rising edge on the SCLn line and drives the SCLn line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCLn line, it starts counting out the width at low level period specified in ICBRL, and then stops driving the SCLn line (releases the line) once counting of the width at low level is complete. The SCL (clock) signal is thus generated.

If multiple master devices are connected to the I²C bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When the RIIC has detected a rising edge on the SCLn line and thus started counting out the width at high level specified in ICBRH, and the level on the SCL 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 SCL line low, and starts counting out the width at low level specified in ICBRL. When the RIIC finishes counting out the width at low level, it stops driving the SCLn line to the low level (i.e. releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL line has been released. When the RIIC finishes outputting the low-level period of the SCL clock of, the SCLn line is released and the SCL clock rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the SCLE bit in ICFER is set to 1.

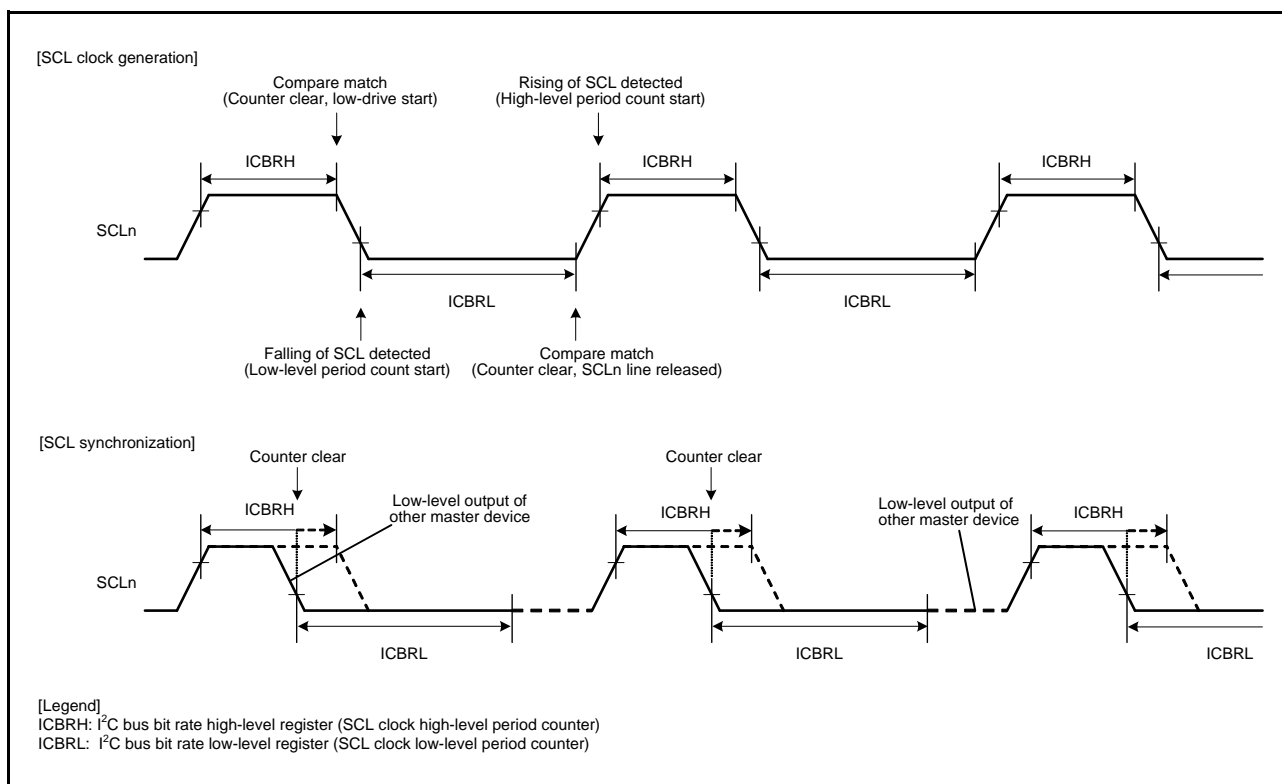


Figure 31.20 Generation and Synchronization of the SCL Signal from the RIIC

31.5 Facility for Delaying SDA Output

The RIIC module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL (clock) signal is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices, with the aim of satisfying the 300-ns (min.) data-hold time requirement of the SMBus specification.

The output delay facility is enabled by setting the SDDL[2:0] bits in IMCR2 to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay facility is enabled (i.e. while the SDDL[2:0] bits in IMCR2 are set to any value other than 000b), the DLCS bit in IMCR2 selects the clock source for counting by the SDA output delay counter as the internal base clock (IICφ) for the RIIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IICφ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in IMCR2. After counting of the set number of cycles of delay is completed, the RIIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.

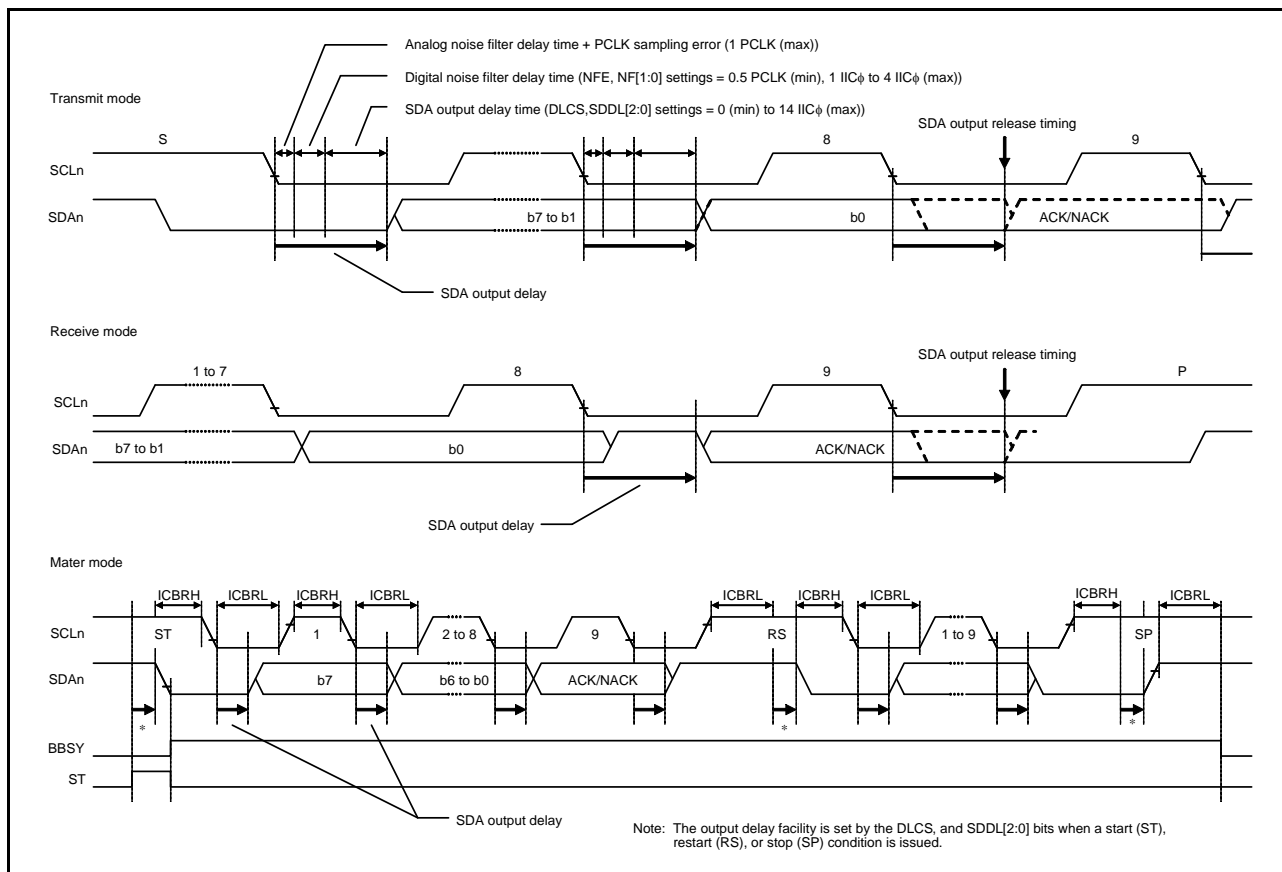


Figure 31.21 SDA Output Delay Facility

31.6 Digital Noise-Filter Circuits

The states of the SCLn and SDAn pins are conveyed to the internal circuitry through analog noise-filter and digital noise-filter circuits. Figure 31.22 is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the RIIC consists of four flip-flop circuit stages connected in series and a match-detection circuit.

The number of effective stages in the digital noise filter is selected by the NF[1:0] bits in ICMR3. The selected number of effective stages determines the noise-filtering capability as a period from one to four IIC ϕ cycles.

The input signal to the SCLn pin (or SDAn pin) is sampled on falling edges of the PCLK signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the NF[1:0] bits in ICMR3, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is small (e.g. data transfer at 400 kbps with PCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, it is possible to disable the digital noise-filter circuit (by clearing the NFE bit in ICFER) and use only the analog noise-filter circuit.

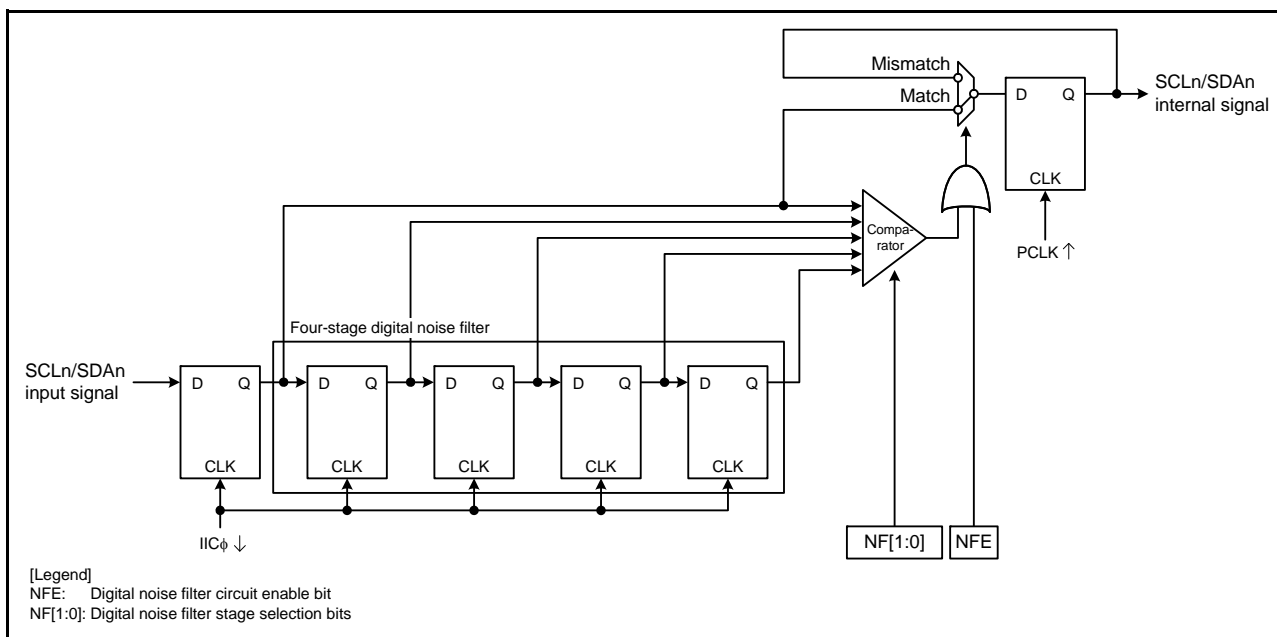


Figure 31.22 Block Diagram of Digital Noise Filter Circuit

31.7 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.

31.7.1 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 SARnE bit (n = 0 to 2) in ICSER is set to 1, the slave addresses set in SARUn and SARLn (n = 0 to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding AASy flag (n = 0 to 2) in ICSR1 is set to 1 at the falling edge of the ninth SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the following R/W# bit. This causes a receive data full interrupt (ICRXI) or transmit data empty interrupt (ICTXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 31.23 to Figure 31.25 show the AASy (y = 0 to 2) flag set timing in three cases.

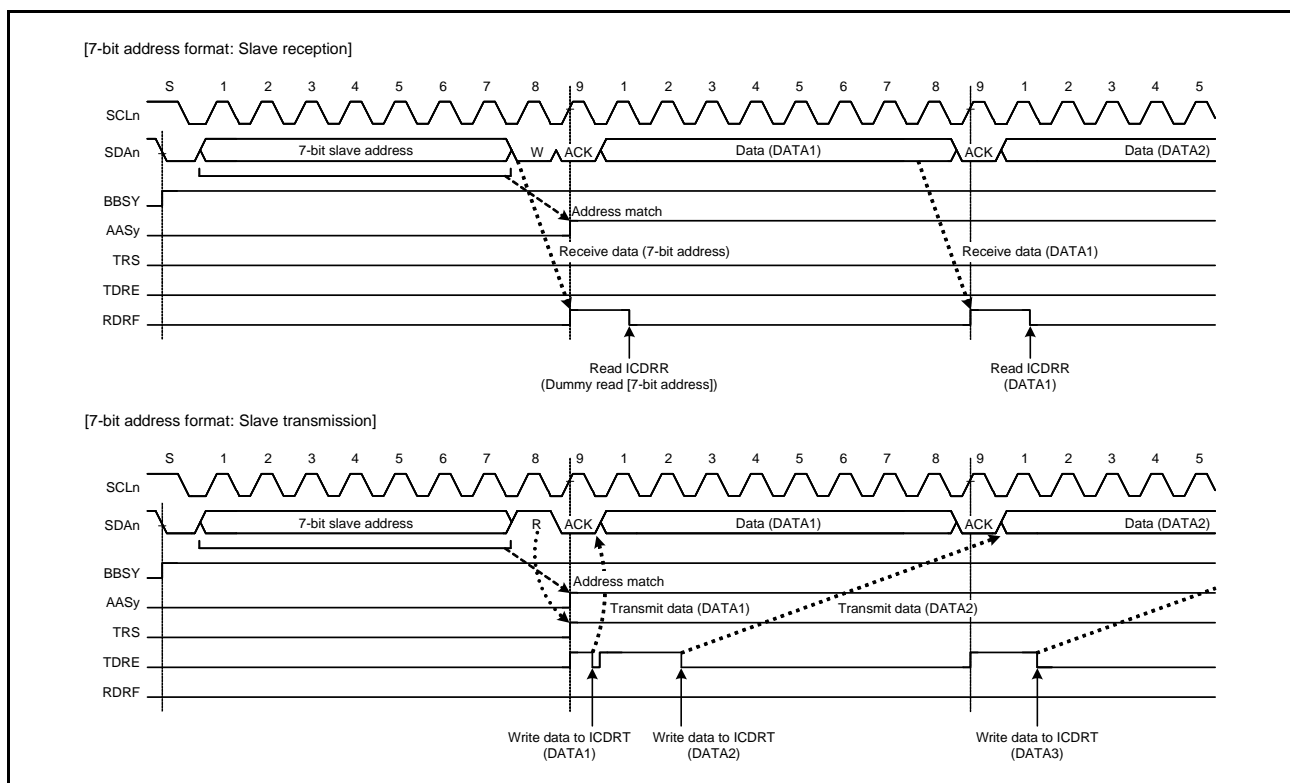


Figure 31.23 AASy Flag Set Timing with 7-Bit Address Format Selected

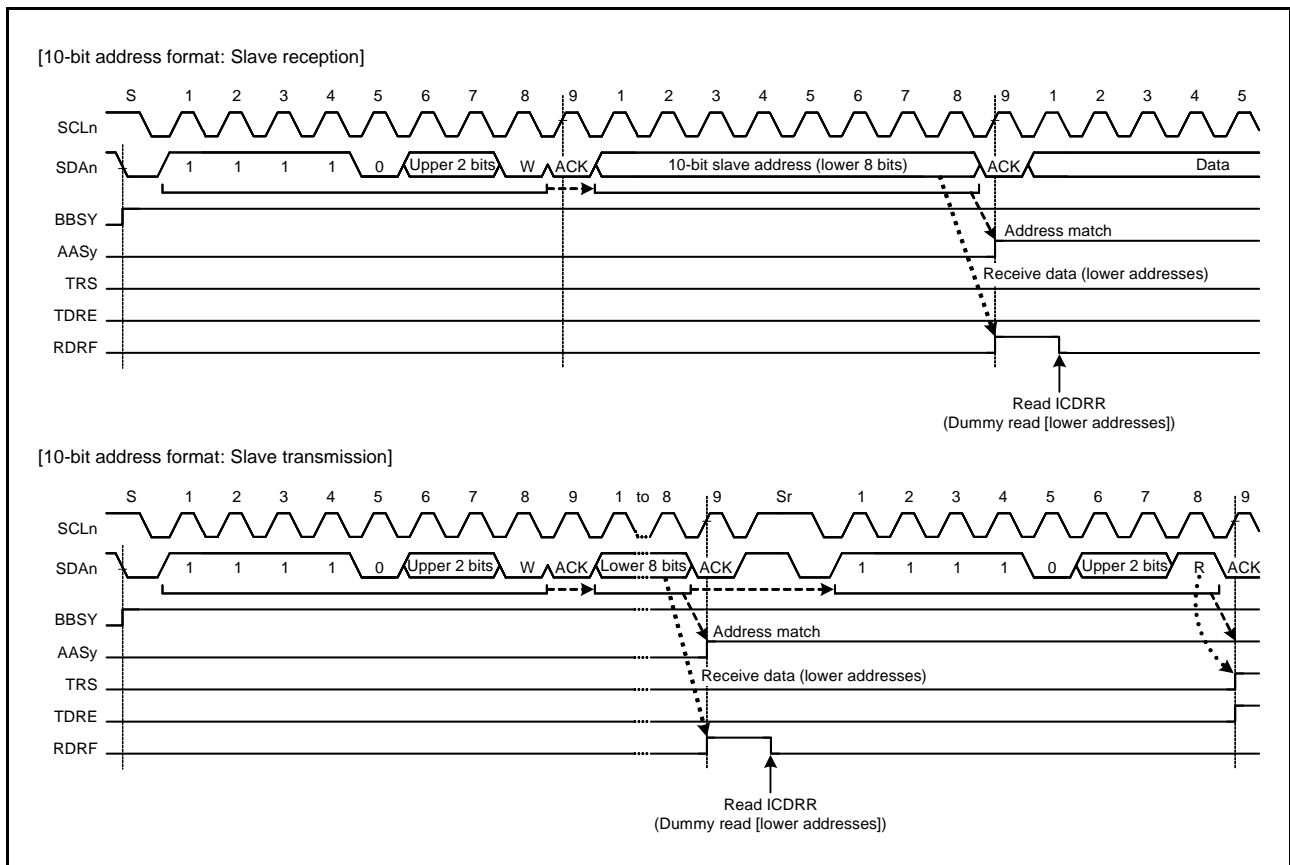


Figure 31.24 AASy Flag Set Timing with 10-Bit Address Format Selected

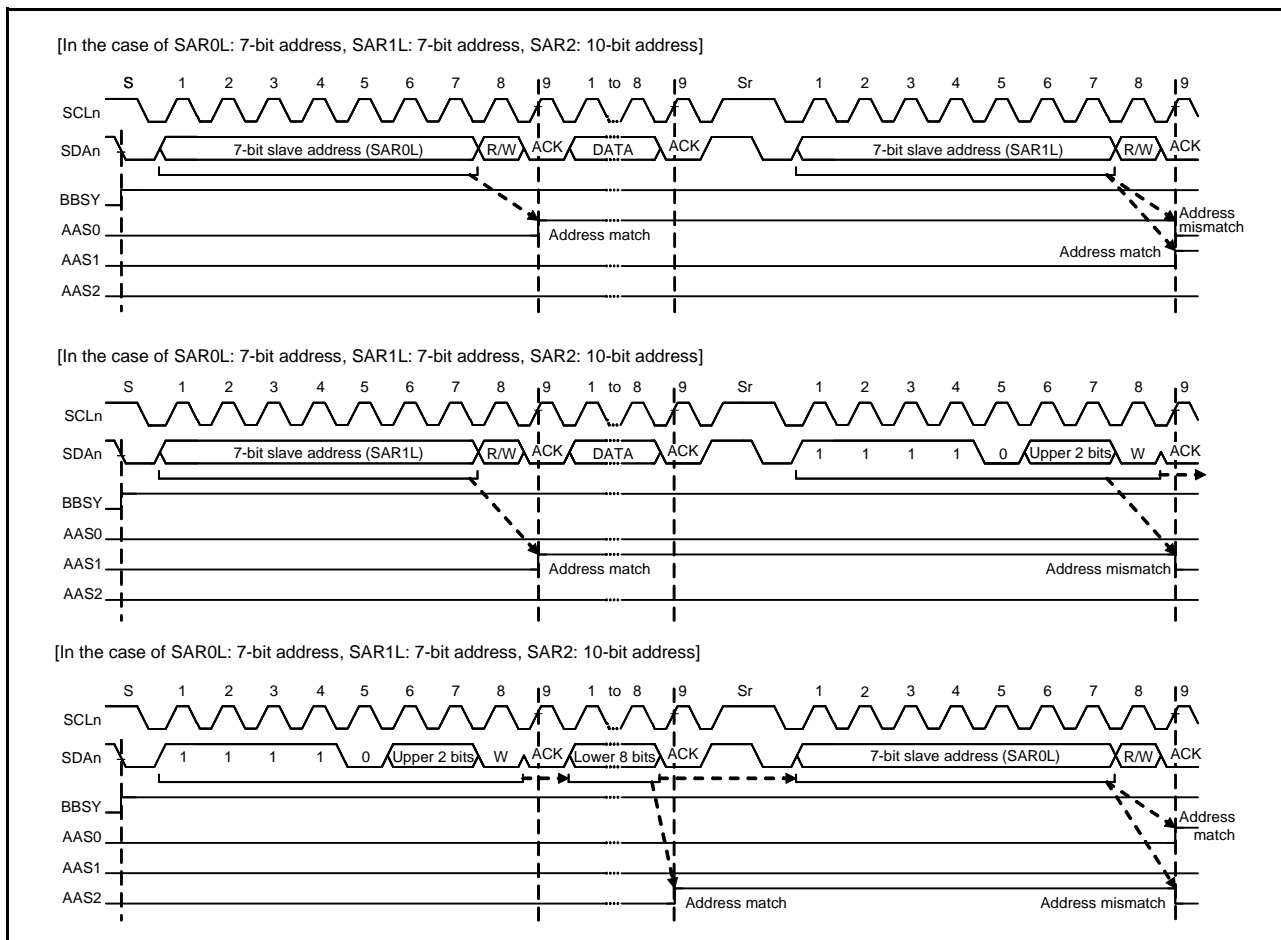


Figure 31.25 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

31.7.2 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 ICSR1 and the RDRF flag in ICSR2 are set to 1 on the falling edge of the ninth cycle of SCL clock. This leads to the generation of a receive data full interrupt (ICRXI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

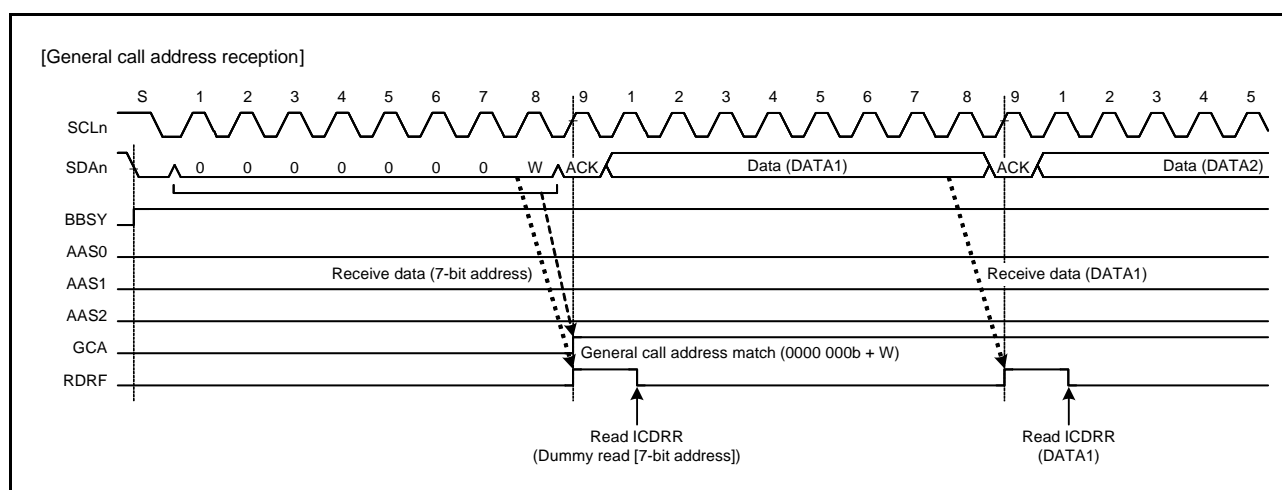


Figure 31.26 Timing of GCA Flag Setting during Reception of General Call Address

31.7.3 Device-ID Address Detection

The RIIC module has a facility for detecting device-ID addresses conformant with the I²C bus standard (Rev. 03). When the RIIC receives 1111 100b as the first byte after a start condition or restart condition was issued with the DIDE bit in ICSER set to 1, the RIIC recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the ninth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding AASy flag (y = 0 to 2) in ICSR1 to 1.

After that, when the first byte received after a restart condition is issued matches the device ID address (1111 100b) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

In the device-ID address detection function, the RIIC clears the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Furthermore, prepare the device-ID fields (three bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

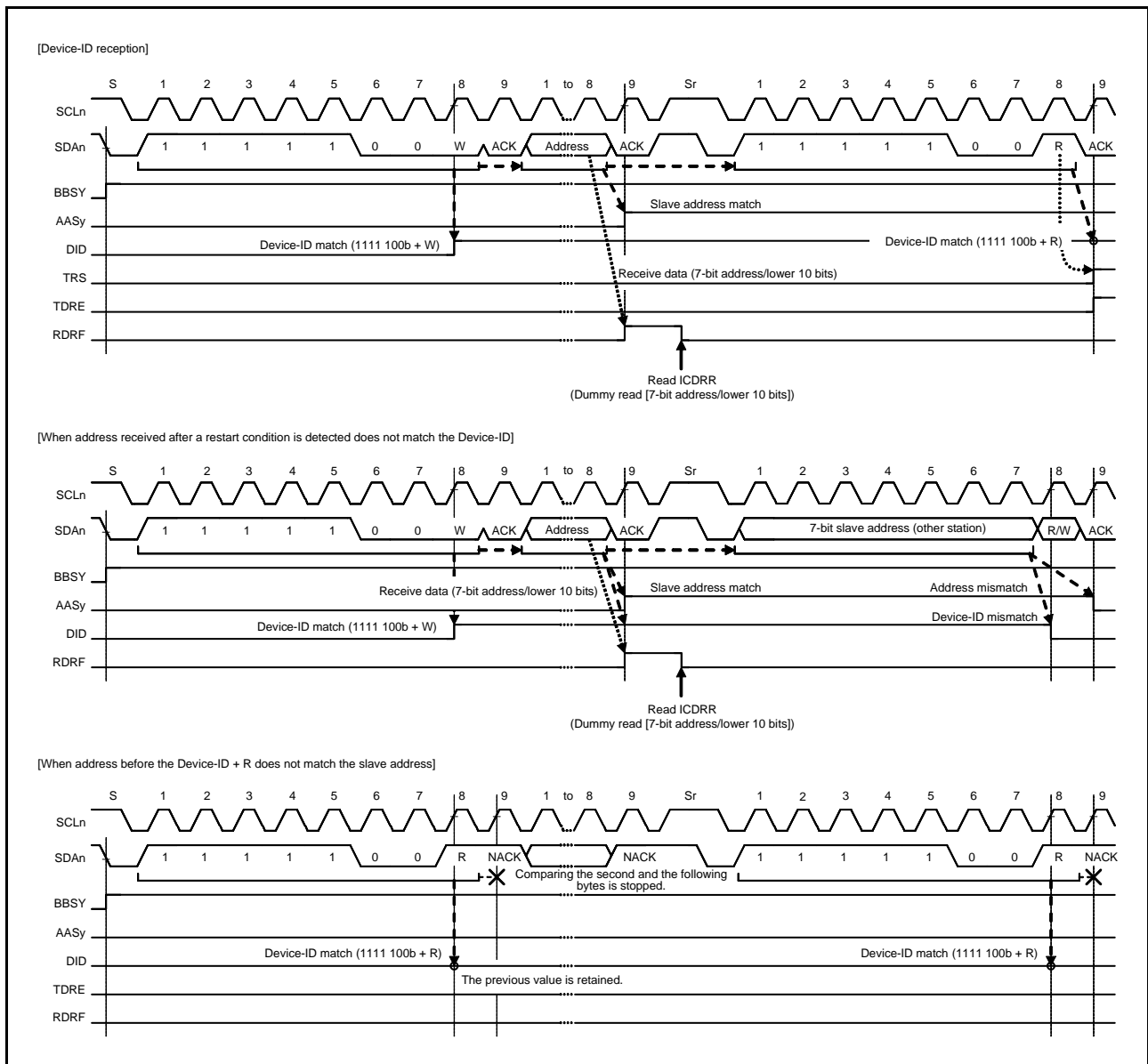


Figure 31.27 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

31.7.4 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 falling edge of the ninth SCL clock cycle, and at the same time, the TDRE flag in IC SR2 is set to 1 when the R/W# bit is 0 (Wr bit). This causes a transmit data empty interrupt (ICTXI) 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.

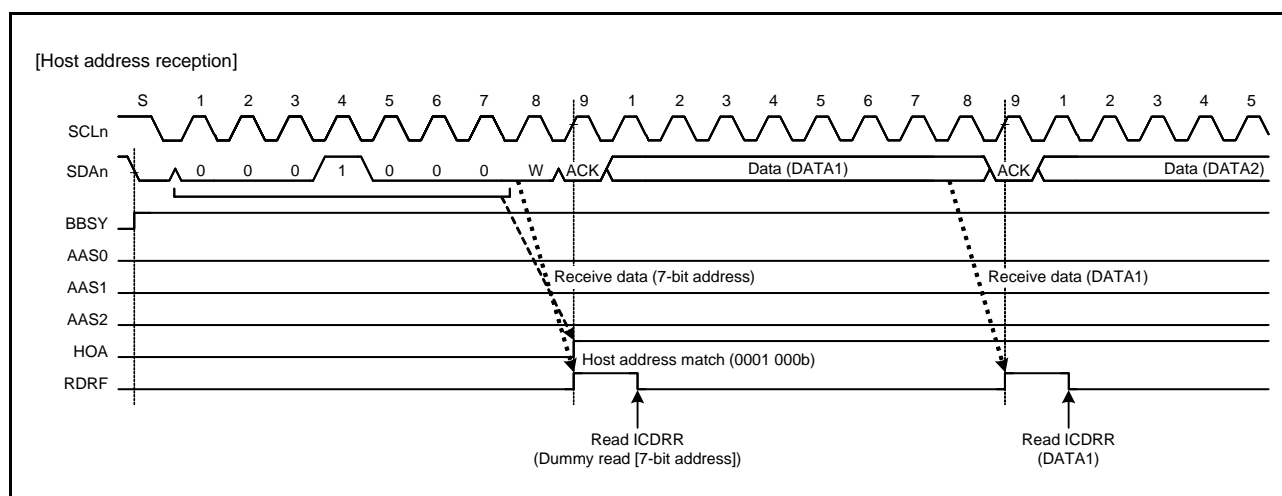


Figure 31.28 HOA Flag Set Timing during Reception of Host Address

31.8 Function to Automatically Hold SCLn Clock Low

31.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (ICDRS) is empty when data have not been written to the transmit data register (ICDRT) with the RIIC in transmission mode (TRS bit = 1 in ICCR2), the SCLn signal is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

<Master transmitter mode>

- Low-level interval after a start condition or restart condition is issued
- Low-level interval of one clock cycle between the ninth clock cycle of one transfer and the first clock cycle of the next

<Slave transmitter mode>

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

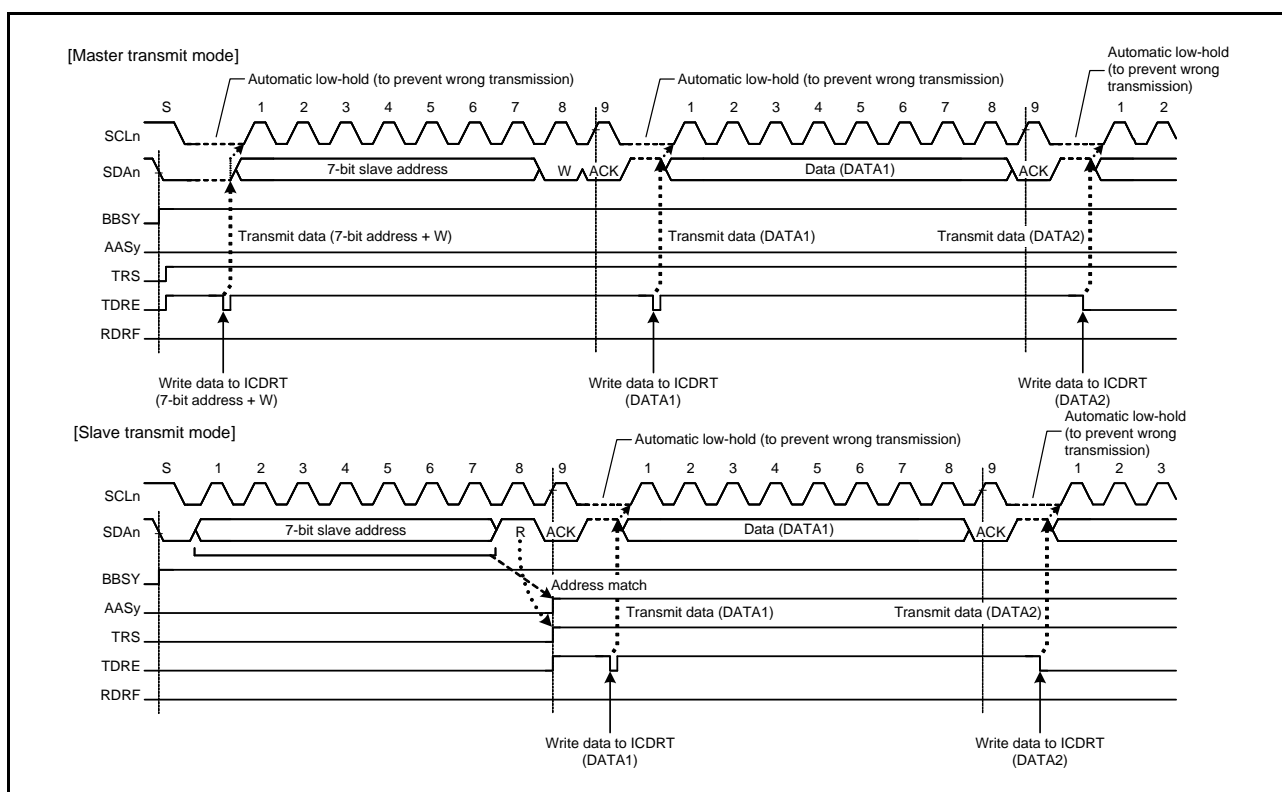


Figure 31.29 Automatic Low-Hold Operation in Transmit Mode

31.8.2 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 SDA_n line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF flag = 1 in ICSR2), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to clear the NACKF flag to 0. In master transmit mode, clear the NACKF flag to 0, issue a restart or stop condition, and then issue a start condition again.

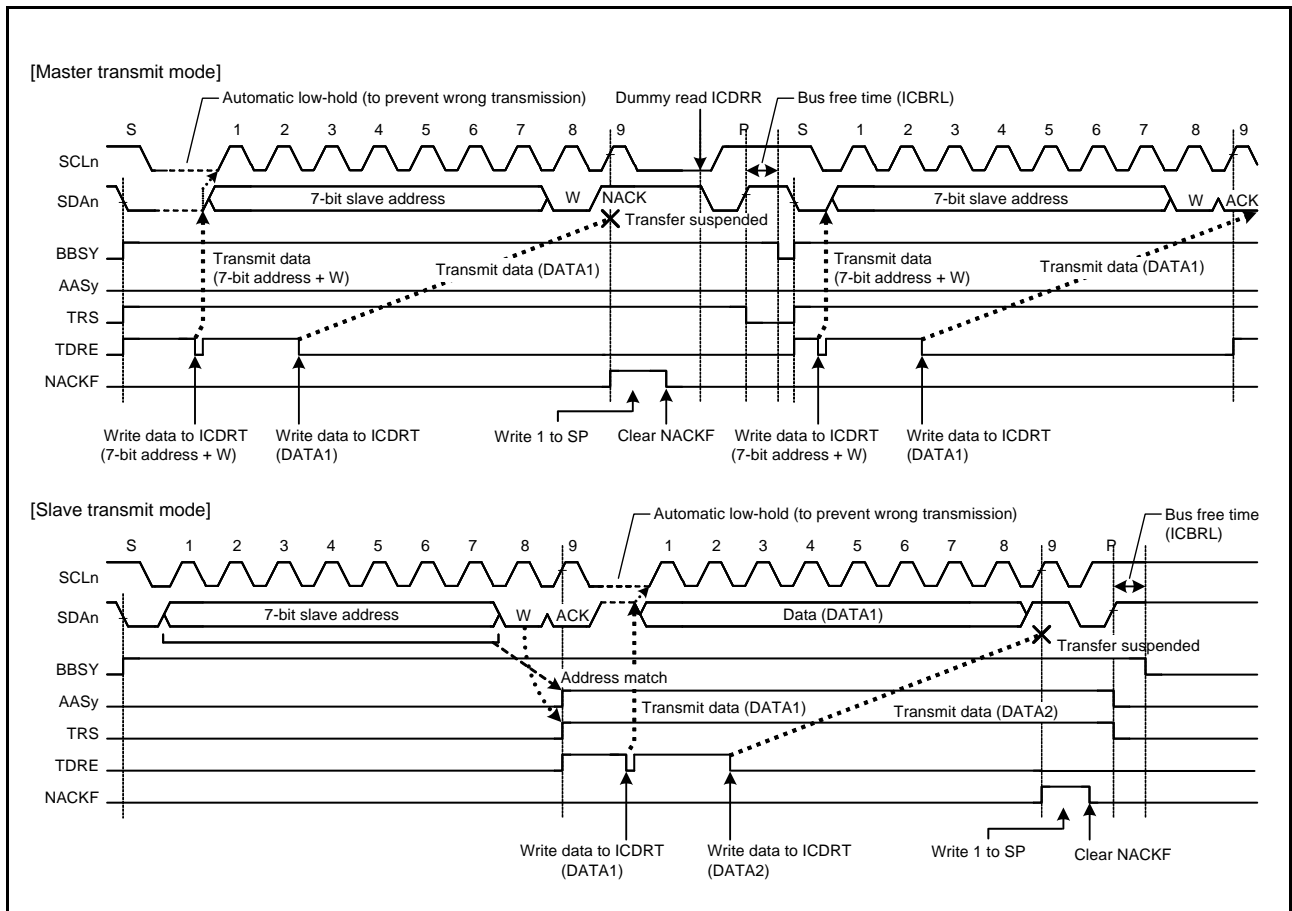


Figure 31.30 Suspension of Data Transfer when NACK is Received (NACKE = 1)

31.8.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the RIIC holds the SCLn line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is issued. This function does not disturb other communication because the RIIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

(1) One-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the WAIT bit in ICMR3 is set to 1, the RIIC performs one-byte receive operation using the WAIT bit function. Furthermore, when the ICMR3.RDRFS bit is 0, the RIIC automatically sends the ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCLn line low at the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRR, which enables 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) One-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the RDRFS bit in ICMR3 is set to 1, the RIIC performs one-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the RDRF flag (receive data full) in ICSR2 is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCLn line is automatically held low at the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

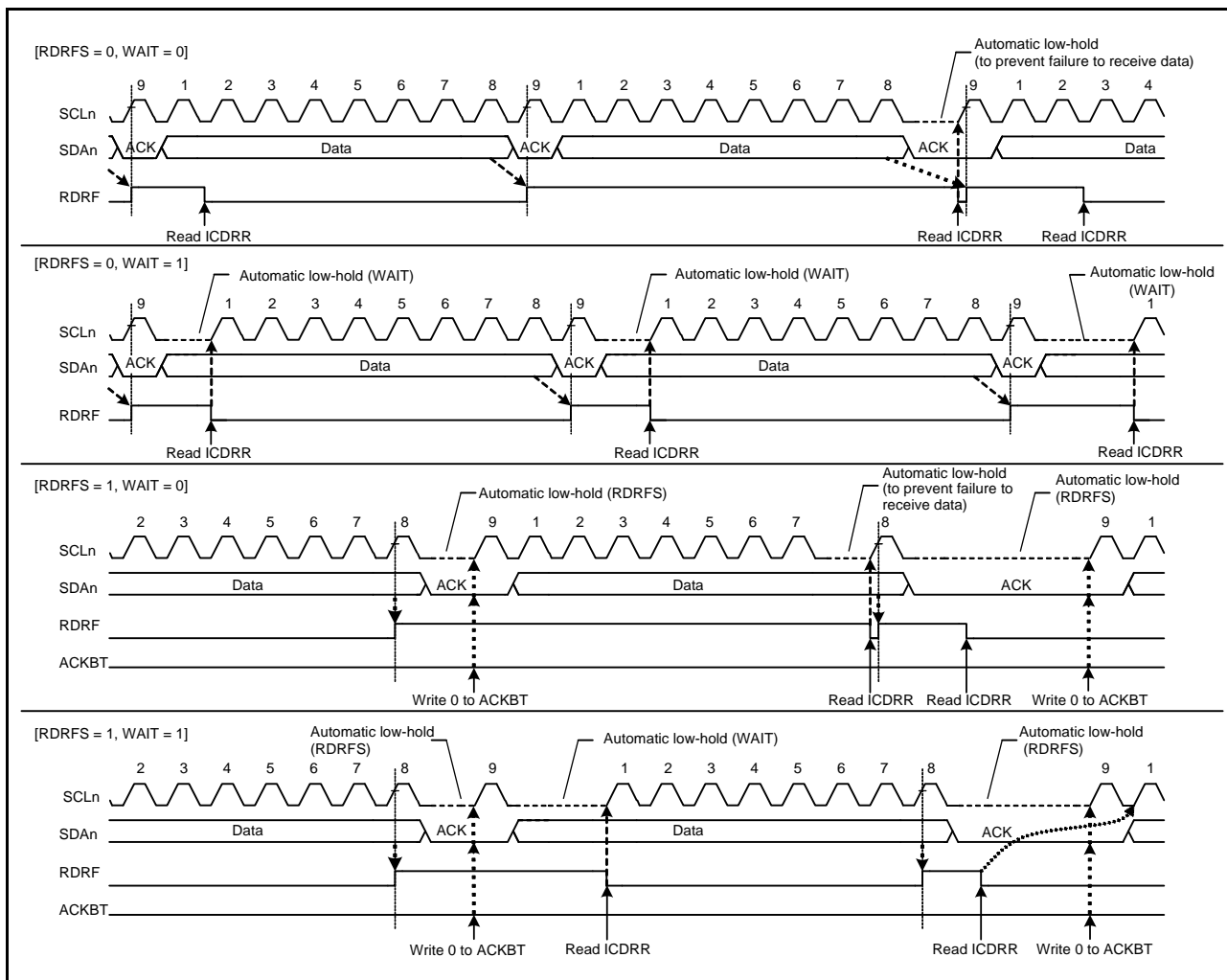


Figure 31.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

31.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I²C bus standard, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration during transmission of NACK, and to detect arbitration in slave transmit mode.

31.9.1 Master Arbitration Lost Detection (MALE Bit)

The RIIC drives the SDA_n line low to issue a start condition. However, if the SDA_n line has already been driven low by another master device issuing a start condition, the RIIC regards its own issuing of a start condition as an error and considers this a loss in arbitration, so priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), the RIIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration, thus preventing a failure of transfer due to issuing of a start condition while transfer is in progress.

When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA_n line do not match (the 1 output as the internal SDA output; i.e. the SDA_n pin is in the high-impedance state) and the low level is detected on the SDA line, the RIIC loses in arbitration.

After a loss in arbitration of mastership, the RIIC immediately enters slave receiver mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the MALE bit in ICFER is 1 (master arbitration lost detection enabled).

[Master arbitration lost conditions]

- Non-matching of the internal level for output on SDA and the level on the SDA_n line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 was cleared to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA_n line in master transmit mode (MST and TRS bits = 11b in ICCR2)

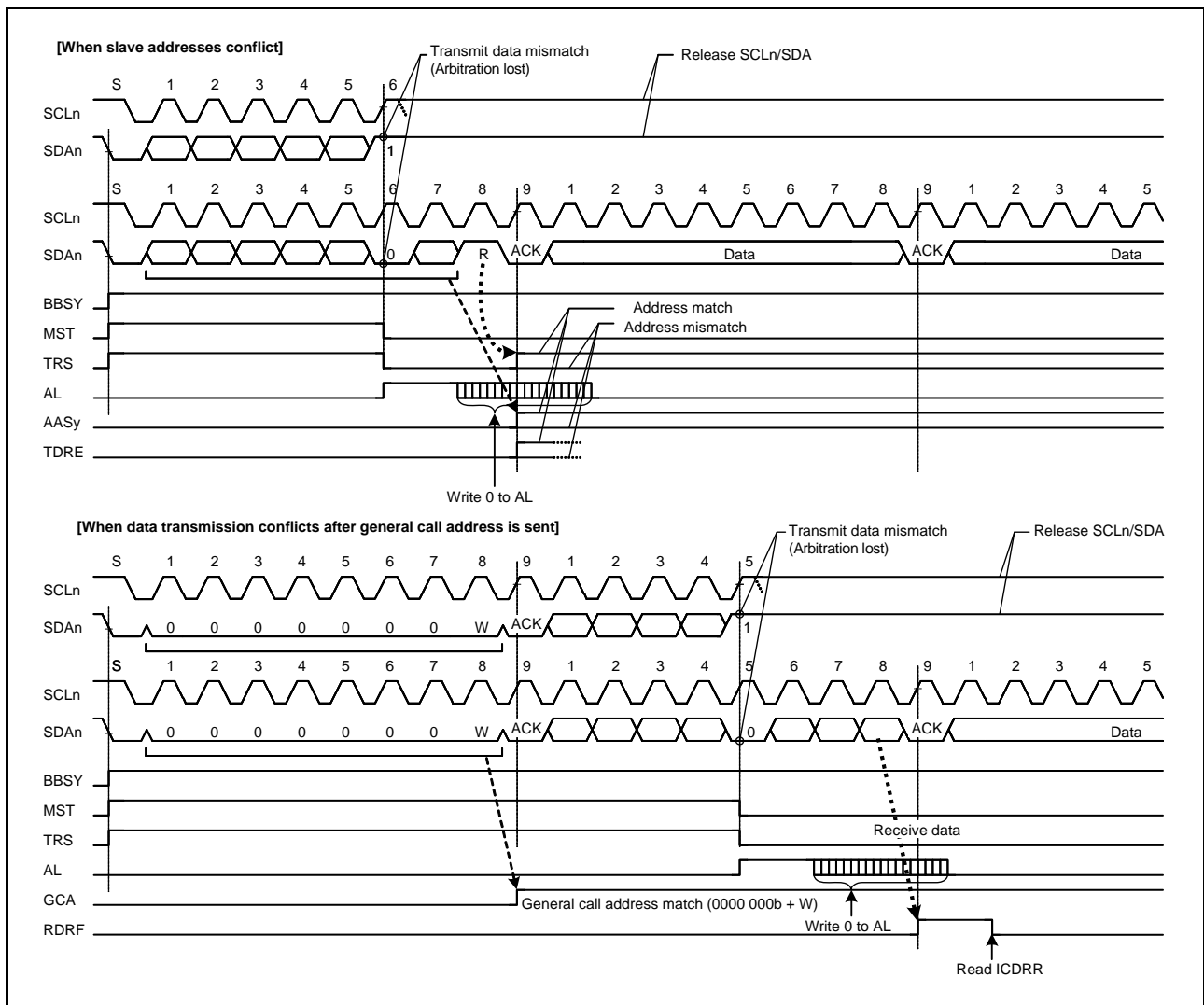


Figure 31.32 Examples of Master Arbitration Lost Detection (MALE = 1)

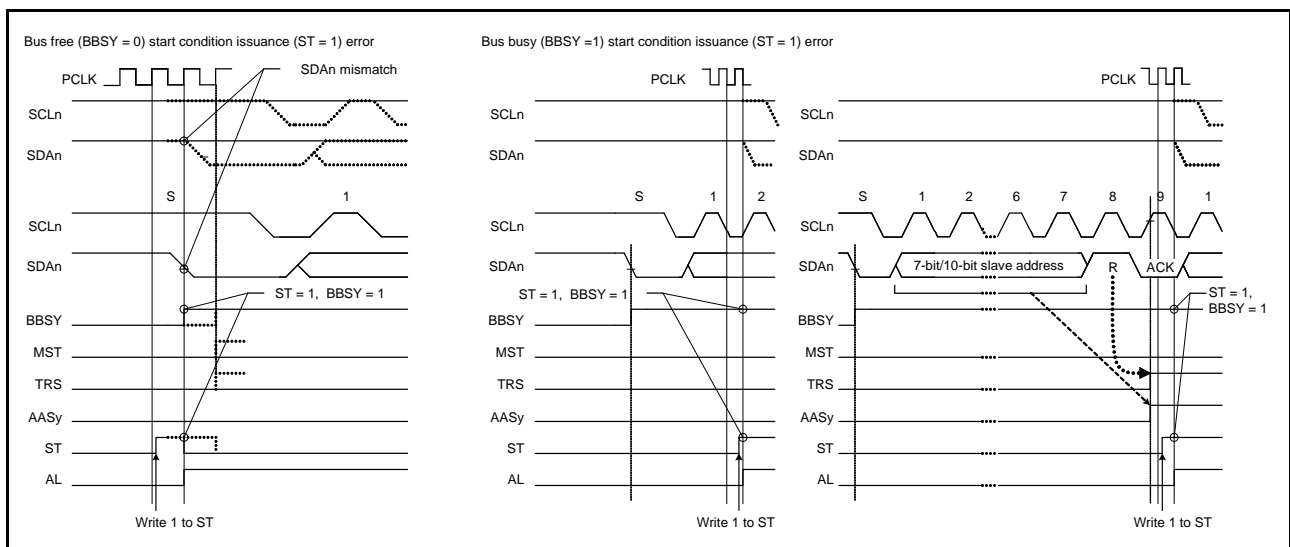


Figure 31.33 Arbitration Lost when a Start Condition is Issued (MALE = 1)

31.9.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA_n line (the 1 output as the internal SDA output; i.e. the SDA_n pin is in the high-impedance state) and the low level is detected on the SDA line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. Figure 31.34 shows an example of arbitration lost detection during transmission of NACK.

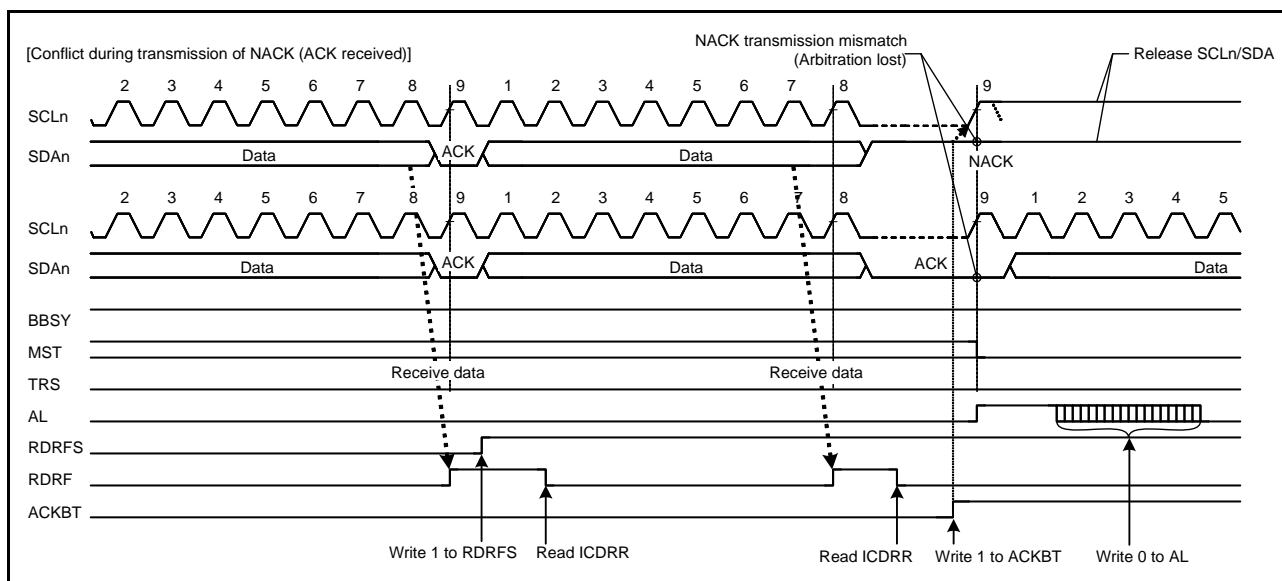


Figure 31.34 Example of Arbitration Lost Detection during Transmission of NACK (NALE = 1)

The following explains arbitration lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives two bytes of data from the slave device, and master B receives four bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received two final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition. Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When the RIIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, the RIIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing (such as FFh transmission processing) necessary if no response is received in the Get UDID (general) processing after the Assign address command.

The RIIC detects arbitration lost during transmission of NACK when the following condition is met with the NALE bit in ICFER set to 1 (arbitration lost detection during NACK transmission enabled).

[Condition for arbitration lost during NACK transmission]

- When the internal SDA output level does not match the SDA line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3)

31.9.3 Slave Arbitration Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA line do not match (the 1 output as the internal SDA output; i.e. the SDA pin is in the high-impedance state) and the low level is detected on the SDA line in slave transmitter 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 receiver mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminates subsequent redundant processing (processing for the transmission of FFh).

The RIIC detects slave arbitration lost when the following condition is met with the SALE bit in ICFER set to 1 (slave arbitration lost detection enabled).

[Condition for slave arbitration lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDA line in slave transmit mode (MST and TRS bits = 01b in ICCR2)

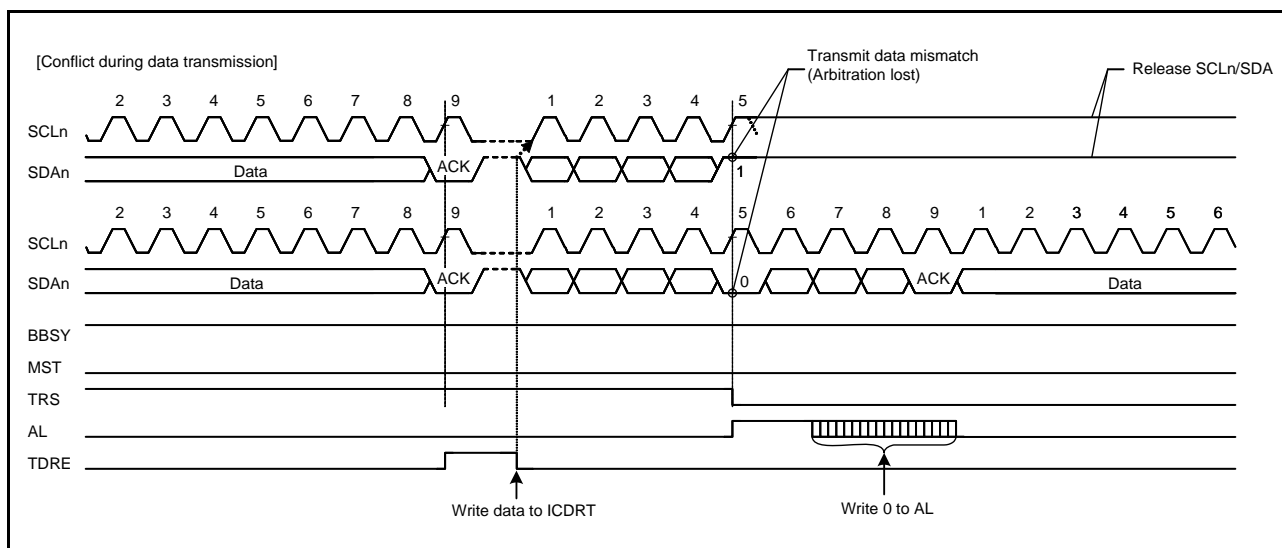


Figure 31.35 Example of Slave Arbitration Lost Detection (SALE = 1)

31.10 Start Condition/Restart Condition/Stop Condition Issuing Function

31.10.1 Issuing a Start Condition

The RIIC issues a start condition when the ST bit in ICCR2 is set to 1.

When the ST bit is set to 1, a start condition issuance request is made and the RIIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.

A start condition is issued in the following sequence.

[Start condition issuance]

- Drive the SDA_n line low (high level to low level).
- Ensure the time set in ICBRH and the start condition hold time.
- Drive the SCL_n line low (high level to low level).
- Detect low level of the SCL_n line and ensure the low-level period of SCL_n set in ICBRL.

31.10.2 Issuing a Restart Condition

The RIIC issues a restart condition when the RS bit in ICCR2 is set to 1.

When the RS bit is set to 1, a restart condition issuance request is made and the RIIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy) and the MST bit in ICCR2 is 1 (master mode).

A restart condition is issued in the following sequence.

[Restart condition issuance]

- Release the SDA_n line.
- Ensure the low-level period of SCL_n line set in ICBRL.
- Release the SCL_n line (low level to high level).
- Detect a high level of the SCL_n line and ensure the time set in ICBRL and the restart condition setup time.
- Drive the SDA_n line low (high level to low level).
- Ensure the time set in ICBRH and the restart condition hold time.
- Drive the SCL_n line low (high level to low level).
- Detect a low level of the SCL_n line and ensure the low-level period of SCL_n line set in ICBRL.

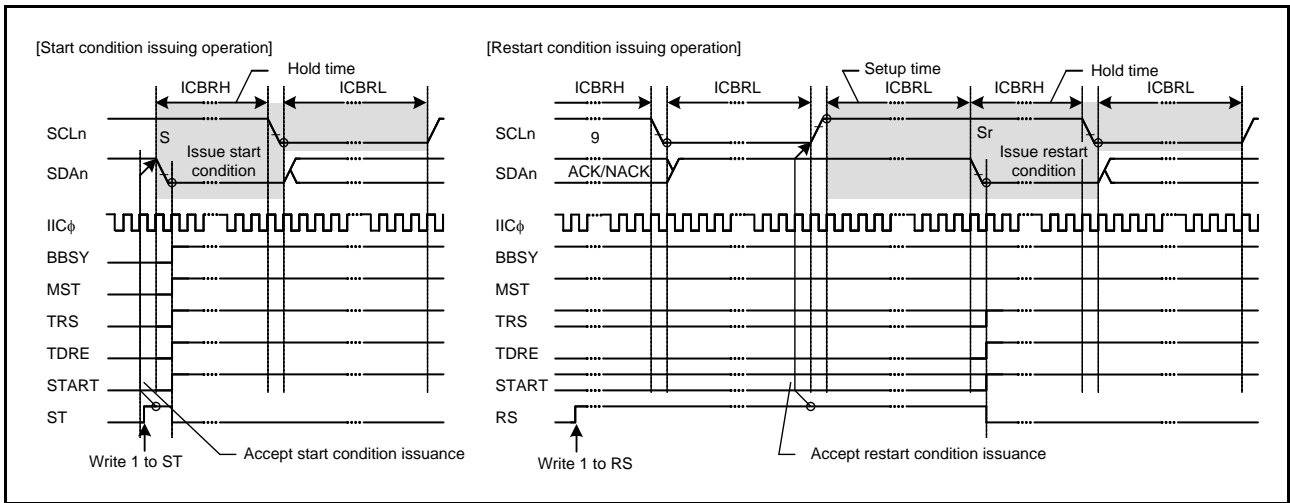


Figure 31.36 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

31.10.3 Issuing a Stop Condition

The RIIC issues a stop condition when the SP bit in ICCR2 is set to 1.

When the SP bit is set to 1, a stop condition issuance request is made and the RIIC issues a stop condition when the BBSY flag in ICCR2 is 1 (bus busy) and the MST bit in ICCR2 is 1 (master mode).

A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDAn line low (high level to low level).
- Ensure the low-level period of SCLn line set in ICBRL.
- Release the SCLn line (low level to high level).
- Detect a high level of the SCLn line and ensure the time set in ICBRH and the stop condition setup time.
- Release the SDAn 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).

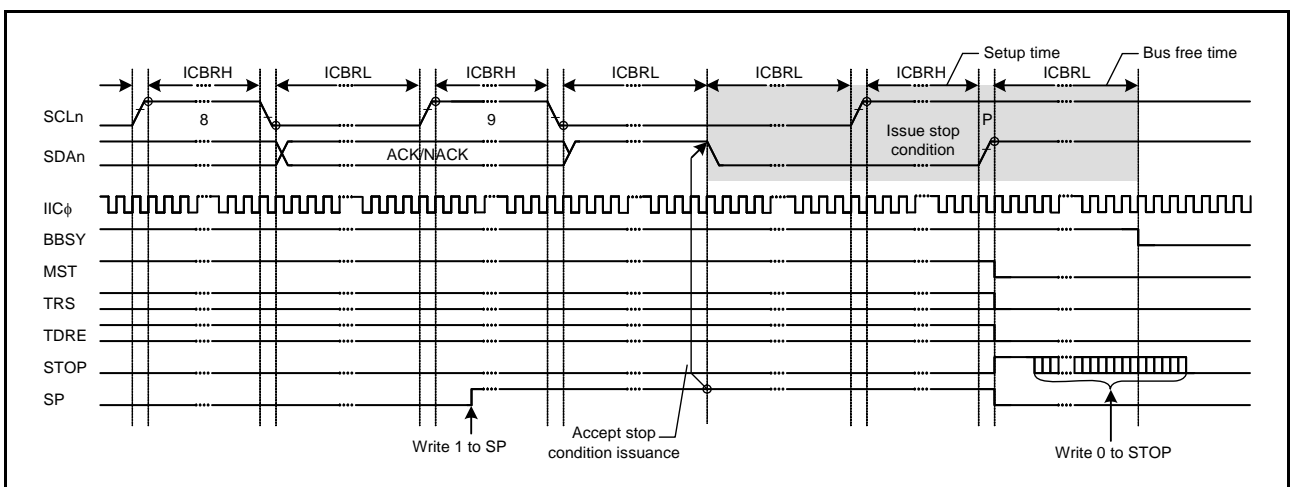


Figure 31.37 Stop Condition Issue Timing (SP Bit)

31.11 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I²C bus might hang with a fixed level on the SCLn line and/or SDA_n line.

As measures against the bus hanging, the RIIC has a timeout detection function to detect hanging by monitoring the SCLn line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, and the RIIC/internal reset function.

By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCLn or SDA_n lines.

31.11.1 Timeout Detection Function

The RIIC has the timeout detection function to detect an abnormality that the SCLn line is held for a certain period of time. In the bus busy state, the RIIC can detect an abnormal bus state by monitoring that the SCLn line is held low or high for a predetermined time.

The timeout detection function monitors the SCLn line state and counts the low-level period or high-level period using the internal counter. The timeout detection function resets the internal counter each time the SCLn line changes (rising or falling), but continues to count unless the SCLn line changes. If the internal counter overflows due to no SCLn line change, the RIIC can detect the timeout and report the bus abnormality.

This timeout detection function is enabled when the TMOE bit in ICFER is 1. It detects an abnormal bus state that the SCLn line is held low or high when the bus is busy (BBSY flag = 1 in ICCR2) in master mode or when the BBSY flag is 1 and the RIIC's own slave address matches (ICSR1 is not 00h) in slave mode.

The internal counter of the timeout detection function works using the internal reference clock (IIC ϕ) set by the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCLn line level (low/high or both levels) during which this counter is activated can be selected by the setting of the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are cleared to 0, the internal counter does not work.

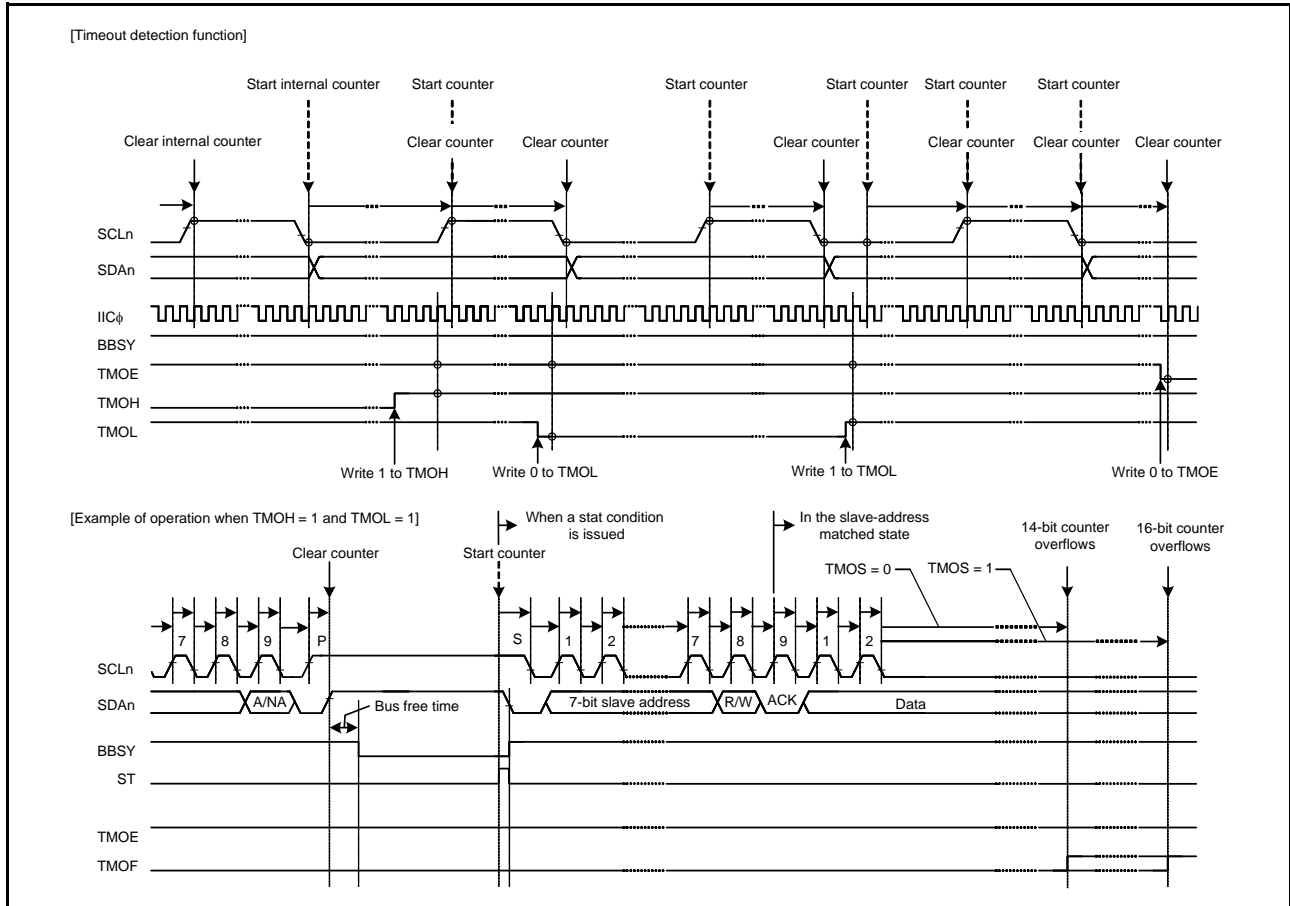


Figure 31.38 Timeout Detection Function (TMOE, TMOS, TMOH, and TMOL Bits)

31.11.2 Extra SCL Clock Cycle Output Function

In master mode, the RIIC module has a facility for the output of extra SCL (clock) cycles to release the SDAn line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the SDAn line of the slave device from the state of being fixed to the low level by including extra cycles of SCLn output from the RIIC with single cycles of the SCL (clock) signal as the unit in the case of a bus error where the RIIC cannot issue a stop condition because the slave device is holding the SDAn line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the frequency corresponding to the transfer rate settings (settings of the CKS[2:0] bits in ICMR1, and of the ICBRH and ICBRL registers) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically cleared to 0. Therefore, further extra clock cycles can be output consecutively by the software program writing 1 to the CLO bit after having read CLO = 0.

When the RIIC module is in master mode and the slave device is holding the SDAn line at the low level because synchronization with the slave device has been lost due to the effects of noise etc., the output of a stop condition is not possible. The facility for output of an extra cycle of the SCL (clock) signal can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming release of the SDAn line by the slave device, complete communications by reissuing the stop condition.

Use this facility with the MALE bit (master arbitration lost detection disabled) in ICFER cleared to 0. If the MALE bit is set to 1 (master arbitration lost detection enabled), arbitration is lost when the value of the SDAO bit in ICCR1 does not match the state of the SDAn line, so take care on this point.

[Output conditions for using the CLO bit in ICCR1]

- When the bus is free (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)
- When the communication device does not hold the SCLn line low

Figure 31.39 shows the operation timing of the extra SCL clock cycle output function.

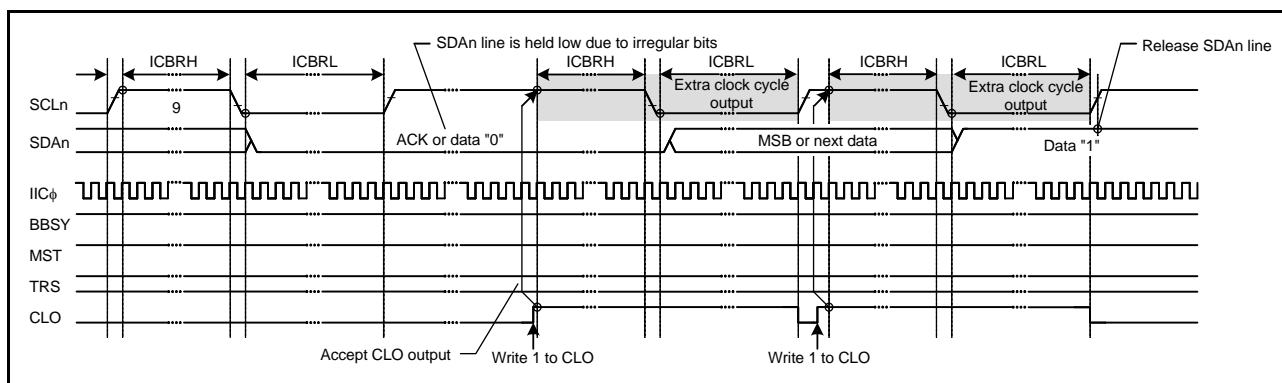


Figure 31.39 Extra SCL Clock Cycle Output Function (CLO Bit)

31.11.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the BBSY flag in ICCR2. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings. After issuing a reset, be sure to clear the IICRST bit in ICCR1 to 0.

Both types of reset are effective for release from bus-hung states since both restore the output state of the SCLn and SDAn pins to the high impedance state.

Issuing a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoided this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (ICE and IICRST bits = 01b in ICCR1).

For a detailed description of the RIIC and internal resets, see section 31.14, Reset States 30.14, Reset States.

31.12 SMBus Operation

The RIIC is available for data communication conforming to the SMBus standard (Version 2.0). To perform SMBus communication, set the SMBS bit in ICMR3 to 1 to select input level conforming to the SMBus for the SCLn pin/SDAn pin function. 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. When the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the ICBRL needs to be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (SARL0, SARL1, and SARL2), and set the corresponding FS bit (7-bit/10-bit address format select) in SARUn (n = 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.

31.12.1 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_{TIMEOUT} : 25 ms (min.) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the RIIC. When an internal reset is issued, the RIIC stops driving the bus for the SCLn pin and SDAn pin and make the SCLn/SDAn pin outputs high impedance, which releases the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: $T_{\text{LOW:MEXT}}$) 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 (ICTEI) or receive data full

interrupt (ICRXI) of the RIIC. The measured timeout period must be within the total clock low-level extended period [master device] $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus standard, and the total of all $T_{LOW:MEXT}$ from start condition to stop condition must be within $T_{LOW:SEXT}$: 25 ms (max.).

For the ACK receive timing (rising edge of the ninth SMBCLK clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). For this reason, perform bitwise transmit operation in master transmit mode, and hold the RDRFS bit in ICMR3 0 until the byte just before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SMBCLK clock cycle.

If the period measured with the MTU or TMR exceeds the total clock low-level extended period [master device] $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 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).

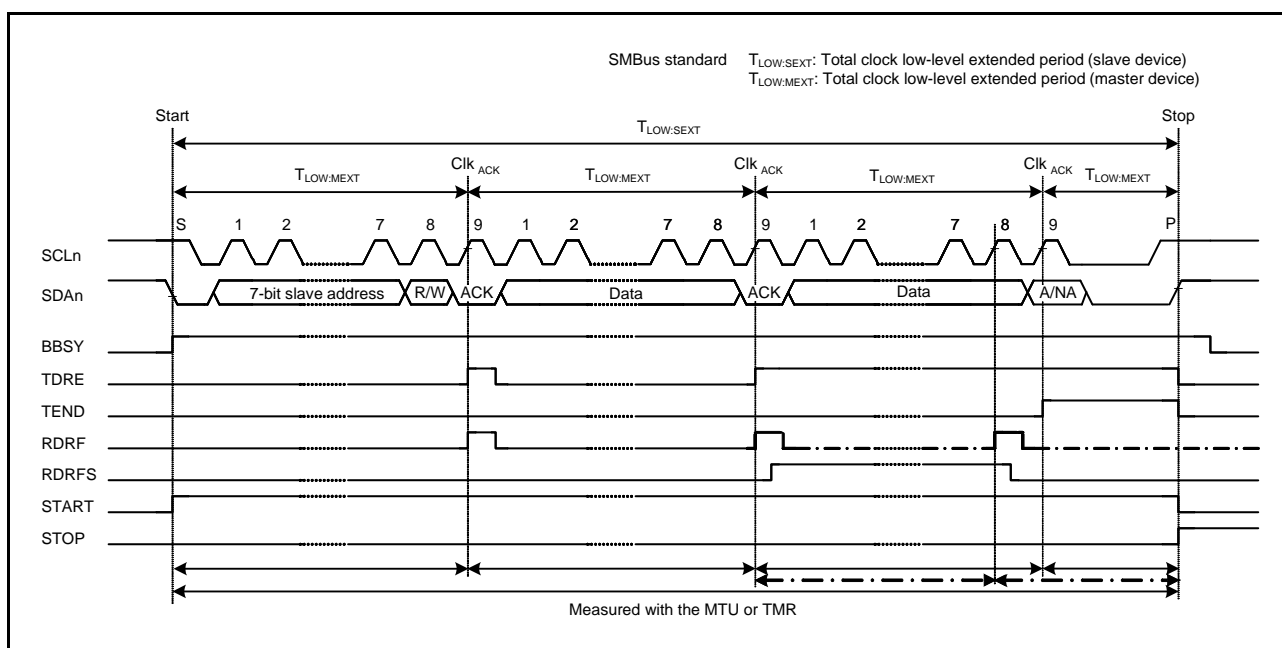


Figure 31.40 SMBus Timeout Measurement

31.12.2 Packet Error Code (PEC)

The RX62N/RX621 Group incorporates a CRC operation circuit. The CRC operation circuit 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 operation circuit, see section 30, CRC Calculator (CRC).

The PEC data in master transmit mode (master transmitter) can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC operation circuit.

The REC data in master receive mode (master receiver) can be checked by writing all receive data to CRCDIR in the CRC operation circuit and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the RDRFS bit in ICMR3 to 1 before the rising edge of the eighth SMBCLK clock cycle during reception of the final byte, and hold the SCLn line low at the falling edge of the eighth clock cycle.

31.12.3 SMBus Host Notification Protocol/Notify ARP Master

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of (or request the SMBus host for) its own slave address or to request its own slave address from the SMBus host.

For a product of the RX62N/RX621 Group to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so the RIIC has a function for detecting the host address. To detect the host address as a slave address, set the SMBS bit in ICMR3 and the HOAE bit in IC SER to 1. Operation after the host address has been detected is the same as normal slave operation.

31.13 Interrupt Request

The RIIC issues four types of interrupt request: transfer error or event generation (arbitration lost, NACK detection, timeout detection, start condition detection, and stop condition detection), receive data full, transmit data empty, and transmit end.

Table 31.7 shows details of the several interrupt requests. The receive data full and transmit data empty are both capable of launching data transfer by the DTC or DMACA.

Table 31.7 Interrupt Sources

Abbreviation	Interrupt Request	Interrupt Flag	DTC Launching	DMACA Launching	Priority	Interrupt Condition
ICEEI	Transfer Error/ Event Generation	AL, NACKF, TMOF, START, STOP	Not possible	Not possible	High	(AL=1) • (ALIE=1) (NACKF=1) • (NAKIE=1) (TMOF=1) • (TMOIE=1) (START=1) • (STIE=1) (STOP=1) • (SPIE=1)
ICRXI	Receive Data Full	—	Possible	Possible	↑	(RDRF=1) • (RIE=1)
ICTXI	Transmit Data Empty	—	Possible	Possible		(TDRE=1) • (TIE=1)
ICTEI	Transmit End	TEND	Not possible	Not possible		Low

Clear or mask the various interrupt sources during interrupt handling.

Notes on interrupt processing:

1. There is a latency (delay) between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt processing. Returning from interrupt processing without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.
2. Since ICTXI is an edge-detected interrupt, it does not require clearing. Furthermore, the TDRE flag in ICSR2 (a condition for ICTXI) is automatically cleared to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).
3. Since ICRXI is an edge-detected interrupt, it does not require clearing. Furthermore, the RDRF flag in ICSR2 (a condition for ICRXI) is automatically cleared to 0 when data are read out from ICDRR.
4. When using the ICTEI interrupt, clear the TEND flag in ICSR2 in the ICTEI interrupt processing. Note that the TEND flag in ICSR2 is automatically cleared to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).

31.14 Reset States

The RIIC has chip reset, RIIC reset, and internal reset functions. Table 31.8 shows the scope of each reset and reset conditions.

Table 31.8 Reset Conditions

		Chip Reset	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start Condition/ Restart Condition Detection	Stop Condition Detection
ICCR1	ICE, IICRST	At a reset	Retained	Retained	Operation (retained)	Operation (retained)
	SCLO, SDAO		At a reset	At a reset		
	Others			Retained		
ICCR2	BBSY	At a reset	At a reset	Operation	Operation	Operation
	ST			At a reset	At a reset	Operation (retained)
	Others					At a reset
ICMR1	BC[2:0]	At a reset	At a reset	At a reset	At a reset	Operation (retained)
	Others			Retained	Operation (retained)	
ICMR2		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICMR3		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICFER		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICSER		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICIER		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICSR1		At a reset	At a reset	At a reset	Operation (retained)	At a reset
ICSR2	TDRE, TEND	At a reset	At a reset	At a reset	Operation (retained)	At a reset
	START				Operation	
	STOP				Operation (retained)	Operation
	Others					Operation (retained)
SARL0 to SARL2 SARU0 to SARU2		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICBRH, ICBRL		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICDRT		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICDRR		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)
ICDRS		At a reset	At a reset	At a reset	Operation (retained)	Operation (retained)
Timeout detection function		At a reset	At a reset	Operation	Operation	Operation
Bus free time measurement		At a reset	At a reset	Operation	Operation	Operation

31.15 Usage Notes

31.15.1 Setting Module Stop Function

Module stop state can be entered or canceled using module stop control register B (MSTPCRB). The initial setting is for operation of the RIIC to be halted. RIIC register access is enabled by clearing module stop state.

For details of module stop control register B, see [section 9, Low Power Consumption](#).

31.15.2 Setting Input Buffer Control Register

Input to peripheral modules can be enabled or disabled using the input buffer control register (PORTm.ICR). The initial setting is for input to the RIIC to be disabled.

As the SCL and SDA lines on the I²C bus are bidirectional, the SCLn and SDAn pins of the RIIC are input/output pins. Make appropriate settings in the input buffer control bits in the PORTm.ICR corresponding to the SCLn and SDAn pins of the RIIC to enable input to the RIIC. If the required input is disabled, the RIIC cannot detect start conditions (including restart conditions) or stop conditions, or count the SCL clock cycles.

For details of the input buffer control register, see [section 17, I/O Ports](#).

32. CAN Module (CAN)

32.1 Overview

The RX62N/RX621 Group implements one channel of the CAN (Controller Area Network) module that complies with the ISO11898-1 Specifications. The CAN module transmits and receives both formats of messages, namely the standard identifier (11 bits) (identifier is hereafter referred to as ID) and extended ID (29 bits).

Table 32.1 lists the specifications of the CAN module, and Figure 32.1 shows a block diagram of the CAN module.

Connect the CAN bus transceiver externally.

Note: Only R5F562NxBxxx and R5F5621xBxxx have the CAN module. For details on the part numbers, see Table 1.3 and Table 1.4.

Table 32.1 Specifications of CAN Module (1 / 2)

Item	Description
Protocol	<ul style="list-style-type: none"> ISO11898-1 compliant (standard and extended frames)
Bit rate	<ul style="list-style-type: none"> Programmable bit rate up to 1 Mbps (fCAN ≥ 8 MHz) fCAN: CAN clock source
Message box	<ul style="list-style-type: none"> 32 mailboxes: Two selectable mailbox modes Normal mailbox mode: 32 mailboxes can be configured for either transmission or reception. FIFO mailbox mode: 24 mailboxes can be configured for either transmission or reception. Of the other mailboxes, four FIFO stages can be configured for transmission and four FIFO stages for reception.
Reception	<ul style="list-style-type: none"> Data frame and remote frame can be received. Selectable receiving ID format (only standard ID, only extended ID, or both IDs) Programmable one-shot reception function Selectable from overwrite mode (message overwritten) and overrun mode (message discarded) The reception complete interrupt can be individually enabled or disabled for each mailbox.
Acceptance filter	<ul style="list-style-type: none"> Eight acceptance masks (one mask for every four mailboxes) The mask can be individually enabled or disabled for each mailbox.
Transmission	<ul style="list-style-type: none"> Data frame and remote frame can be transmitted. Selectable transmitting ID format (only standard ID, only extended ID, or both IDs) Programmable one-shot transmission function Selectable from ID priority mode and mailbox number priority mode Transmission request can be aborted (the completion of abort can be confirmed with a flag) The transmission complete interrupt can be individually enabled or disabled for each mailbox.
Mode transition for bus-off recovery	<ul style="list-style-type: none"> Mode transition for the recovery from the bus-off state can be selected: <ul style="list-style-type: none"> ISO11898-1 Specifications compliant Automatic entry to CAN halt mode at bus-off entry Automatic entry to CAN halt mode at bus-off end Entry to CAN halt mode by a program Transition into error-active state by a program
Error status monitoring	<ul style="list-style-type: none"> CAN bus errors (stuff error, form error, ACK error, CRC error, bit error, and ACK delimiter error) can be monitored. Transition to error states can be detected (error-warning, error-passive, bus-off entry, and bus-off recovery). The error counters can be read.
Time stamp function	<ul style="list-style-type: none"> Time stamp function using a 16-bit counter The reference clock can be selected from 1-, 2-, 4- and 8-bit time periods.
Interrupt function	<ul style="list-style-type: none"> Five types of interrupt sources (reception complete, transmission complete, receive FIFO, transmit FIFO, and error interrupts)
CAN sleep mode	<ul style="list-style-type: none"> Current consumption can be reduced by stopping the CAN clock.

Table 32.1 Specifications of CAN Module (2 / 2)

Item	Description
Software support unit	<ul style="list-style-type: none"> Three software support units: <ul style="list-style-type: none"> Acceptance filter support Mailbox search support (receive mailbox search, transmit mailbox search, and message lost search) Channel search support
CAN clock source	<ul style="list-style-type: none"> Peripheral module clock (PCLK)
Test mode	<ul style="list-style-type: none"> Three test modes available for user evaluation <ul style="list-style-type: none"> Listen-only mode Self-test mode 0 (external loopback) Self-test mode 1 (internal loopback)

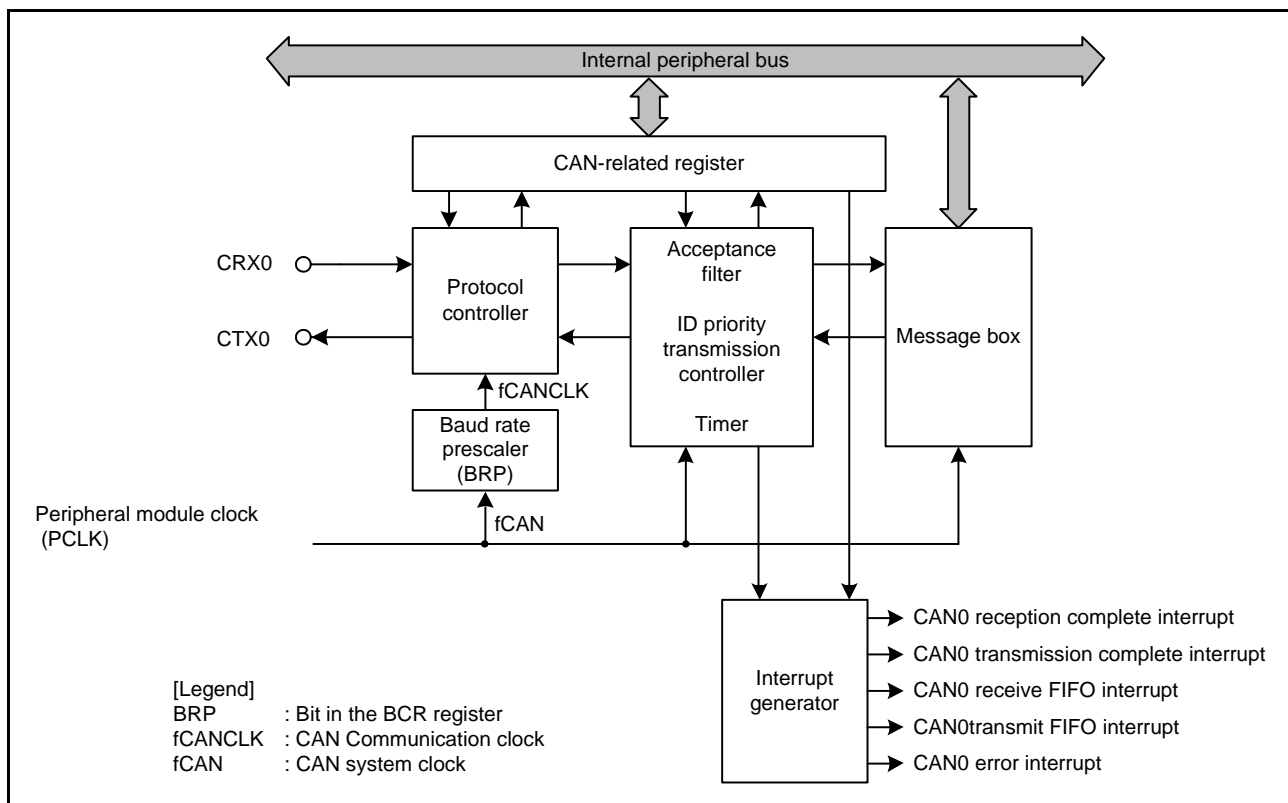


Figure 32.1 Block Diagram of CAN Module

- CRX0 and CTX0
CAN input and output pins
- Protocol controller
Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, and error handling, etc.
- Message box
Consists of 32 mailboxes which can be configured as either transmit or receive mailboxes. Each mailbox has an individual ID, data length code, a data field (8 bytes), and a time stamp.
- Acceptance filter
Performs filtering of received messages. MKR0 to MKR7 are used for the filtering process.
- Timer
Used for the time stamp function. The timer value when a message is stored into the mailbox is written as the time stamp value.
- Interrupt generator:
Generates the following five types of interrupts:
CAN0 reception complete interrupt
CAN0 transmission complete interrupt
CAN0 receive FIFO interrupt
CAN0 transmit FIFO interrupt
CAN0 error interrupt

Table 32.2 shows the CAN module pins.

The CAN functions should be selected for the pins multiplexed with other signals. For details, see section 17, I/O Ports.

Table 32.2 Pin Configuration

Pin Name	I/O	Function
CRX0	Input	Pin for receiving data
CTX0	Output	Pin for transmitting data

32.2 Register Descriptions

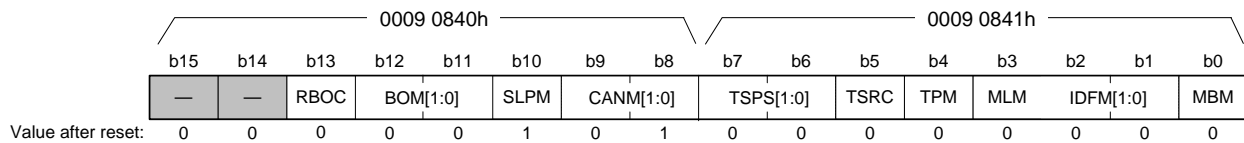
Table 32.3 lists the registers of the CAN module.

Table 32.3 Registers of CAN Module

Channel	Register Name	Symbol	Value after Reset	Address	Access Size
CAN0	Control register	CTLR	0500h	0009 0840h	8, 16
	Bit configuration register	BCR	0000 0000h	0009 0844h	8, 16, 32
	Mask register 0	MKR0	Undefined	0009 0400h	8, 16, 32
	Mask register 1	MKR1	Undefined	0009 0404h	8, 16, 32
	Mask register 2	MKR2	Undefined	0009 0408h	8, 16, 32
	Mask register 3	MKR3	Undefined	0009 040Ch	8, 16, 32
	Mask register 4	MKR4	Undefined	0009 0410h	8, 16, 32
	Mask register 5	MKR5	Undefined	0009 0414h	8, 16, 32
	Mask register 6	MKR6	Undefined	0009 0418h	8, 16, 32
	Mask register 7	MKR7	Undefined	0009 041Ch	8, 16, 32
	FIFO received ID compare register 0	FIDCR0	Undefined	0009 0420h	8, 16, 32
	FIFO received ID compare register 1	FIDCR1	Undefined	0009 0424h	8, 16, 32
	Mask invalid register	MKIVLR	Undefined	0009 0428h	8, 16, 32
	Mailbox registers 0 to 31	MB0 to MB31	Undefined	0009 0200h to 0009 03FFh	8, 16, 32
	Mailbox interrupt enable register	MIER	Undefined	0009 042Ch	8, 16, 32
	Message control registers 0 to 31	MCTL0 to MCTL31	00h	0009 0820h to 0009 083Fh	8
	Receive FIFO control register	RFCR	80h	0009 0848h	8
	Receive FIFO pointer control register	RFPCR	Undefined	0009 0849h	8
	Transmit FIFO control register	TFCR	80h	0009 084Ah	8
	Transmit FIFO pointer control register	TFPCR	Undefined	0009 084Bh	8
	Status register	STR	0500h	0009 0842h	8, 16
	Mailbox search mode register	MSMR	00h	0009 0853h	8
	Mailbox search status register	MSSR	80h	0009 0852h	8
	Channel search support register	CSSR	Undefined	0009 0851h	8
	Acceptance filter support register	AFSR	Undefined	0009 0856h	8, 16
	Error interrupt enable register	EIER	00h	0009 084Ch	8
	Error interrupt factor judge register	EIFR	00h	0009 084Dh	8
	Receive error count register	RECR	00h	0009 084Eh	8
	Transmit error count register	TECR	00h	0009 084Fh	8
	Error code store register	ECSR	00h	0009 0850h	8
	Time stamp register	TSR	0000h	0009 0854h	8, 16
	Test control register	TCR	00h	0009 0858h	8

32.2.1 Control Register (CTRLR)

Address: 0009 0840h



Bit	Symbol	Bit Name	Description	R/W
b0	MBM	CAN Mailbox Mode Select*1	0: Normal mailbox mode 1: FIFO mailbox mode	R/W
b2, b1	IDFM[1:0]	ID Format Mode Select*1	b2 b1 0 0: Standard ID mode All mailboxes (including FIFO mailboxes) handle only standard IDs. 0 1: Extended ID mode All mailboxes (including FIFO mailboxes) handle only extended IDs. 1 0: Mixed ID mode All mailboxes (including FIFO mailboxes) handle both standard IDs and extended IDs. Standard IDs or extended IDs are specified by using the IDE bit in the corresponding mailbox in normal mailbox mode. In FIFO mailbox mode, the IDE bit in the corresponding mailbox is used for mailboxes [0] to [23], the IDE bits in FIDCR0 and FIDCR1 are used for the receive FIFO, and the IDE bit in mailbox [24] is used for the transmit FIFO. 1 1: Do not use this combination	R/W
b3	MLM	Message Lost Mode Select*2	0: Overwrite mode 1: Overrun mode	R/W
b4	TPM	Transmission Priority Mode Select*2	0: ID priority transmit mode 1: Mailbox number priority transmit mode	R/W
b5	TSRC	Time Stamp Counter Reset Command*4	0: Nothing occurred 1: Reset*3	R/W
b7, b6	TSPS[1:0]	Time Stamp Prescaler Select*1	b7 b6 0 0: Every bit time 0 1: Every 2-bit time 1 0: Every 4-bit time 1 1: Every 8-bit time	R/W
b9, b8	CANM[1:0]	CAN Operating Mode Select*5	b9 b8 0 0: CAN operation mode 0 1: CAN reset mode 1 0: CAN halt mode 1 1: CAN reset mode (forcible transition)	R/W
b10	SLPM	CAN Sleep Mode*5*6	0: Other than CAN sleep mode 1: CAN sleep mode	R/W
b12, b11	BOM[1:0]	Bus-Off Recovery Mode*1	b12 b11 0 0: Normal mode (ISO11898-1 compliant) 0 1: Entry to CAN halt mode automatically at bus-off entry 1 0: Entry to CAN halt mode automatically at bus-off end 1 1: Entry to CAN halt mode (during bus-off recovery period) by a program request	R/W
b13	RBOC	Forcible Return From Bus-Off*2	0: Nothing occurred 1: Forcible return from bus-off*3	R/W
b15, b14	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Note 1. Write to the BOM[1:0], TSPS[1:0], TPM, MLM, IDFM[1:0], and MBM bits in CAN reset mode.

Note 2. Set the RBOC bit to 1 in the bus-off state.

Note 3. This bit is automatically set back to 0 after being set to 1. It should be read as 0.

Note 4. Set the TSRC bit to 1 in CAN operation mode.

Note 5. When the CANM[1:0] and SLPM bits are changed, check STR to ensure that the mode has been switched. Do not change the CANM[1:0] bits or SLPM bit until the mode has been switched.

Note 6. Write to the SLPM bit in CAN reset mode or CAN halt mode. When changing the SLPM bit, write 0 or 1 to only the SLPM bit.

MBM Bit (CAN Mailbox Mode Select)

When the MBM bit is 0 (normal mailbox mode), mailboxes [0] to [31] are configured as transmit or receive mailboxes.

When the MBM bit is 1 (FIFO mailbox mode), mailboxes [0] to [23] are configured as transmit or receive mailboxes.

Mailboxes [24] to [27] are configured as a transmit FIFO and mailboxes [28] to [31] as a receive FIFO.

Transmit data is written into mailbox [24] (mailbox [24] is a window mailbox for the transmit FIFO). Receive data is read from mailbox [28] (mailbox [28] is a window mailbox for the receive FIFO).

Table 32.4 lists the mailbox configuration.

IDFM[1:0] Bits (ID Format Mode Select)

The IDFM[1:0] bits specify the ID format.

MLM Bit (Message Lost Mode Select)

The MLM bit specifies the operation when a new message is captured in the unread mailbox. Overwrite mode or overrun mode can be selected. All mailboxes (including the receive FIFO) are set to either overwrite mode or overrun mode.

When the MLM bit is 0, all mailboxes are set to overwrite mode and the new message is overwriting the old message.

When the MLM bit is 1, all mailboxes are set to overrun mode and the new message is discarded.

TPM Bit (Transmission Priority Mode Select)

The TPM bit specifies the priority of modes when transmitting messages. ID priority transmit mode or mailbox number transmit mode can be selected. All mailboxes are set for either ID priority transmission or mailbox number priority transmission.

When the TPM bit is 0, ID priority transmit mode is selected and transmission priority complies with the CAN bus arbitration rule, as defined in the ISO11898-1 Specifications. In ID priority transmit mode, mailboxes [0] to [31] (in normal mailbox mode), and mailboxes [0] to [23] (in FIFO mailbox mode), and the transmit FIFO are compared for the IDs of mailboxes configured for transmission. If two or more mailbox IDs are the same, the mailbox with the smaller number has higher priority.

Only the next message to be transmitted from the transmit FIFO is included in the transmission arbitration. If a transmit FIFO message is being transmitted, the next pending message within the transmit FIFO is included in the transmission arbitration.

When the TPM bit is 1, mailbox number transmit mode is selected and the transmit mailbox with the smallest mailbox number has the highest priority. In FIFO mailbox mode, the transmit FIFO has lower priority than normal mailboxes (mailboxes [0] to [23]).

TSRC Bit (Time Stamp Counter Reset Command)

The TSRC bit is used to reset the time stamp counter. When the TSR bit is set to 1, TSR is set to 0000h. Then this bit is automatically set to 0.

TSPS[1:0] Bits (Time Stamp Prescaler Select)

The TSPS bits select the prescaler for the time stamp. The reference clock for the time stamp can be selected to be either 1-, 2-, 4- or 8-bit time periods.

CANM[1:0] Bits (CAN Operating Mode Select)

The CANM[1:0] bits select one of the following modes for the CAN module: CAN operation mode, CAN reset mode, or CAN halt mode. CAN sleep mode is set by the SLPM bit. For details, refer to section 32.3, Operating Mode.

When the CAN module enters CAN halt mode according to the setting of the BOM[1:0] bits, the CANM[1:0] bits are automatically set to 10b.

SLPM Bit (CAN Sleep Mode)

When the SLPM bit is set to 1, the CAN module enters CAN sleep mode. When the SLPM bit is set to 0, the CAN module exits CAN sleep mode. For details, refer to section 32.3, Operating Mode.

BOM[1:0] Bits (Bus-Off Recovery Mode)

The BOM[1:0] bits are used to select bus-off recovery mode for the CAN module.

When the BOM[1:0] bits are 00b, the recovery from bus-off is compliant with the ISO11898-1 Specifications, i.e. the CAN module re-enters CAN communication (error-active state) after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off.

When the BOM[1:0] bits are 01b, the CAN module enters CAN halt mode as soon as it reaches the bus-off state and the CANM[1:0] bits in CTLR are set to 10b (CAN halt mode). No bus-off recovery interrupt request is generated when recovering from bus-off, and TECR and RECR are set to 00h.

When the BOM[1:0] bits are 10b, the CANM[1:0] bits are set to 10b as soon as the CAN module reaches the bus-off state. The CAN module enters CAN halt mode after the recovery from the bus-off state, i.e. after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off, and TECR and RECR are set to 00h.

When the BOM[1:0] bits are 11b, the CAN module enters CAN halt mode by setting the CANM[1:0] bits to 10b while the CAN module is still in the bus-off state. No bus-off recovery interrupt request is generated when recovering from bus-off and TECR and RECR are set to 00h. However, if the CAN module recovers from bus-off after detecting 11 consecutive recessive bits 128 times before the CANM[1:0] bits are set to 10b, a bus-off recovery interrupt request is generated.

If the CPU requests an entry to CAN reset mode at the same time as the CAN module attempts to enter CAN halt mode (at bus-off entry when the BOM[1:0] bits are 01b, or at bus-off end when the BOM[1:0] bits are 10b), then the CPU request to enter CAN reset mode has higher priority.

RBOC Bit (Forcible Return From Bus-Off)

When the RBOC bit is set to 1 (force return from bus-off) in the bus-off state, the CAN module forcibly returns from the bus-off state. Then this bit is automatically set to 0. The error state changes from bus-off to error-active. When the RBOC bit is set to 1, RECR and TECR are set to 00h and the BOST bit in STR is set to 0 (the CAN module is not in bus-off state). The other registers remain unchanged even when the RBOC bit is set to 1. No bus-off recovery interrupt request is generated by this recovery from the bus-off state. Use the RBOC bit only when the BOM[1:0] bits are 00b (normal mode).

Table 32.4 Mailbox Configuration

Mailbox	MBM Bit = 0 (Normal Mailbox Mode)	MBM Bit = 1 (FIFO Mailbox Mode)
Mailboxes [0] to [23]	Normal mailbox	Normal mailbox
Mailboxes [24] to [27]		Transmit FIFO
Mailboxes [28] to [31]		Receive FIFO

Note: Points 1 to 5 below should be considered when the CTLR.MBM bit is set to 1.

Note 1. Transmit FIFO is controlled by TFCR.MCTLj (j = 0 to 31) of mailboxes [24] to [27] is disabled. MCTL24 to MCTL27 cannot be used by the transmit FIFO.

Note 2. Receive FIFO is controlled by RFCR.MCTLj (j = 0 to 31) of mailboxes [28] to [31] is disabled. MCTL28 to MCTL31 cannot be used by the receive FIFO.

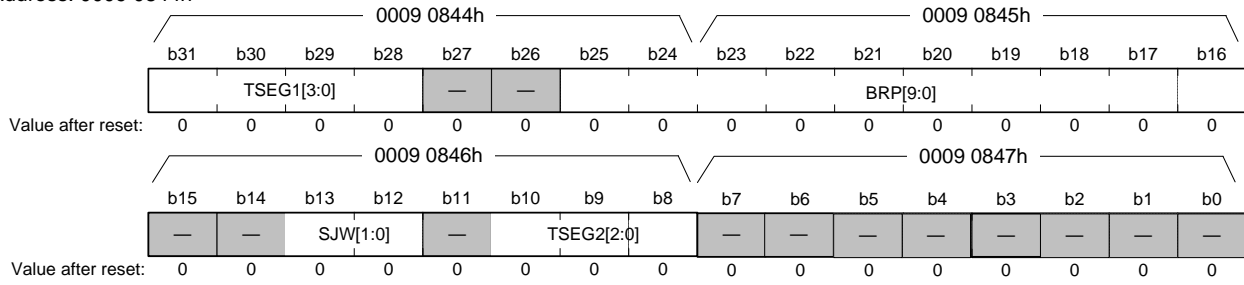
Note 3. Refer to MIER about the FIFO interrupts.

Note 4. The corresponding bits in MKIVLR for mailboxes [24] to [31] are disabled. Set 0 to these bits.

Note 5. Transmit/receive FIFOs can be used for both data frames and remote frames.

32.2.2 Bit Configuration Register (BCR)

Address: 0009 0844h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b10 to b8	TSEG2[2:0]	Time Segment 2 Control	b10 b9 b8 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq	R/W
b11	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b13, b12	SJW[1:0]	Resynchronization Jump Width Control	b13 b12 0 0: 1Tq 0 1: 2Tq 1 0: 3Tq 1 1: 4Tq	R/W
b15, b14	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b25 to b16	BRP[9:0]	Prescaler Division Ratio Select	These bits set the frequency of the CAN communication clock (fCANCLK).	R/W
b26	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b27	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b31 to 28	TSEG1[3:0]	Time Segment 1 Control	b31 b28 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq	R/W

[Legend] Tq: Time Quantum

BCR specifies the segment length with a T_q value.

For bit timing setting, refer to section 32.4, CAN Communication Speed Setting.

Set BCR before entering CAN halt mode or CAN operation mode from CAN reset mode. After the setting is made once, this register can be written to in CAN reset mode or CAN halt mode.

BCR consists of 24 bits. A 32-bit read/write access should be performed carefully not to rewrite bits b7 to b0.

TSEG2[2:0] Bits (Time Segment 2 Control)

The TSEG2[2:0] bits are used to specify the length of the phase buffer segment 2 (PHASE_SEG2) with a T_q value. A value from 2 to 8 T_q can be set. Set a value smaller than that of the TSEG1[3:0] bits.

SJW[1:0] Bits (Resynchronization Jump Width Control)

The SJW[1:0] bits are used to specify the resynchronization jump width with a T_q value. A value from 1 to 4 T_q can be set. Set a value smaller than or equal to that of the TSEG2[2:0] bits.

BRP[9:0] Bits (Prescaler Division Ratio Select)

The BRP[9:0] bits are used to set the frequency of the CAN communication clock (fCANCLK). The fCANCLK cycle is 1 T_q . If the setting is P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

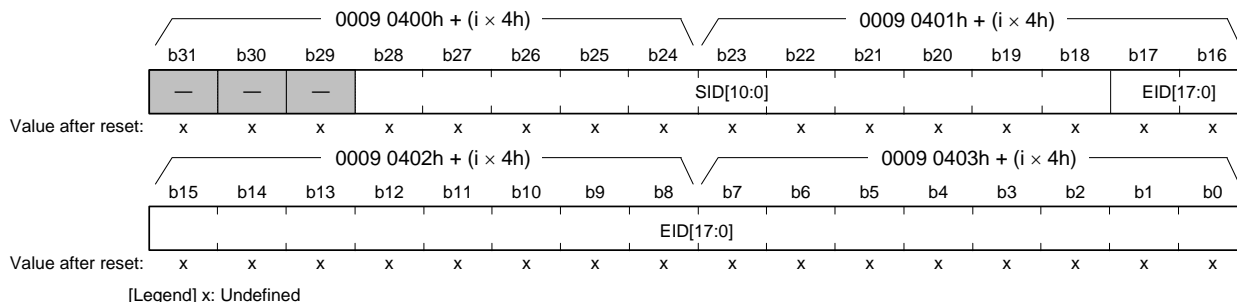
TSEG1[3:0] Bits (Time Segment 1 Control)

The TSEG1[3:0] bits are used to specify the total length of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) with a time quantum (T_q) value.

A value from 4 to 16 T_q can be set.

32.2.3 Mask Register i (MKRi) (i = 0 to 7)

Addresses: 0009 0400h to 0009 041Ch



Bit	Symbol	Bit Name	Description	R/W
b17 to b0	EID[17:0]	Extended ID	0: Corresponding EID[17:0] bit is not compared 1: Corresponding EID[17:0] bit is compared	R/W
b28 to b18	SID[10:0]	Standard ID	0: Corresponding SID[10:0] bit is not compared 1: Corresponding SID[10:0] bit is compared	R/W
b31 to b29	—	Reserved	The read value is undefined. The write value should always be 0.	R/W

For the mask function in FIFO mailbox mode, refer to section 32.6, Acceptance Filtering and Masking Functions. Write to MKR0 to MKR7 in CAN reset mode or CAN halt mode.

EID[17:0] Bits (Extended ID)

The EID[17:0] bits are the filter mask bits for the CAN extended ID bits.

These bits are used to receive extended ID messages.

When the EID[17:0] bit is set to 0, the received ID is compared with the mailbox ID for the corresponding EID[17:0] bit.

When the EID[17:0] bit is set to 1, the received ID is not compared with the mailbox ID for the corresponding EID[17:0] bit.

SID[10:0] Bits (Standard ID)

The SID[10:0] bits are the filter mask bits corresponding to the CAN standard ID bits.

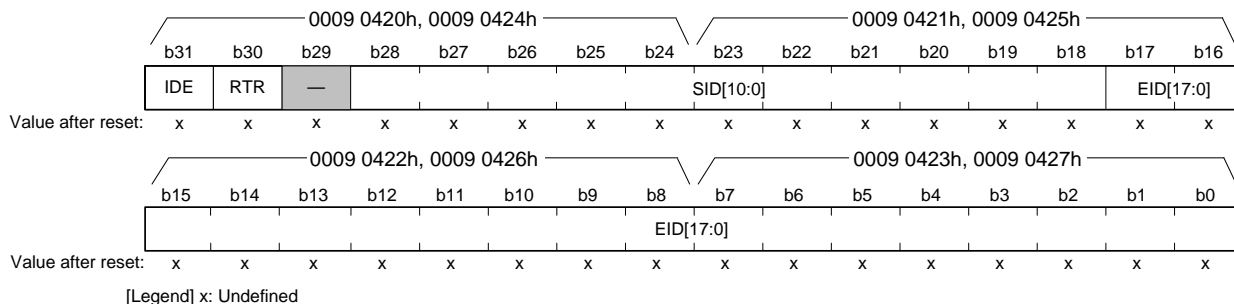
These bits are used to receive both standard ID and extended ID messages.

When the SID[10:0] bit is set to 0, the received ID is compared with the mailbox ID for the corresponding SID[10:0] bit.

When the SID[10:0] bit is set to 1, the received ID is not compared with the mailbox ID for the corresponding SID[10:0] bit.

32.2.4 FIFO Received ID Compare Registers 0 and 1 (FIDCR0, FIDCR1)

Addresses: FIDCR1 0009 0420h, FIDCR1 0009 0424h



Bit	Symbol	Bit Name	Description	R/W
b17 to b0	EID[17:0]	Extended ID	0: Corresponding EID[17:0] bit is 0 1: Corresponding EID[17:0] bit is 1	R/W
b28 to b18	SID[10:0]	Standard ID	0: Corresponding SID[10:0] bit is 0 1: Corresponding SID[10:0] bit is 1	R/W
b29	—	Reserved	The read value is undefined. The write value should always be 0.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R/W
b31	IDE	ID Extension*	0: Standard ID 1: Extended ID	R/W

Note : * When the IDFM[1:0] bits are not 10, the IDE bit should be written with 0 and read as 0.

FIDCR0 and FIDCR1 are enabled when the MBM bit in CTLR is set to 1 (FIFO mailbox mode). Bits EID[17:0], SID[10:0], RTR, and IDE in MB28 to MB31 are disabled.

For the usage of FIDCR0 and FIDCR1, refer to section 32.6, Acceptance Filtering and Masking Functions.

Write to FIDCR0 and FIDCR1 in CAN reset mode or CAN halt mode.

EID[17:0] Bits (Extended ID)

The EID[17:0] bits set the extended ID of data frames and remote frames.

These bits are used to receive extended ID messages.

SID[10:0] Bits (Standard ID)

The SID[10:0] bits set the standard ID of data frames and remote frames.

These bits are used to receive both standard ID and extended ID messages.

RTR Bit (Remote Transmission Request)

The RTR bit sets the specified frame format of data frames or remote frames.

- When both RTR bits in FIDCR0 and FIDCR1 are set to 0, only data frames can be received.
- When both RTR bits in FIDCR0 and FIDCR1 are set to 1, only remote frames can be received.
- When the RTR bits in FIDCR0 and FIDCR1 are set to 0 or 1 individually, both data frames and remote frames can be received.

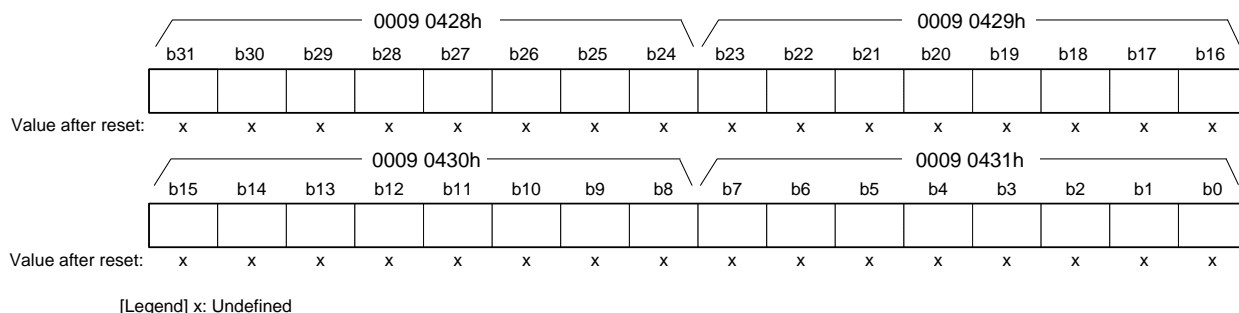
IDE Bit (ID Extension)

The IDE bit sets the ID format to standard ID or extended ID. The IDE bit is enabled when the IDFM[1:0] bits in CTLR is 10 (mixed ID mode).

- When both IDE bits in FIDCR0 and FIDCR1 are set to 0, only standard ID frames can be received.
- When both IDE bits in FIDCR0 and FIDCR1 are set to 1, only extended ID frames can be received.
- When the IDE bits in FIDCR0 and FIDCR1 are set to 0 or 1 individually, both standard ID and extended ID frames can be received.

32.2.5 Mask Invalid Register (MKIVLR)

Address: 0009 0428h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	0: Mask valid 1: Mask invalid	R/W

Each bit in MKIVLR corresponds to a mailbox.

The correspondence between the bits and mailboxes is shown below.

Bit 0 in MKIVLR corresponds to mailbox 0 and bit 31 corresponds to mailbox 31.*

When a bit is set to 1, the relevant acceptance mask register becomes invalid for the corresponding mailbox. When a mask invalid bit is set to 1, a message is received by the corresponding mailbox only if the receive message ID matches the mailbox ID exactly.

Write to MKIVLR in CAN reset mode or CAN halt mode.

Note: * Set bits 31 to 24 to 0 in FIFO mailbox mode.

32.2.6 Mailbox Register j (MBj) (j = 0 to 31)

Table 32.5 lists the CAN0 mailbox memory mapping, and Table 32.6 lists the CAN data frame configuration.

The value after reset of the CAN0 mailbox is undefined.

Write to MBj only when the related MCTLj (j = 0 to 31) is 00h and the corresponding mailbox is not processing an abort request.

See Table 32.3 for detailed register addresses.

Table 32.5 Mailbox Memory Mapping

Address	Register Symbol	Message Content
CAN0	CAN0	Memory Mapping
0009 0200h + 16 x j + 0	MB.ID	IDE, RTR, SID10 to SID6
0009 0200h + 16 x j + 1		SID5 to SID0, EID17, EID16
0009 0200h + 16 x j + 2		EID15 to EID8
0009 0200h + 16 x j + 3		EID7 to EID0
0009 0200h + 16 x j + 4	MB.DLC	—
0009 0200h + 16 x j + 5		Data length code (DLC[3:0])
0009 0200h + 16 x j + 6	MB.DATA0 to MB.DATA7	Data byte 0
0009 0200h + 16 x j + 7		Data byte 1
:		:
0009 0200h + 16 x j + 13		Data byte 7
0009 0200h + 16 x j + 14		MB.TS
0009 0200h + 16 x j + 15	Time stamp lower byte	

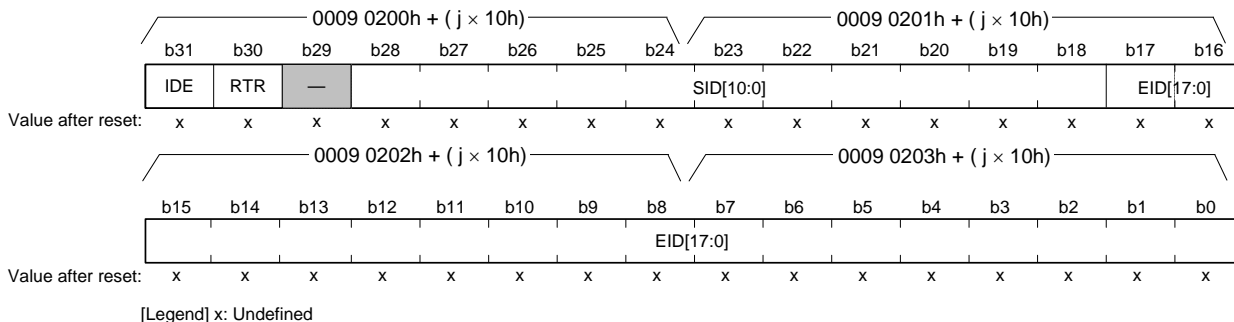
Table 32.6 CAN Data Frame Configuration

SID10 to SID6	SID5 to SID0	EID17 to EID16	EID15 to EID8	EID7 to EID0	DLC3 to DLC0	DATA0	DATA1	...	DATA7
------------------	--------------	----------------	---------------	--------------	--------------	-------	-------	-----	-------

The previous value of each mailbox is retained unless a new message is received.

(a) MB.ID

Addresses: 0009 0200h to 0009 03FFh



Bit	Symbol	Bit Name	Description	R/W
b17 to b0	EID[17:0]	Extended ID*1	0: Corresponding EID[17:0] bit is 0 1: Corresponding EID[17:0] bit is 1	R/W
b28 to b18	SID[10:0]	Standard ID	0: Corresponding SID[10:0] bit is 0 1: Corresponding SID[10:0] bit is 1	R/W
b29	—	Reserved	The read value is undefined. The write value should always be 0.	R/W
b30	RTR	Remote Frame Request	0: Data frame 1: Remote frame	R/W
b31	IDE	ID Extension*2	0: Standard ID 1: Extended ID	R/W

Note 1. If the mailbox has received a standard ID message, the EID bits in the mailbox are undefined.

Note 2. The IDE bit is enabled when the IDFM[1:0] bits in CTLR are 10b (mixed ID mode). When the IDFM[1:0] bits are not 10b, it should be written with 0 and read as 0.

EID[17:0] Bits (Extended ID)

The EID[17:0] bits set the extended ID of data frames and remote frames.

These bits are used to transmit or receive extended ID messages.

SID[10:0] Bits (Standard ID)

The SID[10:0] bits set the standard ID of data frames and remote frames.

These bits are used to transmit or receive both standard ID and extended ID messages.

RTR Bit (Remote Frame Request)

The RTR bit sets the frame format of data frames or remote frames.

- Receive mailbox receives only frames with the format specified by the RTR bit.
- Transmit mailbox transmits according to the frame format specified by the RTR bit.
- Receive FIFO mailbox receives the data frame, remote frame, or both frames specified by the RTR bit in FIDCR0 and FIDCR1.
- Transmit FIFO mailbox transmits the data frame or remote frame specified by the RTR bit in the relevant transmit message.

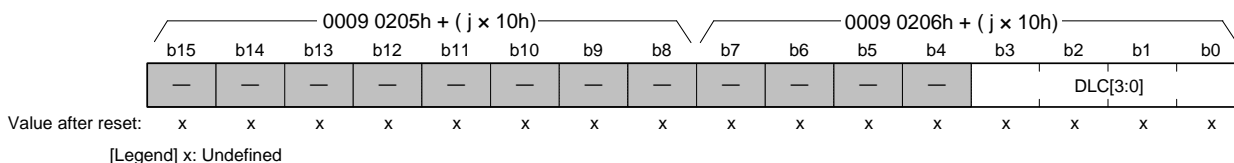
IDE Bit (ID Extension)

The IDE bit sets the ID format of standard IDs or extended IDs. The IDE bit is enabled when the IDFM[1:0] bits in CTLR is 10b (mixed ID mode).

- Receive mailbox receives only the ID format specified by the IDE bit.
- Transmit mailbox transmits with the ID format specified by the IDE bit.
- Receive FIFO mailbox receives messages with the standard ID, extended ID, or both IDs specified by the IDE bit in FIDCR0 and FIDCR1.
- Transmit FIFO mailbox transmits messages with the standard ID or extended ID specified by the IDE bit in the relevant transmit message.

(b) MB.DLC

Addresses: 0009 0205h to 0009 03F5h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DLC[3:0]	Data Length Code*	b3 b0 0 0 0 0: Data length = 0 byte 0 0 0 1: Data length = 1 byte 0 0 1 0: Data length = 2 bytes 0 0 1 1: Data length = 3 bytes 0 1 0 0: Data length = 4 bytes 0 1 0 1: Data length = 5 bytes 0 1 1 0: Data length = 6 bytes 0 1 1 1: Data length = 7 bytes 1 x x x: Data length = 8 bytes Note: x represents any value.	R/W
b15 to b4	—	Reserved	The read value is undefined. The write value should always be 0.	R/W

Note : * If the mailbox has received a message whose data length set by the DLC[3:0] bits is less than 8 bytes, the values of DATA larger than the data length set by the DLC[3:0] bits in the mailbox are undefined.

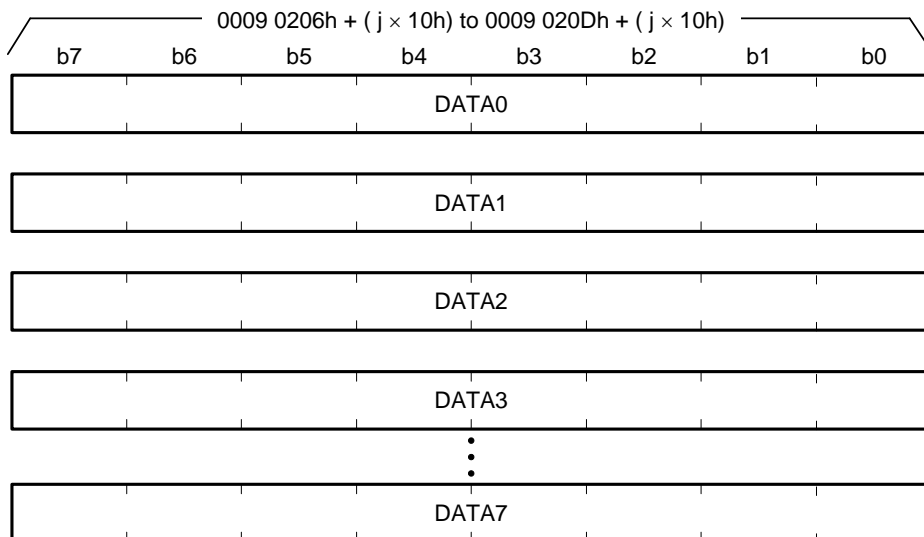
DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits are used to set the number of data bytes to be transmitted in a data frame. When data is requested using a remote frame, the number of data bytes to be requested is set.

When a data frame is received, the number of received data bytes is stored. When a remote frame is received, the number of requested data bytes is stored.

(c) MB.DATA0 to MB.DATA7

Addresses: 0009 0206h to 0009 03FDh



Value after reset: X X X X X X X X

[Legend] x: Undefined

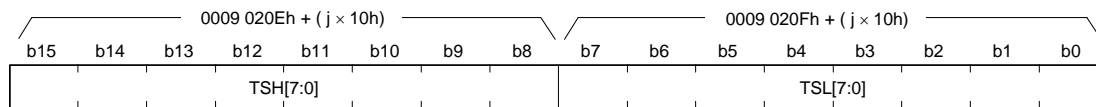
Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DATA0 to DATA7	Data Bytes 0 to 7*1*2	DATA0 to DATA7 store the transmitted or received CAN message data. Transmission or reception starts from DATA0. The bit order on the CAN bus is MSB first, and transmission or reception starts from bit 7.	R/W

Note 1. If the mailbox has received a message with n bytes less than 8 bytes, the values of DATA0 to DATA7 in the mailbox are undefined.

Note 2. If the mailbox has received a remote frame, the previous values of DATA0 to DATA7 in the mailbox are retained.

(d) MB.TS

Addresses: 0009 020Eh to 0009 03FFh



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

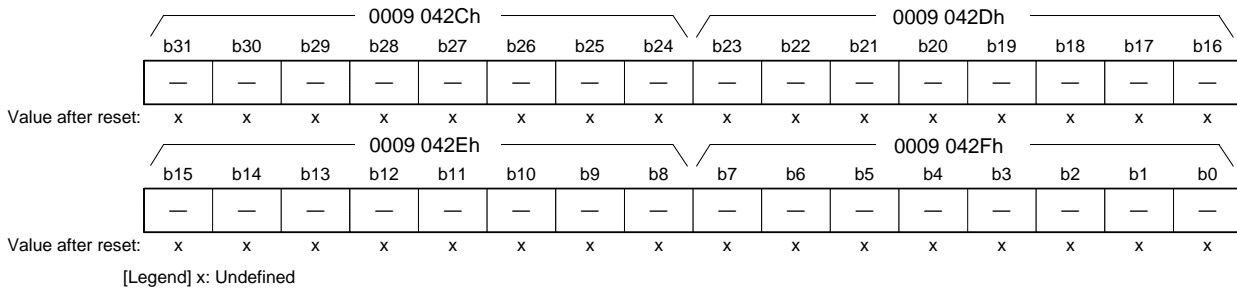
[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TSL[7:0]	Time Stamp Lower Byte	Bits TSH[7:0] and TSL[7:0] store the counter value of the time stamp when received messages are stored in the mailbox.	R/W
b15 to b8	TSH[7:0]	Time Stamp Higher Byte		R/W

32.2.7 Mailbox Interrupt Enable Register (MIER)

- Normal mailbox mode

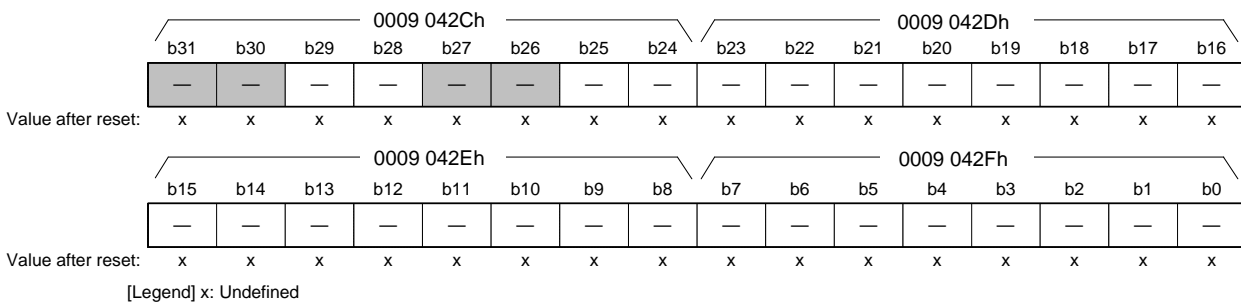
Address: 0009 042Ch



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled Bit 31 corresponds to mailbox 31, and bit 0 corresponds to mailbox 0.	R/W

- FIFO mailbox mode

Address: 0009 042Ch



Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled Bit 23 corresponds to mailbox 23, and bit 0 corresponds to mailbox 0.	R/W
b24	—	Transmit FIFO Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W
b25	—	Transmit FIFO Interrupt Generation Timing Control	0: Every time transmission is completed 1: When the transmit FIFO becomes empty due to completion of transmission	R/W
b27, b26	—	Reserved	The read value is undefined. The write value should always be 0.	R/W
b28	—	Receive FIFO Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W
b29	—	Receive FIFO Interrupt Generation Timing Control*	0: Every time reception is completed 1: When the receive FIFO becomes buffer warning by completion of reception	R/W
b31, b30	—	Reserved	The read value is undefined. The write value should always be 0.	R/W

Note : * No interrupt request is generated when the receive FIFO becomes buffer warning from full. "Buffer warning" indicates a state in which the third message is stored in the receive FIFO.

MIER can individually enable interrupts for each mailbox.

In normal mailbox mode (all bits) and in FIFO mailbox mode (bits 23 to 0 in MIER), each bit corresponds to the mailbox with the related number. These bits enable or disable transmission/reception complete interrupts for the corresponding mailboxes.

- Bit 0 in MIER corresponds to mailbox 0.
- Bit 31 in MIER corresponds to mailbox 31.

In FIFO mailbox mode, bits 29, 28, 25, and 24 of MIER specify whether transmit/receive FIFO interrupts are enabled/disabled and the timing when interrupt requests are generated.

Write to MIER only when the related MCTLj (j = 0 to 31) is 00h and the corresponding mailbox is not processing a transmission or reception abort request. In FIFO mailbox mode, change the bits in MIER1 for the related FIFO only when the TFE bit in TFCR is 0 and the TFEST bit is 1, and the RFE bit in RFCR is 0 and the RFEST bit is 1.

32.2.8 Message Control Register j (MCTLj) (j = 0 to 31)

Addresses: 0009 0820h to 0009 083Fh

• MCTL.TX

Transmit mode (when the TRMREQ bit is 1 and the RECREQ bit is 0)

b7	b6	b5	b4	b3	b2	b1	b0
TRMREQ	RECREQ	—	ONESHOT	—	TRMABT	TRMACTIVE	SENTDATA

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

• MCTL.RX

Receive mode (when the TRMREQ bit is 0 and the RECREQ bit is 1)

b7	b6	b5	b4	b3	b2	b1	b0
TRMREQ	RECREQ	—	ONESHOT	—	MSGLOST	INVALIDDATA	NEWDATA

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

Bit	Symbol	Bit Name	Description	R/W
b0	SENTDATA	Transmission Complete Flag*1*2	0: Transmission is not completed 1: Transmission is completed	R/W
	NEWDATA	Reception Complete Flag*1*2	0: No data has been received or 0 is written to the NEWDATA bit 1: A new message is being stored or has been stored to the mailbox	R/W
b1	TRMACTIVE	Transmission-in-Progress Status Flag	(Transmit mailbox setting enabled) 0: Transmission is pending or transmission is not requested 1: From acceptance of transmission request to completion of transmission, or error/arbitration lost	R
	INVALIDDATA	Reception-in-Progress Status Flag	(Receive mailbox setting enabled) 0: Message valid 1: Message being updated	R
b2	TRMABT	Transmission Abort Complete Flag*1*2	(Transmit mailbox setting enabled) 0: Transmission has started, transmission abort failed because transmission is completed, or transmission abort is not requested 1: Transmission abort is completed	R/W
	MSGLOST	Message Lost Flag*1*2	(Receive mailbox setting enabled) 0: Message is not overwritten or overrun 1: Message is overwritten or overrun	R/W
b3	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b4	ONESHOT	One-Shot Enable*3	0: One-shot reception or one-shot transmission disabled 1: One-shot reception or one-shot transmission enabled	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b6	RECREQ	Receive Mailbox Request*2*3*4*5	0: Not configured for reception 1: Configured for reception	R/W
b7	TRMREQ	Transmit Mailbox Request*2*4	0: Not configured for transmission 1: Configured for transmission	R/W

Note 1. Write 0 only. Writing 1 has no effect.

Note 2. When writing 0 to bits NEWDATA, SENTDATA, MSGLOST, TRMABT, RECREQ, and TRMREQ by a program, do not use the logic operation instruction (AND.B). Write 0 to only the specified bit and write 1 to the other bits before using the transfer (MOV) instruction.

Note 3. To enter one-shot receive mode, write 1 to the ONESHOT bit at the same time as setting the RECREQ bit to 1. To exit one-shot receive mode, write 0 to the ONESHOT bit after writing 0 to the RECREQ bit and confirming that it has been set to 0.

To enter one-shot transmit mode, write 1 to the ONESHOT bit at the same time as setting the TRMREQ bit to 1.

To exit one-shot transmit mode, write 0 to the ONESHOT bit after the message has been transmitted or aborted.

Note 4. Do not set both the RECREQ and TRMREQ bits to 1.

Note 5. When setting the RECREQ bit to 0, set bits MSGLOST, NEWDATA, and RECREQ to 0 simultaneously.

Write to the MCTLj in CAN operation mode or CAN halt mode.

Do not use MCTL24 to MCTL31 in FIFO mailbox mode.

SENTDATA Flag (Transmission Complete Flag)

The SENTDATA flag is set to 1 when data transmission from the corresponding mailbox is completed. The SENTDATA flag is set to 0 by writing 0 by a program.

To set the SENTDATA flag to 0, first set the TRMREQ bit to 0. Bits SENTDATA and TRMREQ cannot be set to 0 simultaneously. To transmit a new message from the corresponding mailbox, set the SENTDATA flag to 0.

NEWDATA Flag (Reception Complete Flag)

The NEWDATA flag is set to 1 when a new message is being stored or has been stored to the mailbox. The timing for setting this bit to 1 is simultaneous with the INVALIDDATA bit. The NEWDATA flag is set to 0 by writing 0 by a program. The NEWDATA flag cannot be set to 0 by writing 0 by a program while the related INVALIDDATA bit is 1.

TRMACTIVE Bit (Transmission-in-Progress Status Flag)

The TRMACTIVE bit is set to 1 when the corresponding mailbox of the CAN module begins transmitting a message. The TRMACTIVE is set to 0 when the CAN module has lost CAN bus arbitration, a CAN bus error occurs, or data transmission is completed.

INVALIDDATA Bit (Reception-in-Progress Status Flag)

After the completion of a message reception, the INVALIDDATA bit is set to 1 while the received message is being updated into the corresponding mailbox. The INVALIDDATA bit is set to 0 immediately after the message has been stored. If the mailbox is read while the INVALIDDATA bit is 1, the data is undefined.

TRMABT Flag (Transmission Abort Complete Flag)

The TRMABT flag is set to 1 in the following cases:

Following a transmission abort request, when the transmission abort is completed before starting transmission.

Following a transmission abort request, when the CAN module detects CAN bus arbitration lost or a CAN bus error.

In one-shot transmission mode (RECREQ bit = 0, TRMREQ bit = 1, and ONESHOT bit = 1), when the CAN module detects CAN bus arbitration lost or a CAN bus error.

The TRMABT flag is not set to 1 when data transmission is completed. In this case, the SENTDATA flag is set to 1. The TRMABT flag is set to 0 by writing 0 by a program.

MSGLOST Flag (Message Lost Flag)

The MSGLOST flag is set to 1 when the mailbox is overwritten or overrun by a new received message while the NEWDATA flag is 1. The MSGLOST flag is set to 1 at the end of the 6th bit of EOF. The MSGLOST flag is set to 0 by writing 0 by a program.

In both overwrite and overrun modes, the MSGLOST flag cannot be set to 0 by writing 0 by a program during five peripheral module clock (PCLK) cycles following the sixth bit of EOF.

ONESHOT Bit (One-Shot Enable)

The ONESHOT bit can be used in the following two ways, receive mode and transmit mode.

- One-shot receive mode

When the ONESHOT bit is set to 1 in receive mode (RECREQ bit = 1 and TRMREQ bit = 0), the mailbox receives a message only one time. (The mailbox does not behave as a receive mailbox after having received a message one time.) The behavior of bits NEWDATA and INVALIDDATA is the same as in normal receive mode. In one-shot receive mode, the MSGLOST flag is not set to 1. To set the ONESHOT bit to 0, first write 0 to the RECREQ bit and ensure that it has been set to 0.

- One-shot transmit mode

When the ONESHOT bit is set to 1 in transmit mode (RECREQ bit = 0 and TRMREQ bit = 1), the CAN module transmits a message only one time. (The CAN module does not transmit the message again if a CAN bus error or CAN bus arbitration lost occurs.) When transmission is completed, the SENTDATA flag is set to 1. If transmission is not completed due to a CAN bus error or CAN bus arbitration lost, the TRMABT flag is set to 1. Set the ONESHOT bit to 0 after the SENTDATA or TRMABT bit is set to 1.

RECREQ Bit (Receive Mailbox Request)

The RECREQ bit selects receive modes shown in Table 32.11.

When the RECREQ bit is set to 1, the corresponding mailbox is configured for reception of a data frame or a remote frame.

When the RECREQ bit is set to 0, the corresponding mailbox is not configured for reception of a data frame or a remote frame.

Due to hardware protection, the RECREQ bit cannot be set to 0 by writing 0 by a program during the following period.

- Hardware protection is started

From the acceptance filter processing (the beginning of CRC field)

- Hardware protection is released

— For the mailbox that is specified to receive the incoming message, after the received data is stored into the mailbox or a CAN bus error occurs (i.e. maximum period of hardware protection is from the beginning of CRC field to the end of the 7th bit of EOF)

— For the other mailboxes, after the acceptance filter processing

— If no mailbox is specified to receive the message, after the acceptance filter processing

When setting the RECREQ bit to 1, do not set the TRMREQ bit to 1. To change the configuration of a mailbox from transmission to reception, first abort the transmission and then set bits SENTDATA and TRMABT to 0 before changing to reception.

TRMREQ Bit (Transmit Mailbox Request)

The TRMREQ bit selects transmit modes shown in Table 32.11.

When the TRMREQ bit is set to 1, the corresponding mailbox is configured for transmission of a data frame or a remote frame.

When the TRMREQ bit is set to 0, the corresponding mailbox is not configured for transmission of a data frame or a remote frame.

If the TRMREQ bit is changed from 1 to 0 to cancel the corresponding transmission request, either the TRMABT or SENTDATA flag is set to 1 with a maximum delay of one frame. This is used to confirm whether the transmission abort request has been completed or not. When setting the TRMREQ bit to 1, do not set the RECREQ bit to 1. To change the configuration of a mailbox from reception to transmission, first abort the reception and then set bits NEWDATA and MSGLOST to 0 before changing to transmission.

32.2.9 Receive FIFO Control Register (RFCR)

Address: 0009 0848h

b7	b6	b5	b4	b3	b2	b1	b0
RFEST	RFWST	RFFST	RFMLF	RFUST[2:0]			RFE
Value after reset:	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFE	Receive FIFO Enable	0: Receive FIFO disabled 1: Receive FIFO enabled	R/W
b3 to b1	RFUST[2:0]	Receive FIFO Unread Message Number Status	b3 b1 0 0 0: No unread message 0 0 1: 1 unread message 0 1 0: 2 unread messages 0 1 1: 3 unread messages 1 0 0: 4 unread messages 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b4	RFMLF	Receive FIFO Message Lost Flag	0: No receive FIFO message lost has occurred 1: Receive FIFO message lost has occurred	R/W
b5	RFFST	Receive FIFO Full Status Flag	0: Receive FIFO is not full 1: Receive FIFO is full (4 unread messages)	R
b6	RFWST	Receive FIFO Buffer Warning Status Flag	0: Receive FIFO is not buffer warning 1: Receive FIFO is buffer warning (3 unread messages)	R
b7	RFEST	Receive FIFO Empty Status Flag	0: Unread message in receive FIFO 1: No unread message in receive FIFO	R

Write to RFCR in CAN operation mode or CAN halt mode.

RFE Bit (Receive FIFO Enable)

When the RFE bit is set to 1, the receive FIFO is enabled.

When the RFE bit is set to 0, the receive FIFO is disabled for reception and becomes empty (RFEST bit = 1). Write 0 to the RFE bit simultaneously with setting the RFMLF bit.

Do not set this bit to 1 in normal mailbox mode (MBM bit in CTLR = 0). Due to hardware protection, the RFE bit is not set to 0 by writing 0 by a program during the following period.

- Hardware protection is started
 - From the acceptance filter processing (the beginning of CRC field)
- Hardware protection is released
 - If the receive FIFO is specified to receive the incoming message, after the received data is stored into the receive FIFO or a CAN bus error occurs (i.e. maximum period of hardware protection is from the beginning of CRC field to the end of 7th bit of EOF)
 - If the receive FIFO is not specified to receive the message, after the acceptance filter processing

RFUST[2:0] Bits (Receive FIFO Unread Message Number Status)

The RFUST[2:0] bits indicate the number of unread messages in the receive FIFO.

The value of the RFUST[2:0] bits is initialized to 00b when the RFE bit is set to 0.

RFMLF Flag (Receive FIFO Message Lost Flag)

The RFMLF bit is set to 1 (receive FIFO message lost has occurred) when the receive FIFO receives a new message and the receive FIFO is full. The timing for setting this bit to 1 is at the end of the 6th bit of EOF.

The RFMLF bit is set to 0 by writing 0 by a program (writing 1 has no effect). In both overwrite and overrun modes, the RFMLF bit cannot be set to 0 (receive FIFO message lost has not occurred) by writing 0 by a program due to hardware protection during five peripheral module clock (PCLK) cycles following the sixth bit of EOF, if the receive FIFO is full and determined to receive a message.

RFFST Flag (Receive FIFO Full Status Flag)

The RFFST bit is set to 1 (receive FIFO is full) when the number of unread messages in the receive FIFO is 4. The RFFST bit is 0 (receive FIFO is not full) when the number of unread messages in the receive FIFO is less than 4. The RFFST bit is set to 0 when the RFE bit is 0.

RFWST Flag (Receive FIFO Buffer Warning Status Flag)

The RFWST bit is set to 1 (receive FIFO is buffer warning) when the number of unread messages in the receive FIFO is 3. The RFWST bit is 0 (receive FIFO is not buffer warning) when the number of unread messages in the receive FIFO is less than 3 or equal to 4. The RFWST bit is set to 0 when the RFE bit is 0.

RFEST Flag (Receive FIFO Empty Status Flag)

The RFEST bit is set to 1 (no unread message in receive FIFO) when the number of unread messages in the receive FIFO is 0. The RFEST bit is set to 1 when the RFE bit is set to 0. The RFEST bit is set to 0 (unread message in receive FIFO) when the number of unread messages in the receive FIFO is one or more.

Figure 32.2 shows the receive FIFO mailbox operation.

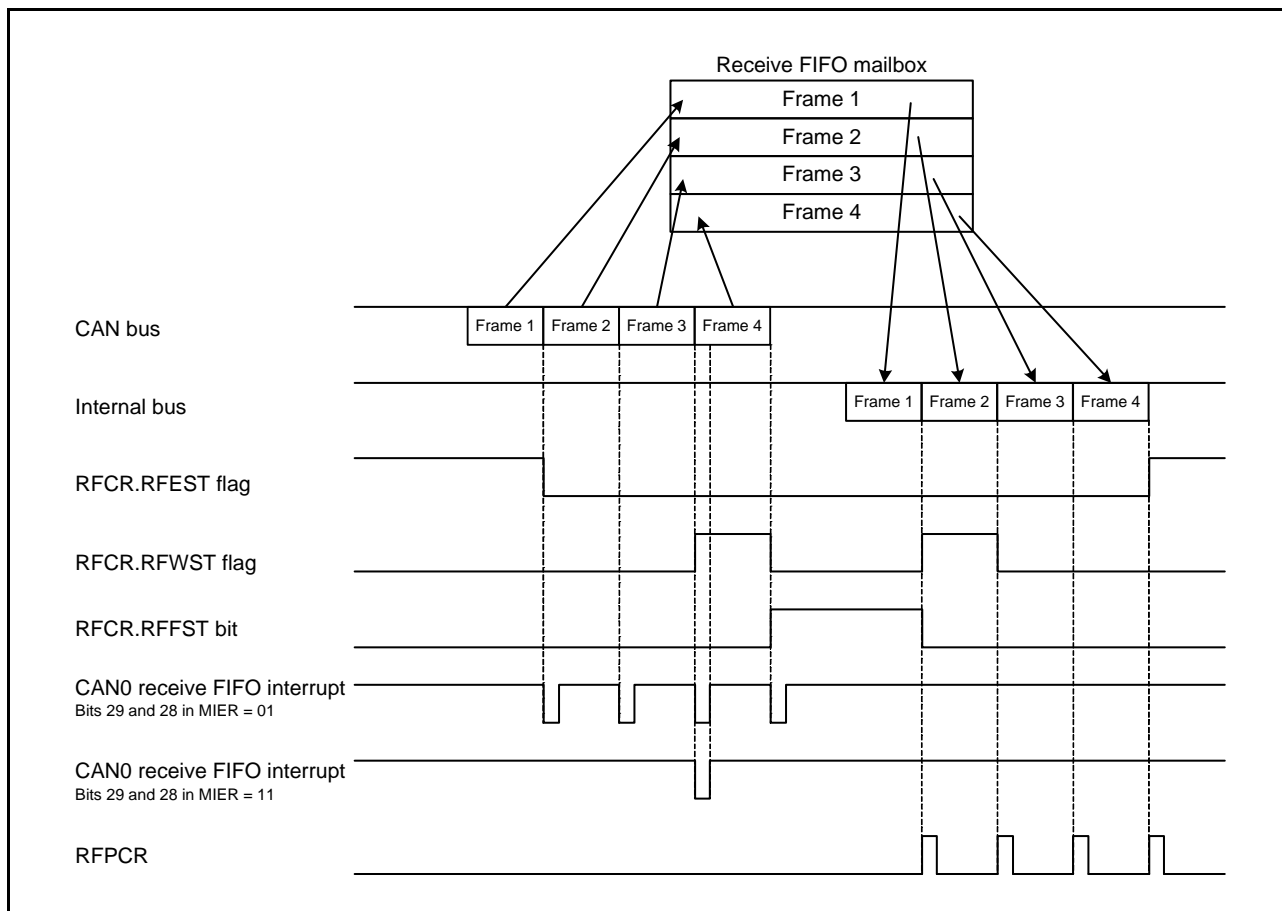


Figure 32.2 Receive FIFO Mailbox Operation (Bits 29 and 28 in MIE = 01 or 11)

32.2.10 Receive FIFO Pointer Control Register (RFPCR)

Address: 0009 0849h

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

[Legend] x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	The CPU-side pointer for the receive FIFO is incremented by writing FFh to RFPCR.	W

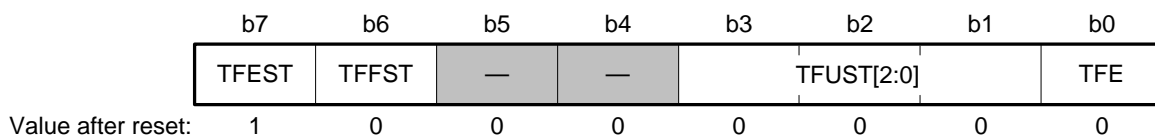
When the receive FIFO is not empty, write FFh to RFPCR by a program to increment the CPU-side pointer for the receive FIFO to the next mailbox location.

Do not write to RFPCR when the RFE bit in RFCR is 0 (receive FIFO disabled).

Both the CAN-side pointer and the CPU-side pointer are incremented when a new message is received and the RFFST bit is 1 (receive FIFO is full) in overwrite mode. When the RFMLF bit is 1 in this condition, the CPU-side pointer cannot be incremented by writing to RFPCR by a program.

32.2.11 Transmit FIFO Control Register (TFCR)

Address: 0009 084Ah



Bit	Symbol	Bit Name	Description	R/W
b0	TFE	Transmit FIFO Enable	0: Transmit FIFO disabled 1: Transmit FIFO enabled	R/W
b3 to b1	TFUST[2:0]	Transmit FIFO Unsent Message Number Status	b3 b1 000: No unsent message 001: 1 unsent message 010: 2 unsent messages 011: 3 unsent messages 100: 4 unsent messages 101: Reserved 110: Reserved 111: Reserved	R
b5, b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b6	TFFST	Transmit FIFO Full Status	0: Transmit FIFO is not full 1: Transmit FIFO is full (4 unsent messages)	R
b7	TFEST	Transmit FIFO Full Status	0: Unsent message in transmit FIFO 1: No unsent message in transmit FIFO	R

Write to TFCR in CAN operation mode or CAN halt mode.

TFE Bit (Transmit FIFO Enable)

When the TFE bit is set to 1, the transmit FIFO is enabled.

When the TFE bit is set to 0, the transmit FIFO becomes empty (TFEST bit = 1) and then unsent messages from the transmit FIFO are lost as described below:

- Immediately if a message from the transmit FIFO is not scheduled for the next transmission or during transmission
- Following the completion of transmission, a CAN bus error, CAN bus arbitration lost, or entry to CAN halt mode if a message from the transmit FIFO is scheduled for the next transmission or already during transmission

Before setting the TFE bit to 1 again, ensure that the TFEST bit has been set to 1. After setting the TFE bit to 1, write transmit data into MB24.

Do not set the TFE bit to 1 in normal mailbox mode (MBM bit in CTRLR = 0).

TFUST[2:0] Bits (Transmit FIFO Unsent Message Number Status)

The TFUST[2:0] bits indicate the number of unsent messages in the transmit FIFO.

The TFUST[2:0] bits are set to 000b when transmission from the transmit FIFO has been aborted.

TFFST Bit (Transmit FIFO Full Status)

The TFFST bit is set to 1 (transmit FIFO is full) when the number of unsent messages in the transmit FIFO is 4. The TFFST bit is set to 0 (transmit FIFO is not full) when the number of unsent messages in the transmit FIFO is less than 4. The TFFST bit is set to 0 when transmission from the transmit FIFO has been aborted.

TFEST Bit (Transmit FIFO Empty Status)

The TFEST bit is set to 1 (no message in transmit FIFO) when the number of unsent messages in the transmit FIFO is 0. The TFEST bit is set to 1 when transmission from the transmit FIFO has been aborted. The TFEST bit is set to 0 (message in transmit FIFO) when the number of unsent messages in the transmit FIFO is not 0.

Figure 32.3 shows the transmit FIFO mailbox operation.

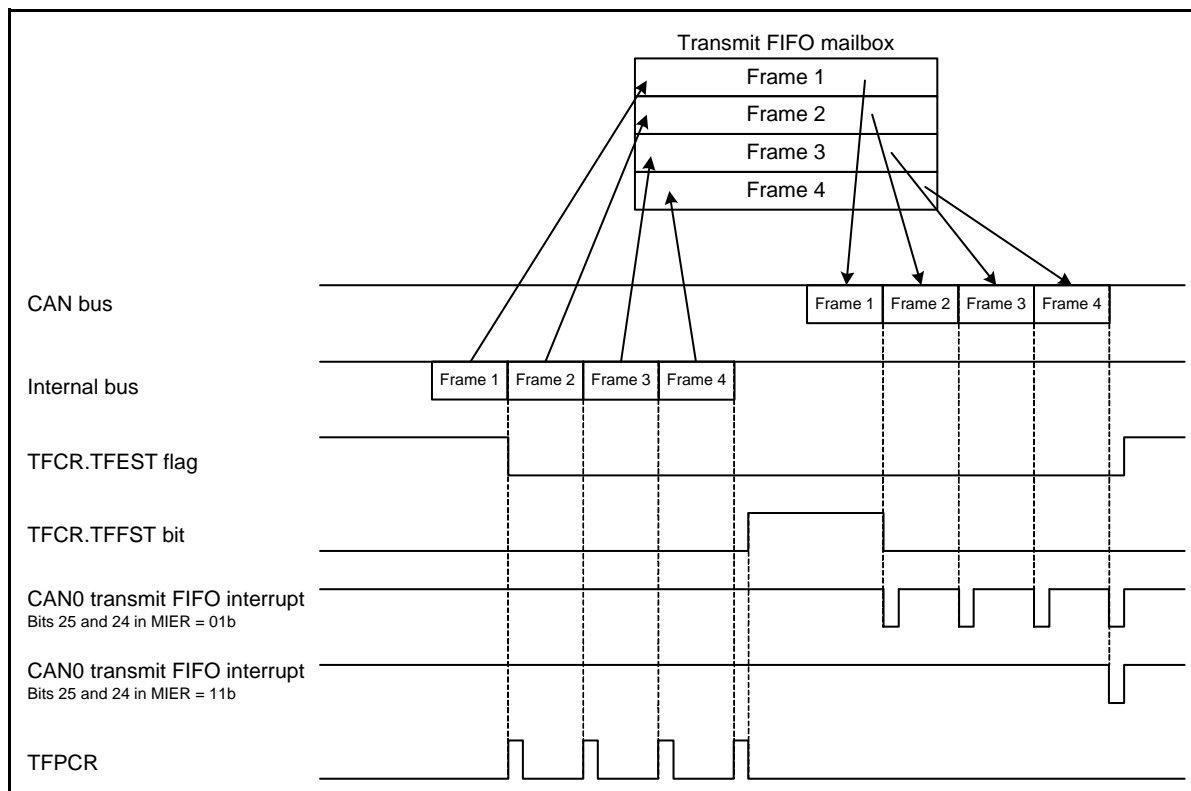


Figure 32.3 Transmit FIFO Mailbox Operation (Bits 25 and 24 in MIER = 01b or 11b)

32.2.12 Transmit FIFO Pointer Control Register (TFPCR)

Address: 0009 084Bh

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

[Legend] x: Undefined

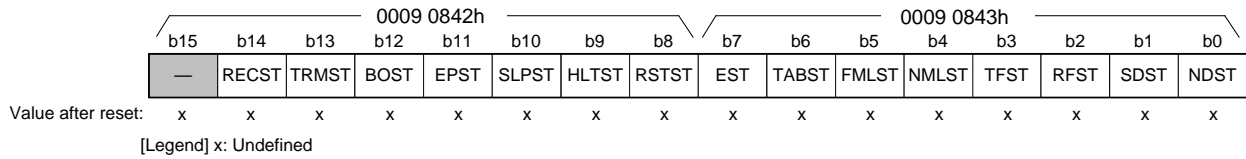
Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	The CPU-side pointer for the transmit FIFO is incremented by writing FFh to TFPCR.	W

When the transmit FIFO is not full, write FFh to TFPCR by a program to increment the CPU-side pointer for the transmit FIFO to the next mailbox location.

Do not write to TFPCR when the TFE bit in TFCR is 0 (transmit FIFO disabled).

32.2.13 Status Register (STR)

Address: 0009 0842h



Bit	Symbol	Bit Name	Description	R/W
b0	NDST	NEWDATA Status Flag	0: No mailbox with NEWDATA bit = 1 1: Mailbox(es) with NEWDATA bit = 1	R
b1	SDST	SENTDATA Status Flag	0: No mailbox with SENTDATA bit = 1 1: Mailbox(es) with SENTDATA bit = 1	R
b2	RFST	Receive FIFO Status Flag	0: No message in receive FIFO (empty) 1: Message in receive FIFO	R
b3	TFST	Transmit FIFO Status Flag	0: Transmit FIFO is full 1: Transmit FIFO is not full	R
b4	NMLST	Normal Mailbox Message Lost Status Flag	0: No mailbox with MSGLOST bit = 1 1: Mailbox(es) with MSGLOST bit = 1	R
b5	FMLST	FIFO Mailbox Message Lost Status Flag	0: RFMLF bit = 0 1: RFMLF bit = 1	R
b6	TABST	Transmission Abort Status Flag	0: No mailbox with TRMABT bit = 1 1: Mailbox(es) with TRMABT bit = 1	R
b7	EST	Error Status Flag	0: No error occurred 1: Error occurred	R
b8	RSTST	CAN Reset Status Flag	0: Not in CAN reset mode 1: In CAN reset mode	R
b9	HLTST	CAN Halt Status Flag	0: Not in CAN halt mode 1: In CAN halt mode	R
b10	SLPST	CAN Sleep Status Flag	0: Not in CAN sleep mode 1: In CAN sleep mode	R
b11	EPST	Error-Passive Status Flag	0: Not in error-passive state 1: In error-passive state	R
b12	BOST	Bus-Off Status Flag	0: Not in bus-off state 1: In bus-off state	R
b13	TRMST	Transmit Status Flag (transmitter)	0: Bus idle or reception in progress 1: Transmission in progress or in bus-off state	R
b14	RECST	Receive Status Flag (receiver)	0: Bus idle or transmission in progress 1: Reception in progress	R
b15	—	Reserved	The read value is undefined. The write value should always be 0.	R

NDST Flag (NEWDATA Status Flag)

The NDST bit is set to 1 when at least one NEWDATA bit in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The NDST bit is set to 0 when all NEWDATA bits are 0.

SDST Flag (SENTDATA Status Flag)

The SDST bit is set to 1 when at least one SENTDATA bit in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The SDST bit is set to 0 when all SENTDATA bits are 0.

RFST Flag (Receive FIFO Status Flag)

The RFST bit is set to 1 when the receive FIFO is not empty. The RFST bit is set to 0 when the receive FIFO is empty or normal mailbox mode is selected.

TFST Flag (Transmit FIFO Status Flag)

The TFST bit is set to 1 when the transmit FIFO is not full. The TFST bit is set to 0 when the transmit FIFO is full or normal mailbox mode is selected.

NMLST Flag (Normal Mailbox Message Lost Status Flag)

The NMLST bit is set to 1 when at least one MSGLOST bit in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The NMLST bit is set to 0 when all MSGLOST bits are 0.

FMLST Flag (FIFO Mailbox Message Lost Status Flag)

The FMLST bit is set to 1 when the RFMLF bit in RFCR is 1 regardless of the value of MIER. The FMLST bit is set to 0 when the RFMLF bit is 0.

TABST Flag (Transmission Abort Status Flag)

The TABST bit is set to 1 when at least one TRMABT bit in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The TABST bit is set to 0 when all TRMABT bits are 0.

EST Flag (Error Status Flag)

The EST bit is set to 1 when at least one error is detected by EIFR regardless of the value of EIER. The EST bit is set to 0 when no error is detected by EIFR.

RSTST Flag (CAN Reset Status Flag)

The RSTST bit is set to 1 when the CAN module is in CAN reset mode. The RSTST bit is 0 when the CAN module is not in CAN reset mode or not in the state from CAN reset mode to CAN sleep mode. Even when the state is changed from CAN reset mode to CAN sleep mode, the RSTST bit remains 1.

HLTST Flag (CAN Halt Status Flag)

The HLTST bit is set to 1 when the CAN module is in CAN halt mode. The HLTST bit is 0 when the CAN module is not in CAN halt mode or not in the state from CAN halt mode to CAN sleep mode. Even when the state is changed from CAN halt mode to CAN sleep mode, the HLTST bit remains 1.

SLPST Flag (CAN Sleep Status Flag)

The SLPST bit is set to 1 when the CAN module is in CAN sleep mode. The SLPST bit is set to 0 when the CAN module is not in CAN sleep mode.

EPST Flag (Error-Passive Status Flag)

The EPST bit is set to 1 when the value of TECR or RECR exceeds 127 and the CAN module is in the error-passive state ($128 \leq \text{TEC} < 256$ or $128 \leq \text{REC} < 256$). The EPST bit is set to 0 when the CAN module is not in the error-passive state.

BOST Flag (Bus-Off Status Flag)

The BOST bit is set to 1 when the value of TECR exceeds 255 and the CAN module is in the bus-off state ($\text{TEC} \geq 256$). The BOST bit is set to 0 when the CAN module is not in the bus-off state.

TRMST Flag (Transmit Status Flag) (transmitter)

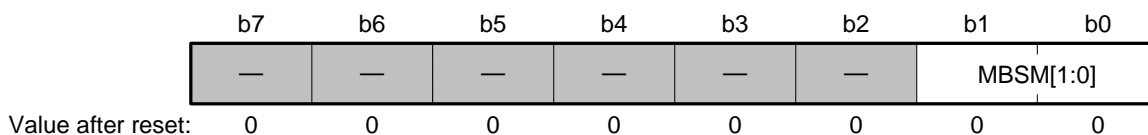
The TRMST bit is set to 1 when the CAN module performs as a transmitter node or is in the bus-off state. The TRMST bit is set to 0 when the CAN module performs as a receiver node or is in the bus-idle state.

RECST Flag (Receive Status Flag) (receiver)

The RECST bit is set to 1 when the CAN module performs as a receiver node. The RECST bit is set to 0 when the CAN module performs as a transmitter node or is in the bus-idle state.

32.2.14 Mailbox Search Mode Register (MSMR)

Address: 0009 0853h



Bit	Symbol	Bit Name	Description	R/W
b1 to b0	MBSM[1:0]	Mailbox Search Mode Select	00: Receive mailbox search mode 01: Transmit mailbox search mode 10: Message lost search mode 11: Channel search mode	R/W
b7 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

Write to MSMR in CAN operation mode or CAN halt mode.

MBSM[1:0] Bits (Mailbox Search Mode Select)

The MBSM[1:0] bits select the search mode for the mailbox search function.

When the MBSM[1:0] bits are 00b, receive mailbox search mode is selected. In this mode, the search targets are the NEWDATA bit in MCTLj (j = 0 to 31) for the normal mailbox and the RFEST bit in RFCR.

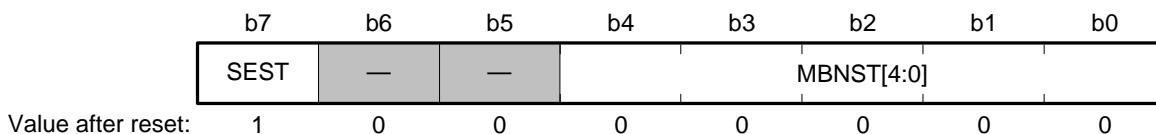
When the MBSM[1:0] bits are 01b, transmit mailbox search mode is selected. In this mode, the search target is the SENTDATA bit in MCTLj.

When the MBSM[1:0] bits are 10b, message lost search mode is selected. In this mode, the search targets are the MSGLOST bit in MCTLj for the normal mailbox and the RFMLF bit in RFCR.

When the MBSM[1:0] bits are 11b, channel search mode is selected. In this mode, the search target is CSSR. Refer to section 32.2.16, Channel Search Support Register (CSSR).

32.2.15 Mailbox Search Status Register (MSSR)

Address: 0009 0852h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MBNST[4:0]	Search Result Mailbox Number Status	These bits output the smallest mailbox number that is searched in each mode of MSMR.	R
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	SEST	Search Result Status	0: Search result found 1: No search result	R

MBNST[4:0] Bits (Search Result Mailbox Number Status)

The MBNST[4:0] bits output the smallest mailbox number that is searched in each mode of MSMR. In receive mailbox search mode, transmit mailbox search mode, and message lost search mode, the value of the mailbox i.e., the search result to be output, is updated as described below:

- When the NEWDATA, SENTDATA or MSGLOST bit for the output mailbox is set to 0
- When the NEWDATA, SENTDATA or MSGLOST bit for a higher-priority mailbox is set to 1

If the MBSM[1:0] bits are set to 00b (receive mailbox search mode) or 10b (message lost search mode), the receive FIFO (mailbox [28]) is output when the receive FIFO is not empty and there are no unread received messages or no lost messages in any of the normal mailboxes (mailboxes [0] to [23]). If the MBSM[1:0] bits are set to 01b (transmit mailbox search mode), the transmit FIFO (mailbox [24]) is not output. Table 32.7 lists the behavior of the MBNST[4:0] bits in FIFO mailbox mode.

In channel search mode, the MBNST[4:0] bits output the corresponding channel number. After MSSR is read by a program, the next target channel number is output.

SEST Bit (Search Result Status)

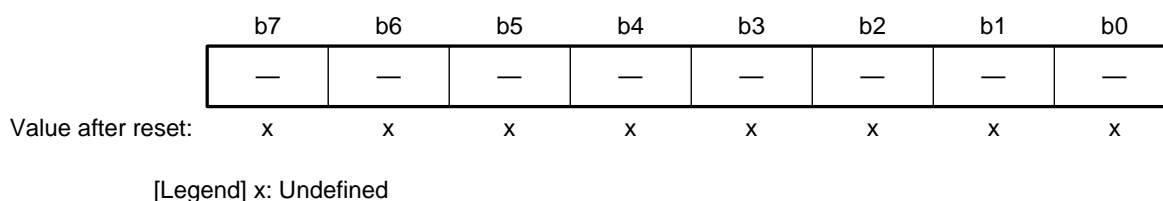
The SEST bit is set to 1 (no search result) when no corresponding mailbox is found after searching all mailboxes. For example, in transmit mailbox search mode, the SEST bit is set to 1 when no SENTDATA bit for mailboxes is 1. The SEST bit is set to 0 when at least one SENTDATA bit is 1. When the SEST bit is 1, the value of the MBNST[4:0] bits is undefined.

Table 32.7 Behavior of MBNST[4:0] Bits in FIFO Mailbox Mode

MBSM[1:0] Bits	Mailbox [24] (Transmit FIFO)	Mailbox [28] (Receive FIFO)
00	Mailbox [24] is not output.	Mailbox [28] is output when no MCTLj.NEWDATA bit for the normal mailboxes is set to 1 (new message is being stored or has been stored to the mailbox) and the receive FIFO is not empty. j = 0 to 23
01		Mailbox [28] is not output.
10		Mailbox [28] is output when no MCTLj.MSGLOST bit for the normal mailboxes is set to 1 (message is overwritten or overrun) and the RFCR.RFMLF bit is set to 1 (receive FIFO message lost has occurred) in the receive FIFO. j = 0 to 23
11		Mailbox [28] is not output.

32.2.16 Channel Search Support Register (CSSR)

Address: 0009 0851h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	When the value for the channel search is input, the channel number is output to MSSR.	R/W

The bits in CSSR, which are set to 1, are encoded by an 8/3 encoder (the LSB position has the higher priority) and output to the MBNST[4:0] bits in MSSR.

MSSR outputs the updated value whenever MSSR is read by a program.

Write to CSSR only when the MBSM[1:0] bits in MSMR are 11b (channel search mode). Do not write to CSSR in CAN reset mode.

Figure 32.4 shows the write and read of CSSR and MSSR.

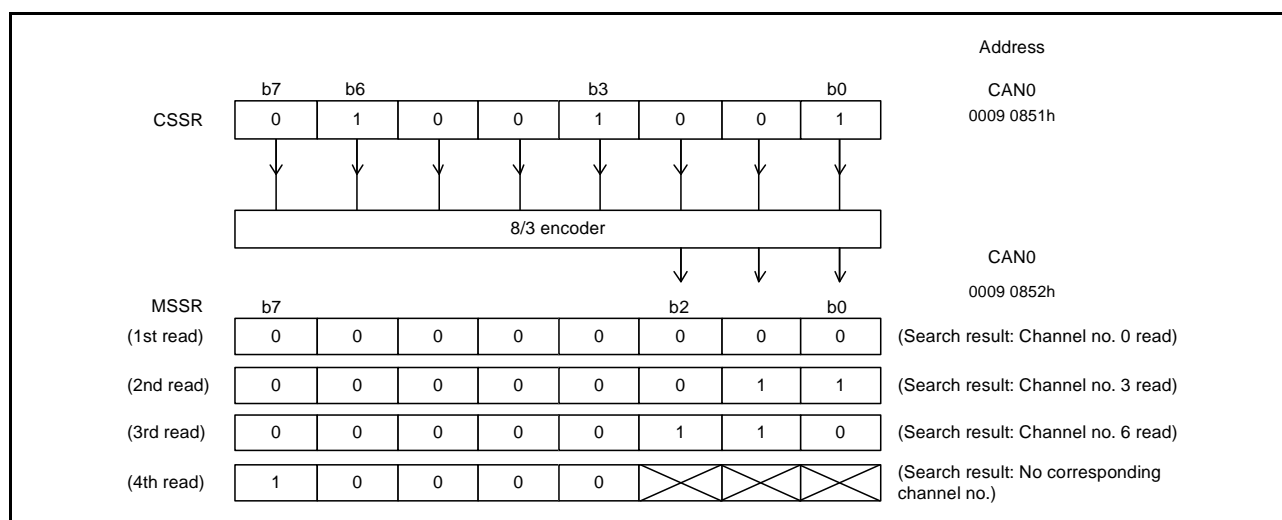
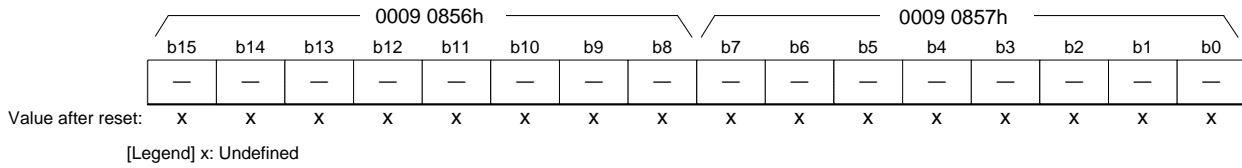


Figure 32.4 Write and Read of CSSR and MSSR

The value of CSSR is also updated whenever MSSR is read. When read, the value prior to conversion by the 8/3 encoder can be read.

32.2.17 Acceptance Filter Support Register (AFSR)

Address: 0009 0856h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	After the standard ID of a received message is written, the value converted for data table search can be read.	R/W

The acceptance filter support unit (ASU) can be used for data table (8 bits × 256) search. In the data table, all standard IDs created by the user are set to be valid/invalid in bit units. When AFSR is written with data in 16-bit units including the SID[10:0] bit in MBj (j = 0 to 31), in which a received standard ID is stored, a decoded row (byte offset) position and column (bit) position for data table search can be read. The ASU can be used for standard (11-bit) IDs only.

The ASU is enabled in the following cases:

- When the ID to receive cannot be masked by the acceptance filter
(Example) IDs to receive: 078h, 087h, and 111h
- When there are too many IDs to receive and software filtering time is expected to be shortened

It should be noted that AFSR cannot be set in CAN reset mode.

Figure 32.5 shows the write and read of AFSR.

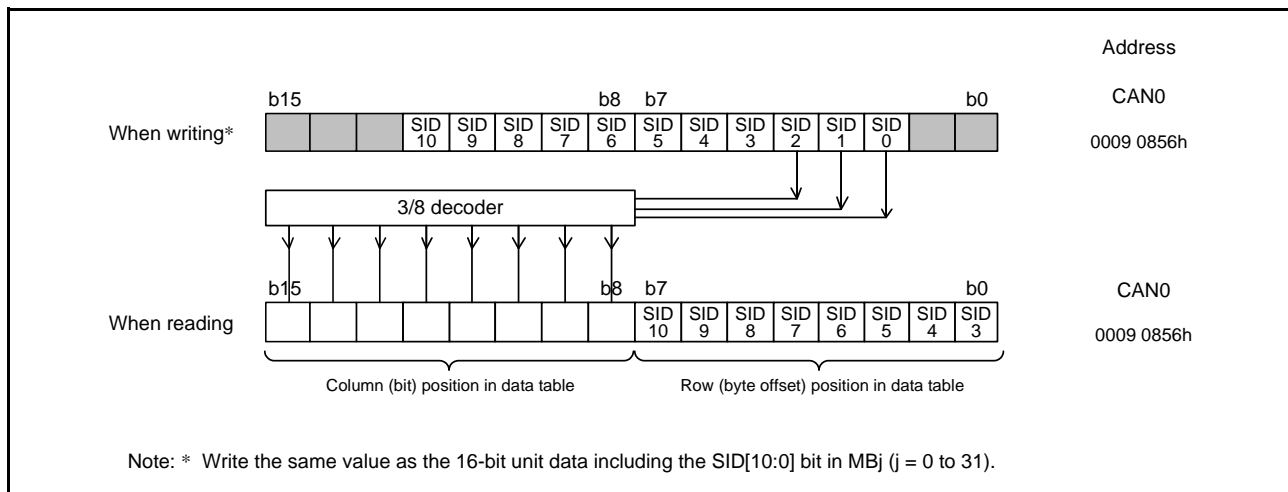


Figure 32.5 Write and Read of AFSR

32.2.18 Error Interrupt Enable Register (EIER)

Address: 0009 084Ch

b7	b6	b5	b4	b3	b2	b1	b0
BLIE	OLIE	ORIE	BORIE	BOEIE	EPIE	EWIE	BEIE

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

Bit	Symbol	Bit Name	Description	R/W
b0	BEIE	Bus Error Interrupt Enable	0: Bus error interrupt disabled 1: Bus error interrupt enabled	R/W
b1	EWIE	Error-Warning Interrupt Enable	0: Error-warning interrupt disabled 1: Error-warning interrupt enabled	R/W
b2	EPIE	Error-Passive Interrupt Enable	0: Error-passive interrupt disabled 1: Error-passive interrupt enabled	R/W
b3	BOEIE	Bus-Off Entry Interrupt Enable	0: Bus-off entry interrupt disabled 1: Bus-off entry interrupt enabled	R/W
b4	BORIE	Bus-Off Recovery Interrupt Enable	0: Bus-off recovery interrupt disabled 1: Bus-off recovery interrupt enabled	R/W
b5	ORIE	Overrun Interrupt Enable	0: Receive overrun interrupt disabled 1: Receive overrun interrupt enabled	R/W
b6	OLIE	Overload Frame Transmit Interrupt Enable	0: Overload frame transmit interrupt disabled 1: Overload frame transmit interrupt enabled	R/W
b7	BLIE	Bus Lock Interrupt Enable	0: Bus lock interrupt disabled 1: Bus lock interrupt enabled	R/W

EIER is used to enable or disable the error interrupt individually for each error interrupt source in EIFR.

Write to EIER in CAN reset mode.

BEIE Bit (Bus Error Interrupt Enable)

When the BEIE bit is 0, no error interrupt request is generated even if the BEIF bit in EIFR is set to 1. When the BEIE bit is 1, an error interrupt request is generated if the BEIF bit is set to 1.

EWIE Bit (Error-Warning Interrupt Enable)

When the EWIE bit is 0, no error interrupt request is generated even if the EWIF bit in EIFR is set to 1. When the EWIE bit is 1, an error interrupt request is generated if the EWIF bit is set to 1.

EPIE Bit (Error-Passive Interrupt Enable)

When the EPIE bit is 0, no error interrupt request is generated even if the EPIF bit in EIFR is set to 1. When the EPIE bit is 1, an error interrupt request is generated if the EPIF bit is set to 1.

BOEIE Bit (Bus-Off Entry Interrupt Enable)

When the BOEIE bit is 0, no error interrupt request is generated even if the BOEIF bit in EIFR is set to 1. When the BOEIE bit is 1, an error interrupt request is generated if the BOEIF bit is set to 1.

BORIE Bit (Bus-Off Recovery Interrupt Enable)

When the BORIE bit is 0, an error interrupt request is not generated even if the BORIF bit in EIFR is set to 1. When the BORIE bit is set to 1, an error interrupt request is generated if the BORIF bit is set to 1.

ORIE Bit (Overrun Interrupt Enable)

When the ORIE bit is 0, an error interrupt request is not generated even if the ORIF bit in EIFR is set to 1. When the ORIE bit is 1, an error interrupt request is generated if the ORIF bit is set to 1.

OLIE Bit (Overload Frame Transmit Interrupt Enable)

When the OLIE bit is 0, no error interrupt request is generated even if the OLIF bit in EIFR is set to 1. When the OLIE bit is 1, an error interrupt request is generated if the OLIF bit is set to 1.

BLIE Bit (Bus Lock Interrupt Enable)

When the BLIE bit is 0, no error interrupt request is generated even if the BLIF bit in EIFR is set to 1. When the BLIE bit is 1, an error interrupt request is generated if the BLIF bit is set to 1.

32.2.19 Error Interrupt Factor Judge Register (EIFR)

Address: 0009 084Dh

b7	b6	b5	b4	b3	b2	b1	b0
BLIF	OLIF	ORIF	BORIF	BOEIF	EPIF	EWIF	BEIF

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

Bit	Symbol	Bit Name	Description	R/W
b0	BEIF	Bus Error Detect Flag	0: No bus error detected 1: Bus error detected	R/W
b1	EWIF	Error-Warning Detect Flag	0: No error-warning detected 1: Error-warning detected	R/W
b2	EPIF	Error-Passive Detect Flag	0: No error-passive detected 1: Error-passive detected	R/W
b3	BOEIF	Bus-Off Entry Detect Flag	0: No bus-off entry detected 1: Bus-off entry detected	R/W
b4	BORIF	Bus-Off Recovery Detect Flag	0: No bus-off recovery detected 1: Bus-off recovery detected	R/W
b5	ORIF	Receive Overrun Detect Flag	0: No receive overrun detected 1: Receive overrun detected	R/W
b6	OLIF	Overload Frame Transmission Detect Flag	0: No overload frame transmission detected 1: Overload frame transmission detected	R/W
b7	BLIF	Bus Lock Detect Flag	0: No bus lock detected 1: Bus lock detected	R/W

If an event corresponding to each bit occurs, the corresponding bit in EIFR is set to 1 regardless of the setting of EIER. To set each bit to 0, write 0 by a program. If the set timing occurs simultaneously with the clear timing by the program, the bit becomes 1.

When a single bit is set to 0 by a program, do not use the logic operation instruction (AND.B) – use the transfer instruction (MOV) to ensure that only the specified bit is set to 0 and the other bits are set to 1. Writing 1 has no effect to these bit values.

BEIF Flag (Bus Error Detect Flag)

The BEIF bit is set to 1 when a bus error is detected.

EWIF Flag (Error-Warning Detect Flag)

The EWIF bit is set to 1 when the value of the receive error counter (REC) or transmit error counter (TEC) exceeds 95.

The EWIF bit is set to 1 only when the REC or TEC initially exceeds 95. Thus, if 0 is written to the EWIF bit by a program while the REC or TEC remains greater than 95, the EWIF bit is not set to 1 until the REC or TEC goes below 95 and then exceeds 95 again.

EPIF Flag (Error-Passive Detect Flag)

The EPIF bit is set to 1 when the CAN error state becomes error-passive (the REC (receive error counter) or TEC (transmit error counter) value exceeds 127).

The EPIF bit is set to 1 only when the REC or TEC initially exceeds 127. Thus, if 0 is written by a program while the REC or TEC remains greater than 127, the EPIF bit is not set to 1 until the REC or TEC goes below 127 and then exceeds 127 again.

BOEIF Flag (Bus-Off Entry Detect Flag)

The BOEIF bit is set to 1 when the CAN error state becomes bus-off (the TEC (transmit error counter) value exceeds 255). The BOEIF bit is also set to 1 when the BOM[1:0] bits in CTLR are 01b (entry to CAN halt mode automatically at bus-off entry) and the CAN module becomes the bus-off state.

BORIF Flag (Bus-Off Recovery Detect Flag)

The BORIF bit is set to 1 when the CAN module recovers from the bus-off state normally by detecting 11 consecutive bits 128 times in the following conditions:

- When the BOM[1:0] bits in CTLR are 00b
- When the BOM[1:0] bits in CTLR are 10b
- When the BOM[1:0] bits in CTLR are 11b

However, the BORIF bit is not set to 1 if the CAN module recovers from the bus-off state in the following conditions:

- When the CANM[1:0] bits in CTLR are set to 01b or 11b (CAN reset mode)
- When the RBOC bit in CTLR is set to 1 (forcible return from bus-off)
- When the BOM[1:0] bits in CTLR are set to 01b
- When the BOM[1:0] bits in CTLR are set to 11b and the CANM[1:0] bits in CTLR are set to 10b (CAN halt mode) before normal recovery occurs

ORIF Flag (Receive Overrun Detect Flag)

The ORIF bit is set to 1 when a receive overrun occurs. This bit is not set to 1 in overwrite mode.

In overwrite mode, a reception complete interrupt request is generated if an overwrite condition occurs and the ORIF bit is not set to 1.

In normal mailbox mode, if an overrun occurs in any of mailboxes [0] to [31] in overrun mode, this bit is set to 1. In FIFO mailbox mode, if an overrun occurs in any of mailboxes [0] to [23] or the receive FIFO in overrun mode, this bit is set to 1.

OLIF Flag (Overload Frame Transmission Detect Flag)

The OLIF bit is set to 1 if the transmitting condition of an overload frame is detected when the CAN module performs transmission or reception.

BLIF Flag (Bus Lock Detect Flag)

The BLIF bit is set to 1 if 32 consecutive dominant bits are detected on the CAN bus while the CAN module is in CAN operation mode.

After the BLIF bit is set to 1, 32 consecutive dominant bits are detected again under either of the following conditions:

- After this bit is set to 0 from 1, recessive bits are detected
- After this bit is set to 0 from 1, the CAN module enters CAN reset mode or CAN halt mode and then enters CAN operation mode again.

Table 32.8 lists the behavior of bits BOEIF and BORIF according to the CTLR.BOM[1:0] bit setting.

Table 32.8 Behavior of BOEIF and BORIF Flags according to CTLR.BOM[1:0] Bit Setting

BOM[1:0] Bits	BOEIF Flag	BORIF Flag
00	Set to 1 on entry to the bus-off state.	Set to 1 on exit from the bus-off state.
01		Do not set to 1.
10		Set to 1 on exit from the bus-off state.
11		Set to 1 if normal bus-off recovery occurs before the CANM[1:0] bits are set to 10b (CAN halt mode).

32.2.20 Receive Error Count Register (RECR)

Address: 0009 084Eh

	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
b7 to b0	—	Receive Error Count Function	RECR increments or decrements the counter value according to the error status of the CAN module during reception.	R

RECR indicates the value of the receive error counter.

Refer to the CAN Specifications (ISO11898-1) about the increment/decrement conditions of the receive error counter.

The value of RECR in the bus-off state is undefined.

32.2.21 Transmit Error Count Register (TECR)

PGBSEL (big endian) area address: 0009 084Fh

	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
b7 to b0	—	Transmit Error Count Function	TECR increments or decrements the counter value according to the error status of the CAN module during transmission.	R

TECR indicates the value of the transmit error counter.

Refer to the CAN Specifications (ISO11898-1) about the increment/decrement conditions of the transmit error counter.

The value of TECR in the bus-off state is undefined.

32.2.22 Error Code Store Register (ECSR)

PGBSEL (big endian) area address: 0009 0850h

b7	b6	b5	b4	b3	b2	b1	b0
EDPM	ADEF	BE0F	BE1F	CEF	AEF	FEF	SEF

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

Bit	Symbol	Bit Name	Description	R/W
b0	SEF	Stuff Error Flag*3*4	0: No stuff error detected 1: Stuff error detected	R/W
b1	FEF	Form Error Flag*3*4	0: No form error detected 1: Form error detected	R/W
b2	AEF	ACK Error Flag*3*4	0: No ACK error detected 1: ACK error detected	R/W
b3	CEF	CRC Error Flag*3*4	0: No CRC error detected 1: CRC error detected	R/W
b4	BE1F	Bit Error (recessive) Flag*3*4	0: No bit error (recessive) detected 1: Bit error (recessive) detected	R/W
b5	BE0F	Bit Error (dominant) Flag*3*4	0: No bit error (dominant) detected 1: Bit error (dominant) detected	R/W
b6	ADEF	ACK Delimiter Error Flag*3*4	0: No ACK delimiter error detected 1: ACK delimiter error detected	R/W
b7	EDPM	Error Display Mode Select*1*2	0: Output of first detected error code 1: Output of accumulated error code	R/W

Note 1. Writing 1 has no effect to these bit values.

Note 2. To write 0 to bits SEF, FEF, AEF, CEF, BE1F, BE0F, and ADEF, do not use the logic operation instruction (AND.B). Use the transfer (MOV) instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1.

Note 3. Write to the EDPM bit in CAN reset mode or CAN halt mode.

Note 4. If more than one error condition is detected simultaneously, all related bits are set to 1.

ECSR can be used to monitor whether an error has occurred on the CAN bus.

Refer to the CAN Specifications (ISO11898-1) to check the generation conditions of each error.

To set each bit except for the EDPM bit to 0, write 0 by a program. If the timing at which each bit is set to 1 and the timing at which 0 is written by a program are the same, the relevant bit is set to 1.

SEF Flag (Stuff Error Flag)

The SEF bit is set to 1 when a stuff error is detected.

FEF Flag (Form Error Flag)

The FEF bit is set to 1 when a form error is detected.

AEF Flag (ACK Error Flag)

The AEF bit is set to 1 when an ACK error is detected.

CEF Flag (CRC Error Flag)

The CEF bit is set to 1 when a CRC error is detected.

BE1F Flag (Bit Error (recessive) Flag)

The BE1F bit is set to 1 when a recessive bit error is detected.

BE0F Flag (Bit Error (dominant) Flag)

The BE0F bit is set to 1 when a dominant bit error is detected.

ADEF Flag (ACK Delimiter Error Flag)

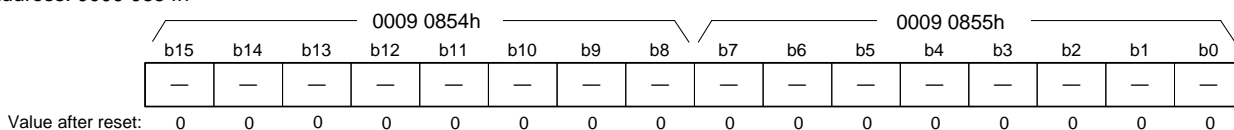
The ADEF bit is set to 1 when a form error is detected with the ACK delimiter during transmission.

EDPM Bit (Error Display Mode Select)

The EDPM bit selects the output mode of ECSR. When the EDPM bit is set to 0, ECSR outputs the first error code. When the EDPM bit is set to 1, ECSR outputs the accumulated error code.

32.2.23 Time Stamp Register (TSR)

Address: 0009 0854h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Time Stamp Register	Free-running counter value for the time stamp function	R

Note : Read TSR in 16-bit units.

When TSR is read, the value of the time stamp counter (16-bit free-running counter) at that moment is read.

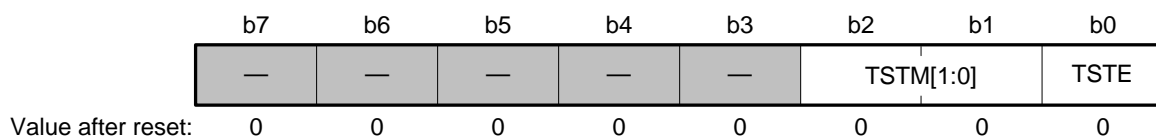
The value of the time stamp counter reference clock is a multiple of 1 bit time, as configured by the TSPS[1:0] bits in CTLR.

The time stamp counter stops in CAN sleep mode and CAN halt mode, and is initialized in CAN reset mode.

The time stamp counter value is stored to bits TSL[7:0] and TSH[7:0] in MB when a received message is stored in a receive mailbox.

32.2.24 Test Control Register (TCR)

Address: 0009 0858h



Bit	Symbol	Bit Name	Description	R/W
b0	TSTE	CAN Test Mode Enable	0: CAN test mode disabled 1: CAN test mode enabled	R/W
b2, b1	TSTM[1:0]	CAN Test Mode Select	b2 b1 0 0: Other than CAN test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback) 1 1: Self-test mode 1 (internal loopback)	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

TCR controls the CAN test mode. Write to TCR in CAN halt mode only.

(1) Listen-Only Mode

The CAN Specifications (ISO11898-1) recommend an optional bus monitoring mode. In listen-only mode, valid data frames and valid remote frames can be received. However, only recessive bits can be sent on the CAN bus, and the ACK bit, overload flag, and active error flag cannot be sent.

Listen-only mode can be used for baud rate detection.

Do not request transmission from any mailboxes in listen-only mode.

Figure 32.6 shows the connection when listen-only mode is selected.

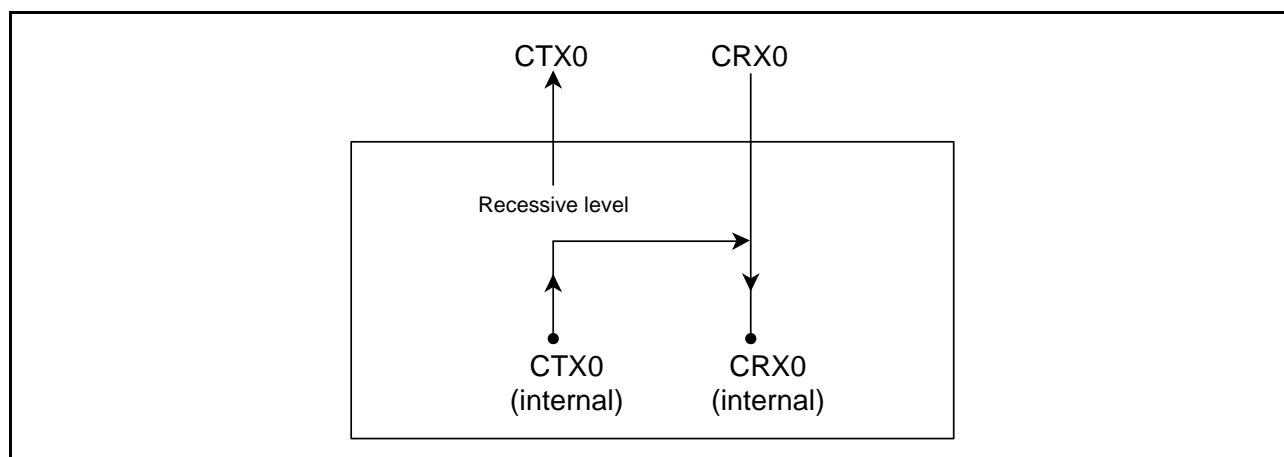


Figure 32.6 Connection when Listen-Only Mode is Selected

(2) Self-Test Mode 0 (External Loopback)

Self-test mode 0 is for testing the CAN transceiver (self-diagnostic function under a self-loaded condition). Connect the CTX0 and CRX0 pins to the CAN transceiver when this mode is to be used.

As a node, this module is capable of transmitting its own ACK bits in self-test mode 0, making self-diagnostic tests to check for normal CAN operation possible even when no other node is connected to the network.

Figure 32.7 shows the connection when self-test mode 0 is selected.

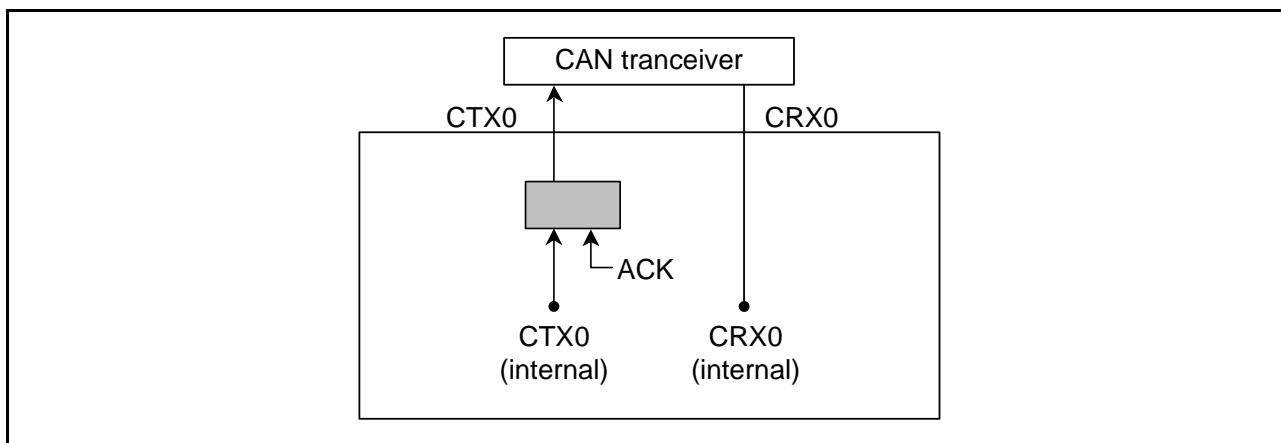


Figure 32.7 Connection when Self-Test Mode 0 is Selected

(3) Self-Test Mode 1 (Internal Loopback)

Self-test mode 1 is provided for self-test functions.

In self-test mode 1, the protocol controller treats its transmitted messages as received messages and stores them into the receive mailbox. To be independent from external stimulation, the protocol controller generates the ACK bit.

In self-test mode 1, the protocol controller performs an internal feedback from the internal CTX0 pin to the internal CRX0 pin. The input value of the external CRX0 pin is ignored. The external CTX0 pin outputs only recessive bits. The CTX0 and CRX0 pins do not need to be connected to the CAN bus or any external device.

Figure 32.8 shows the connection when self-test mode 1 is selected.

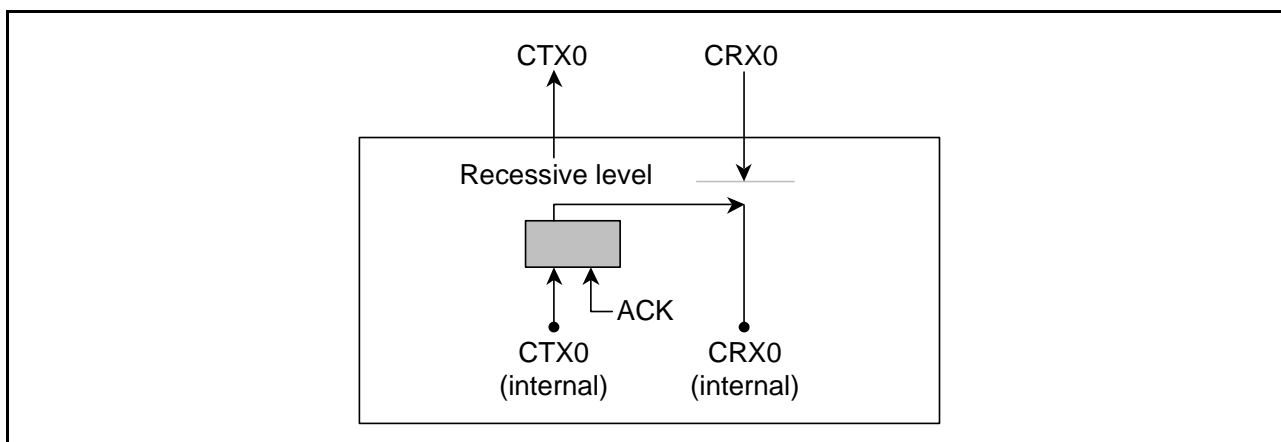


Figure 32.8 Connection when Self-Test Mode 1 is Selected

32.3 Operating Mode

The CAN module has the following four operating modes.

- CAN reset mode
- CAN halt mode
- CAN operation mode
- CAN sleep mode

Figure 32.9 shows the transition between CAN operating modes.

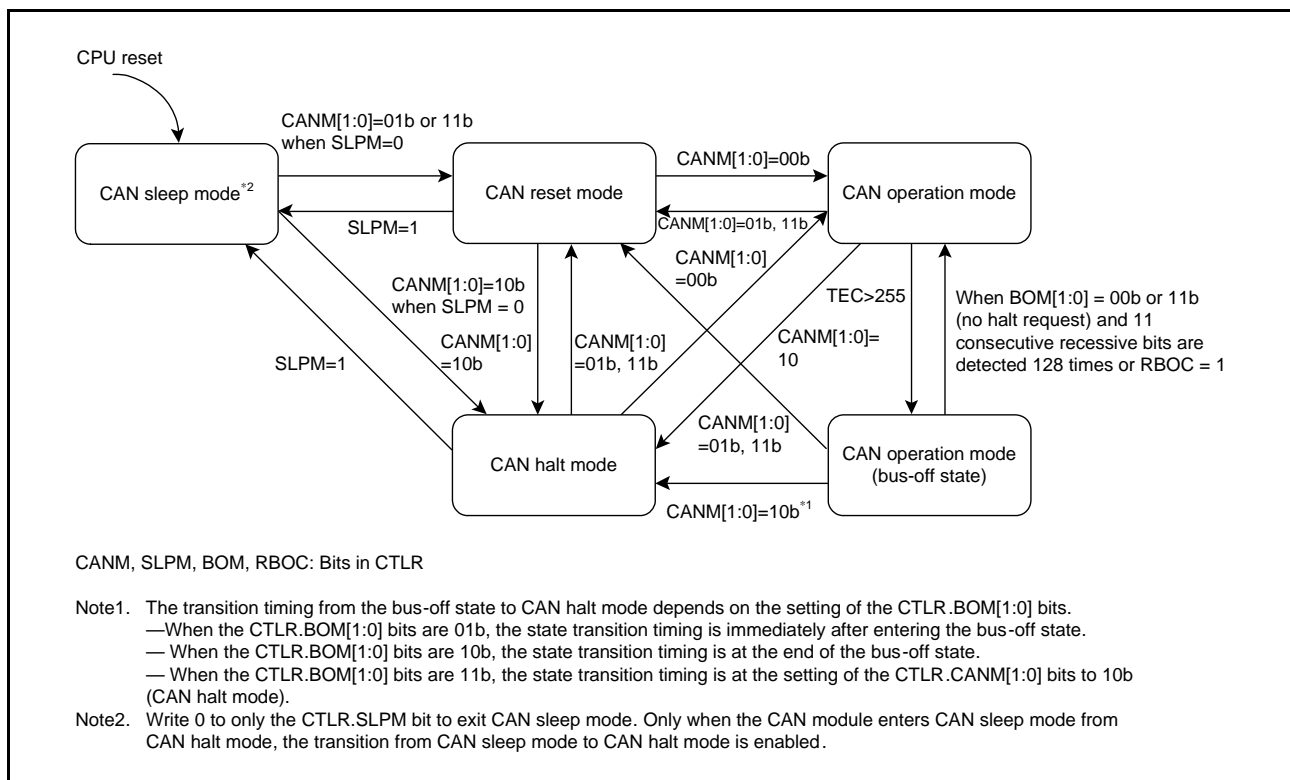


Figure 32.9 Transition between CAN Operating Modes

32.3.1 CAN Reset Mode

CAN reset mode is provided for CAN communication configuration.

When the CTLR.CANM[1:0] bits are set to 01b or 11b, the CAN module enters CAN reset mode. Then, the STR.RSTST bit is set to 1. Do not change the CTLR.CANM[1:0] bits until the RSTST bit is set to 1. Set BCR before exiting CAN reset mode to any other modes.

The following registers are initialized to their reset values after entering CAN reset mode, and their initial values are retained during CAN reset mode:

- MCTLj (j = 0 to 31)
- STR (except for the SLPST and TFST bits)
- EIFR
- RECR
- TECR
- TSR
- MSSR
- MSMR
- RFCR
- TFCR
- TCR
- ECSR (except for the EDPM bit)

The following registers retain their previous values even after entering CAN reset mode.

- CTLR
- STR (only the SLPST and TFST bits)
- MIER
- EIER
- BCR
- CSSR
- ECSR (only the EDPM bit)
- MBj
- MKR0 to MKR7
- FIDCR0 and FIDCR1
- MKIVLR
- AFSR
- RFPCR
- TFPCR

32.3.2 CAN Halt Mode

CAN halt mode is used for mailbox configuration and test mode setting.

When the CTLR.CANM[1:0] bits are set to 10, CAN halt mode is selected. Then the STR.HLTST bit is set to 1. Do not change the CTLR.CANM[1:0] bits until the HLTST bit is set to 1.

See Table 32.9 for the state transition conditions when transmitting or receiving.

All registers except for bits RSTST, HLTST, and SLPST in STR remain unchanged when the CAN enters CAN halt mode.

Do not change CTLR (except for bits CANM[1:0] and SLPM) and EIER in CAN halt mode. BCR can be changed in CAN halt mode only when listen-only mode is selected for automatic baud rate detection.

Table 32.9 Operation in CAN Reset Mode and CAN Halt Mode

Mode	Receiver	Transmitter	Bus-Off
CAN reset mode (forcible transition) CANM[1:0] = 11b	CAN module enters CAN reset mode without waiting for the end of message reception.	CAN module enters CAN reset mode without waiting for the end of message transmission.	CAN module enters CAN reset mode without waiting for the end of bus-off recovery.
CAN reset mode CANM[1:0] = 01b	CAN module enters CAN reset mode without waiting for the end of message reception.	CAN module enters CAN reset mode after waiting for the end of message transmission.*1*4	CAN module enters CAN reset mode without waiting for the end of bus-off recovery.
CAN halt mode	CAN module enters CAN halt mode after waiting for the end of message reception.*2*3	CAN module enters CAN halt mode after waiting for the end of message transmission.*1*4	[When the BOM[1:0] bits are 00b] A halt request from a program will be accepted only after bus-off recovery. [When the BOM[1:0] bits are 01b] CAN module automatically enters CAN halt mode without waiting for the end of bus-off recovery (regardless of a halt request from a program). [When the BOM[1:0] bits are 10b] CAN module automatically enters CAN halt mode after waiting for the end of bus-off recovery (regardless of a halt request from a program). [When the BOM[1:0] bits are 11b] CAN module enters CAN halt mode (without waiting for the end of bus-off recovery) if a halt is requested by a program during bus-off.

[Legend]

BOM[1:0] bits: Bits in CTLR

Note 1. If several messages are requested to be transmitted, mode transition occurs after the completion of the first transmission. In a case that the CAN reset mode is being requested during suspend transmission, mode transition occurs when the bus is idle, the next transmission ends, or the CAN module becomes a receiver.

Note 2. If the CAN bus is locked at the dominant level, the program can detect this state by monitoring the BLIF bit in EIFR.

Note 3. If a CAN bus error occurs during reception after CAN halt mode is requested, the CAN module transits to CAN halt mode immediately.

Note 4. If a CAN bus error or arbitration lost occurs during transmission after CAN reset mode or CAN halt mode is requested, the CAN module transits to the requested CAN mode immediately.

32.3.3 CAN Sleep Mode

CAN sleep mode is used for reducing current consumption by stopping the clock supply to the CAN module. After an MCU pin reset or software reset, the CAN module starts from CAN sleep mode.

When the SLPM bit in CTLR is set to 1, the CAN module enters CAN sleep mode. Then, the SLPST bit in STR is set to 1. Do not change the value of the SLPM bit until the SLPST bit is set to 1. The other registers remain unchanged when the CAN module enters CAN sleep mode.

Write to the SLPM bit in CAN reset mode and CAN halt mode. Do not change any registers (except for the SLPM bit) during CAN sleep mode. Read operation is still allowed.

When the SLPM bit is set to 0, the CAN module is released from CAN sleep mode. When the CAN module exits CAN sleep mode, the other registers remain unchanged.

32.3.4 CAN Operation Mode (Excluding Bus-Off State)

CAN operation mode is used for CAN communication.

When the CANM[1:0] bits in CTLR are set to 00b, the CAN module enters CAN operation mode.

Then bits RSTST and HLTST in STR are set to 0. Do not change the value of the CANM[1:0] bits until bits RSTST and HLTST are set to 0.

If 11 consecutive recessive bits are detected after entering CAN operation mode, the CAN module is in the following states:

- The CAN module becomes an active node on the network, thus enabling transmission and reception of CAN messages.
- Error monitoring of the CAN bus, such as receive and transmit error counters, is performed.

During CAN operation mode, the CAN module may be in one of the following three sub-modes, depending on the status of the CAN bus.

- Idle mode: Transmission or reception is not being performed.
- Receive mode: A CAN message sent by another node is being received.
- Transmit mode: A CAN message is being transmitted. The CAN module may receive its own message simultaneously when self-test mode 0 (TSTM bits in TCR = 10b) or self-test mode 1 (TSTM bits = 11b) is selected.

Figure 32.10 shows the sub-modes of CAN operation mode.

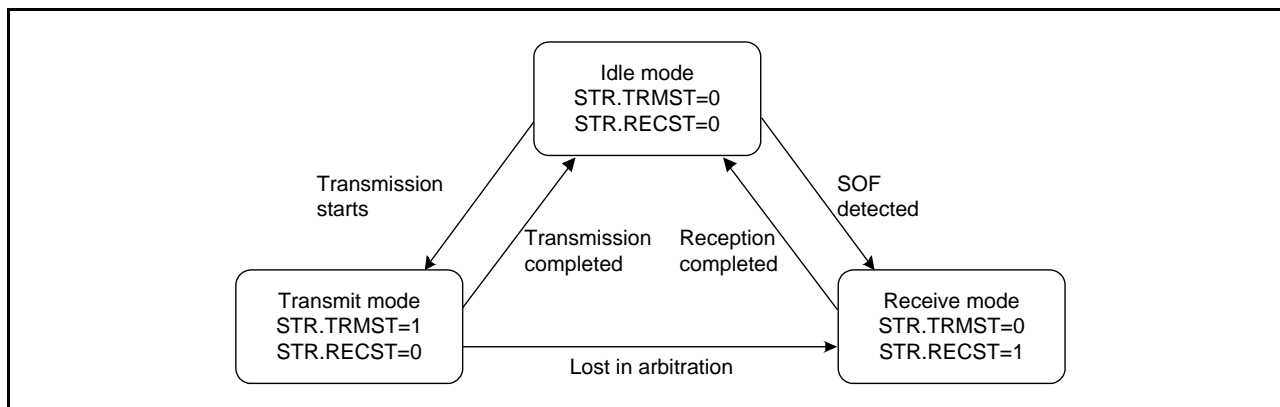


Figure 32.10 Sub-Modes of CAN Operation Mode

32.3.5 CAN Operation Mode (Bus-Off State)

The CAN module enters the bus-off state according to the increment/decrement rules for the transmit/error counters in the CAN Specifications.

The following cases apply when the CAN module is recovering from the bus-off state. When the CAN module is in the bus-off state, the values of the CAN-related registers, except for STR, EIFR, RECR, TECR and TSR, remain unchanged.

(1) When bits BOM[1:0] in CTLR are 00b (normal mode)

The CAN module enters the error-active state immediately after it has completed the recovery from the bus-off state and CAN communication is enabled instantly. The BORIF flag in EIFR is set to 1 (bus-off recovery detected) at this time.

(2) When bit RBOC in CTLR is set to 1 (forcible return from bus-off)

The CAN module enters the error-active state immediately when it is in the bus-off state and the RBOC bit is set to 1. CAN communication is enabled again after 11 consecutive recessive bits are detected. The BORIF bit is not set to 1 at this time.

(3) When bits BOM[1:0] are 01b (automatic transition to CAN halt mode at bus-off entry)

The CAN module enters CAN halt mode immediately when it reaches the bus-off state. The BORIF bit is not set to 1 at this time.

(4) When bits BOM[1:0] are 10b (automatic transition to CAN halt mode at bus-off end)

The CAN module enters CAN halt mode when it has completed the recovery from bus-off. The BORIF bit is set to 1 at this time.

(5) When bits BOM[1:0] are 11b (automatic transition to CAN halt mode by a program) and bits CANM[1:0] in CTLR are set to 10b (CAN halt mode) during bus-off state

The CAN module enters CAN halt mode immediately when it is in the bus-off state and the CANM[1:0] bits are set to 10b (CAN halt mode). The BORIF bit is not set to 1 at this time.

If the CANM[1:0] bits are not set to 10b during bus-off, the same behavior as (1) applies.

32.4 CAN Communication Speed Setting

The following description explains about CAN communication speed setting.

32.4.1 CAN Clock Setting

This LSI has a CAN clock selector.

The CAN clock can be set by the BRP[9:0] bits in BCR.

Figure 32.11 shows a block diagram of the CAN clock generator.

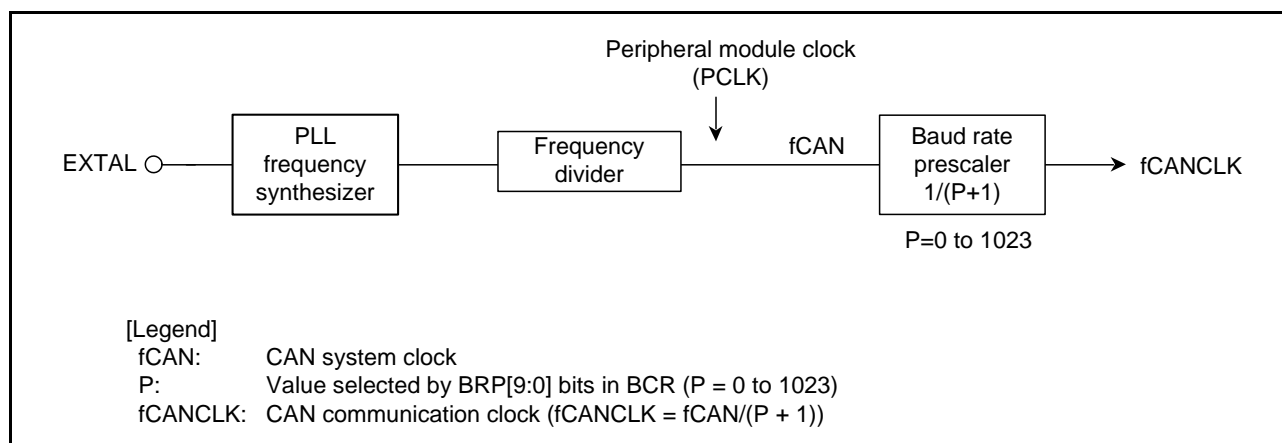


Figure 32.11 Block Diagram of CAN Clock Generator

32.4.2 Bit Timing Setting

The bit time consists of the following three segments.

Figure 32.12 shows the bit timing.

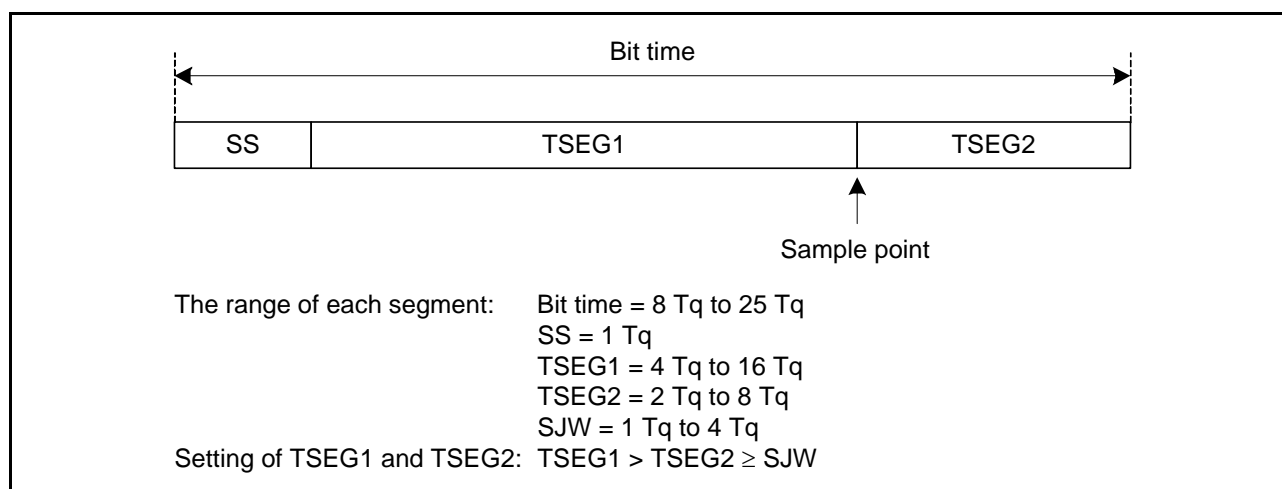


Figure 32.12 Bit Timing

32.4.3 Bit Rate

The bit rate depends on the division value of f_{CAN} (CAN system clock), the division value of the baud rate prescaler, and the number of T_q of one bit time.

$$\text{Bit rate [bps]} = \frac{f_{CAN}}{\text{Baud rate prescaler division value} * \text{x number of } T_q \text{ of one bit time}} = \frac{f_{CANCLK}}{\text{Number of } T_q \text{ of one bit time}}$$

Note: * Division value of baud rate prescaler = $P + 1$ (P: 0 to 1023)
Setting of the BRP[9:0] bits in BCR

Table 32.10 lists bit rate examples.

Table 32.10 Bit Rate Examples

fCAN	50 MHz		48 MHz		40 MHz		32 MHz	
	Number of T_q	P + 1	Number of T_q	P + 1	Number of T_q	P + 1	Number of T_q	P + 1
1 Mbps	10 T_q	5	8 T_q	6	10 T_q	4	8 T_q	4
	25 T_q	2	12 T_q	4	20 T_q	2	16 T_q	2
500 kbps	10 T_q	10	8 T_q	12	10 T_q	8	8 T_q	8
	25 T_q	4	12 T_q	8	20 T_q	4	16 T_q	4
250 kbps	10 T_q	20	8 T_q	24	10 T_q	16	8 T_q	16
	25 T_q	8	12 T_q	16	20 T_q	8	16 T_q	8
125 kbps	10 T_q	40	8 T_q	48	10 T_q	32	8 T_q	32
	25 T_q	16	12 T_q	32	20 T_q	16	16 T_q	16
83.3 kbps	10 T_q	60	8 T_q	72	8 T_q	60	8 T_q	48
	25 T_q	24	12 T_q	48	10 T_q	48	16 T_q	24
33.3 kbps	10 T_q	150	8 T_q	180	8 T_q	150	8 T_q	120
	25 T_q	60	12 T_q	120	10 T_q	120	10 T_q	96
			16 T_q	90	20 T_q	60	16 T_q	60
							20 T_q	48

32.5 Mailbox and Mask Register Structure

Figure 32.13 shows the structure of MBj.

There are 32 mailboxes with the same structure.

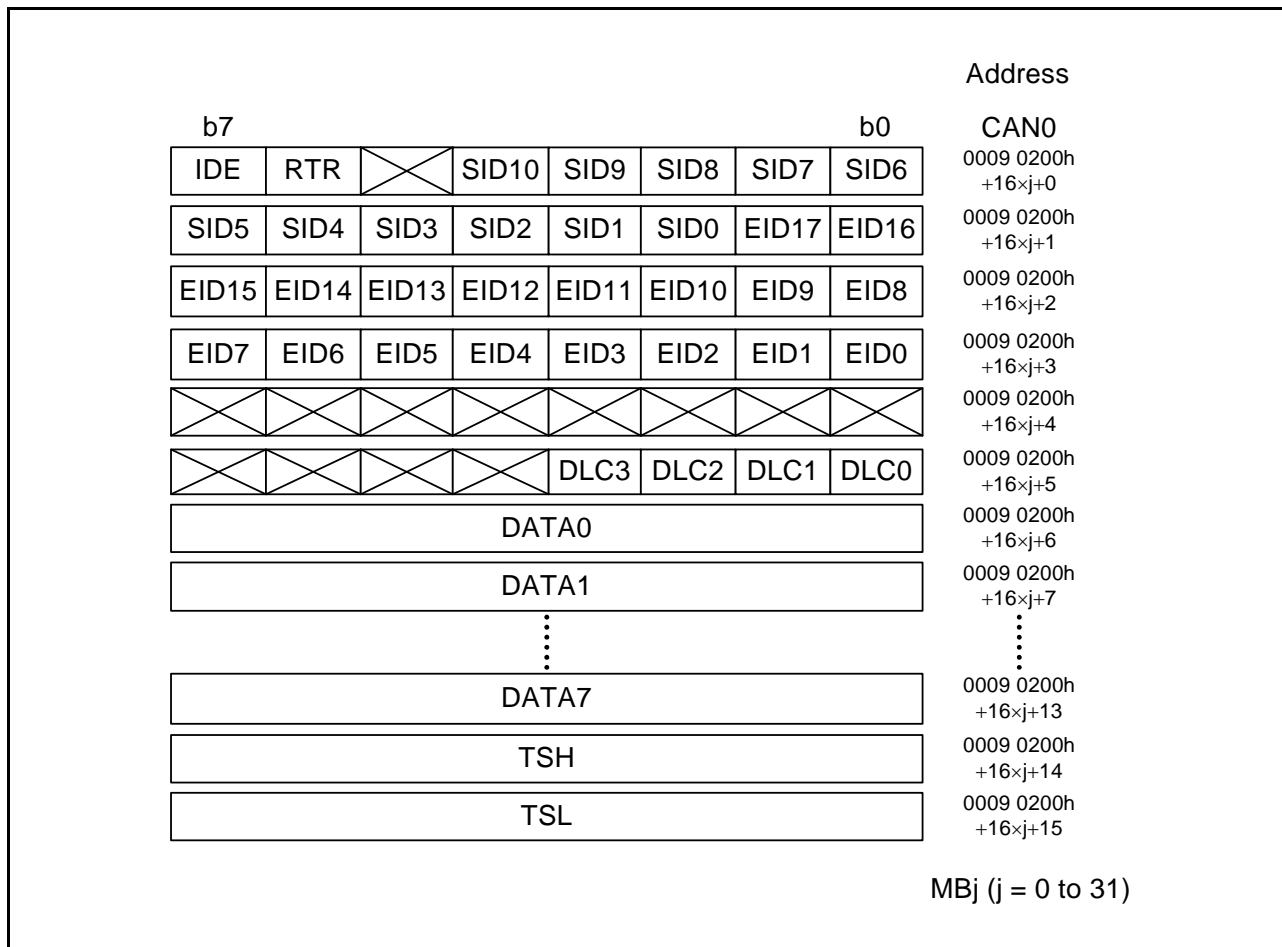


Figure 32.13 Structure of MBj (j = 0 to 31)

Figure 32.14 shows the structure of MKRk.

There are eight mask registers with the same structure.

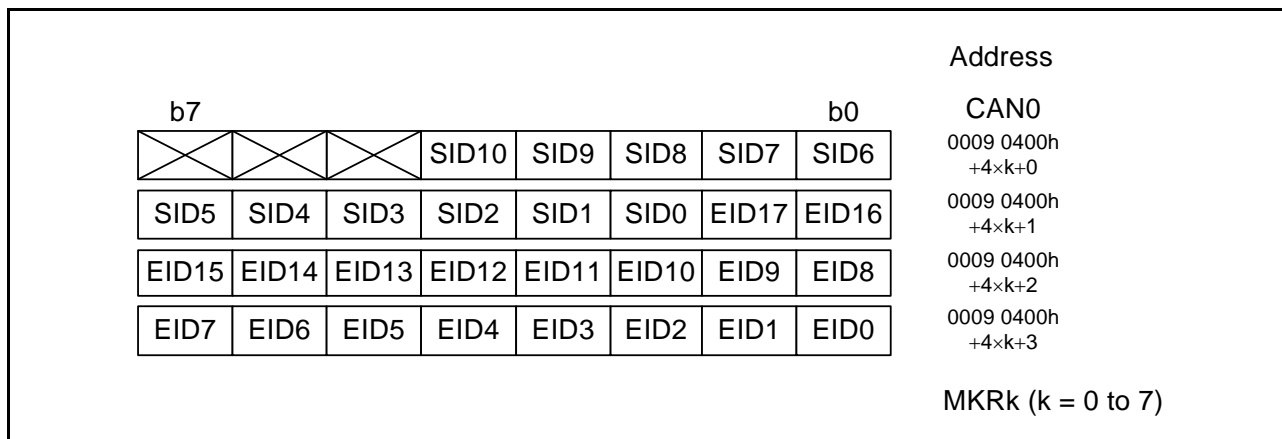


Figure 32.14 Structure of MKRk (k = 0 to 7)

Figure 32.15 shows the structure of FIDCR0 and FIDCR1.

There are two FIFO received ID compare registers with the same structure.

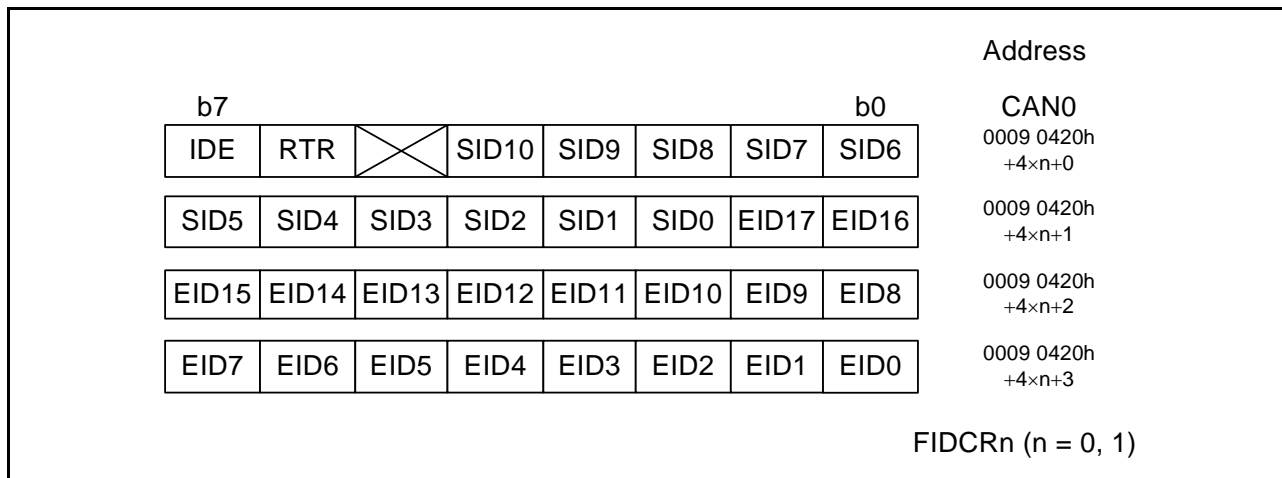


Figure 32.15 Structure of FIDCRn (n = 0, 1)

32.6 Acceptance Filtering and Masking Functions

The acceptance filtering function and masking function allows the user to receive messages with a specified range of multiple IDs for mailboxes.

Registers MKR0 to MKR7 can perform masking of the standard ID and the extended ID of 29 bits.

- MKR0 corresponds to mailboxes [0] to [3]
- MKR1 corresponds to mailboxes [4] to [7]
- MKR2 corresponds to mailboxes [8] to [11]
- MKR3 corresponds to mailboxes [12] to [15]
- MKR4 corresponds to mailboxes [16] to [19]
- MKR5 corresponds to mailboxes [20] to [23]
- MKR6 corresponds to mailboxes [24] to [27] in normal mailbox mode and the receive FIFO mailboxes [28] to [31] in FIFO mailbox mode.
- MKR7 corresponds to mailboxes [28] to [31] in normal mailbox mode and the receive FIFO mailboxes [28] to [31] in FIFO mailbox mode.

MKIVLR disables acceptance filtering individually for each mailbox.

The IDE bit in MBj is valid when the IDFM[1:0] bits in CTLR are 10 (mixed ID mode).

The RTR bit in MBj selects a data frame or a remote frame.

In FIFO mailbox mode, normal mailboxes (mailboxes [0] to [23]) use the single corresponding register among MKR0 to MKR5 for acceptance filtering. Receive FIFO mailboxes (mailboxes [28] to [31]) use two registers MKR6 and MKR7 for acceptance filtering.

Also, the receive FIFO uses two registers FIDCR0 and FIDCR1 for ID comparison. Bits EID[17:0], SID[10:0], RTR, and IDE in MB28 to MB31 for the receive FIFO are disabled. As acceptance filtering depends on the result of two logic AND operations, two ranges of IDs can be received into the receive FIFO.

MKIVLR is disabled for the receive FIFO.

If both the standard ID and extended ID are set in the IDE bits in FIDCR0 and FIDCR1 individually, both ID formats are received.

If both the data frame and remote frame are set in the RTR bits in FIDCR0 and FIDCR1 individually, both data and remote frames are received.

When combination with two ranges of IDs is not necessary, set the same mask value and the same ID into both the FIFO ID and mask register.

Figure 32.16 shows the correspondence between mask registers and mailboxes. Figure 32.17 shows acceptance filtering.

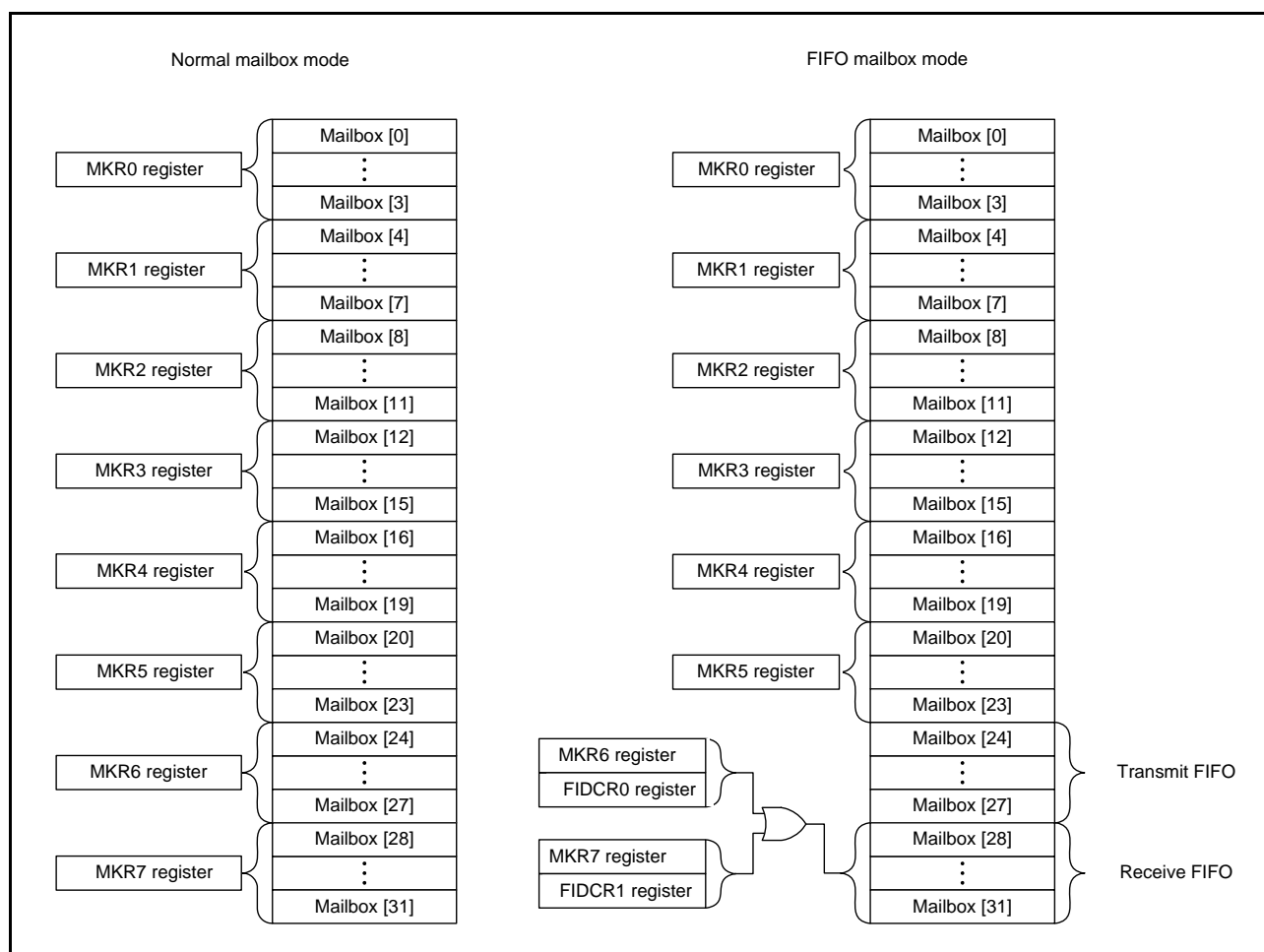


Figure 32.16 Correspondence between Mask Registers and Mailboxes

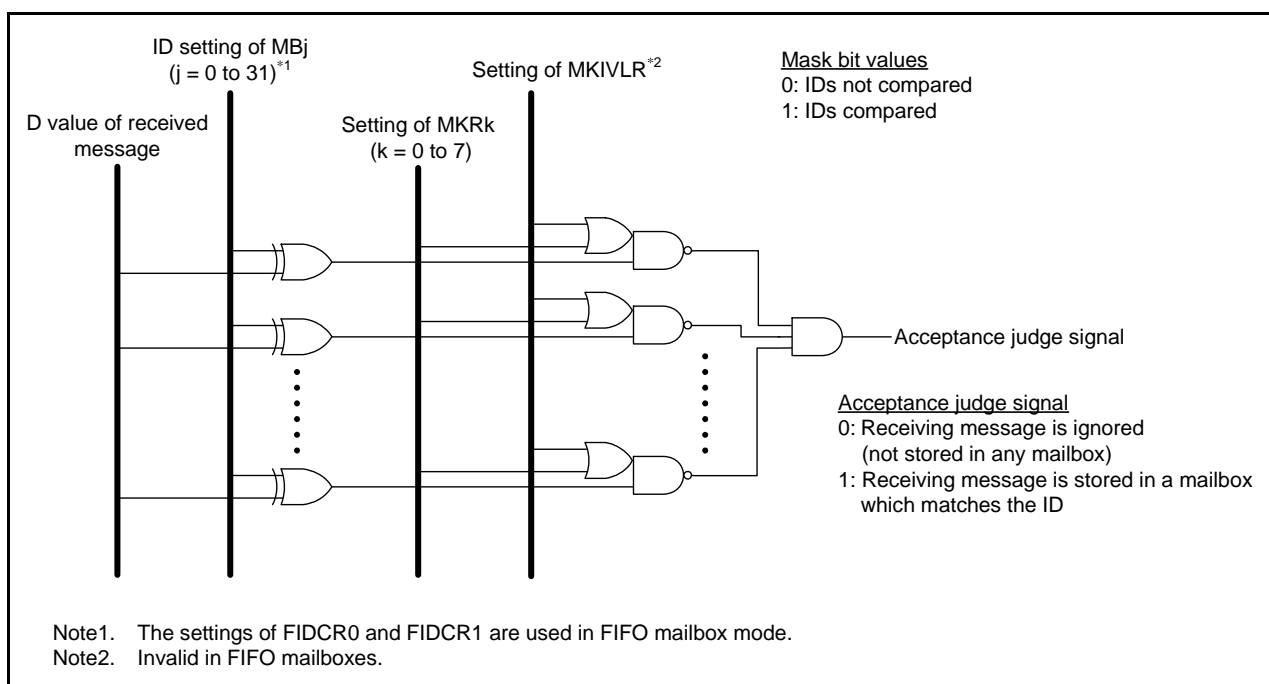


Figure 32.17 Acceptance Filtering

32.7 Reception and Transmission

Table 32.11 shows how to make the CAN communication mode settings.

Table 32.11 Setting of CAN Receive Mode and CAN Transmit Mode

MCTLj. TRMREQ	MCTLj. RECREQ	MCTLj. ONESHOT	Communication Mode of Mailbox
0	0	0	Mailbox disabled or transmission being aborted.
0	0	1	Can be configured only when transmission or reception from a mailbox programmed in one-shot mode is aborted.
0	1	0	Configured as a receive mailbox for a data frame or a remote frame.
0	1	1	Configured as a one-shot receive mailbox for a data frame or a remote frame.
1	0	0	Configured as a transmit mailbox for a data frame or a remote frame.
1	0	1	Configured as a one-shot transmit mailbox for a data frame or a remote frame.
1	1	0	Do not set.
1	1	1	Do not set.

[Legend]

j = 0 to 31

When a mailbox is configured as a receive mailbox or a one-shot receive mailbox, note the following:

1. Before a mailbox is configured as a receive mailbox or a one-shot receive mailbox, set MCTLj to 00h.
2. A received message is stored into the first mailbox that matches the condition according to the result of receive mode setting and acceptance filtering. Upon deciding the mailbox to store the received message, the mailbox with the smaller number has higher priority.
3. In CAN operation mode, when the CAN module transmits a message whose ID matches with the ID/mask set of a mailbox configured to receive messages, the CAN module never receives the transmitted data. In self-test mode, however, the CAN module will receive its transmitted data. In this case, the CAN module returns ACK.

When configuring a mailbox as a transmit mailbox or a one-shot transmit mailbox, note the following:

1. Before a mailbox is configured as a transmit mailbox or a one-shot transmit mailbox, ensure that MCTLj is 00h and that there is no pending abort process.

32.7.1 Reception

Figure 32.18 shows an operation example of data frame reception in overwrite mode.

This example shows the operation of overwriting the first message when the CAN module receives two consecutive CAN messages which match the receiving conditions of MCTLj (j = 0 to 31).

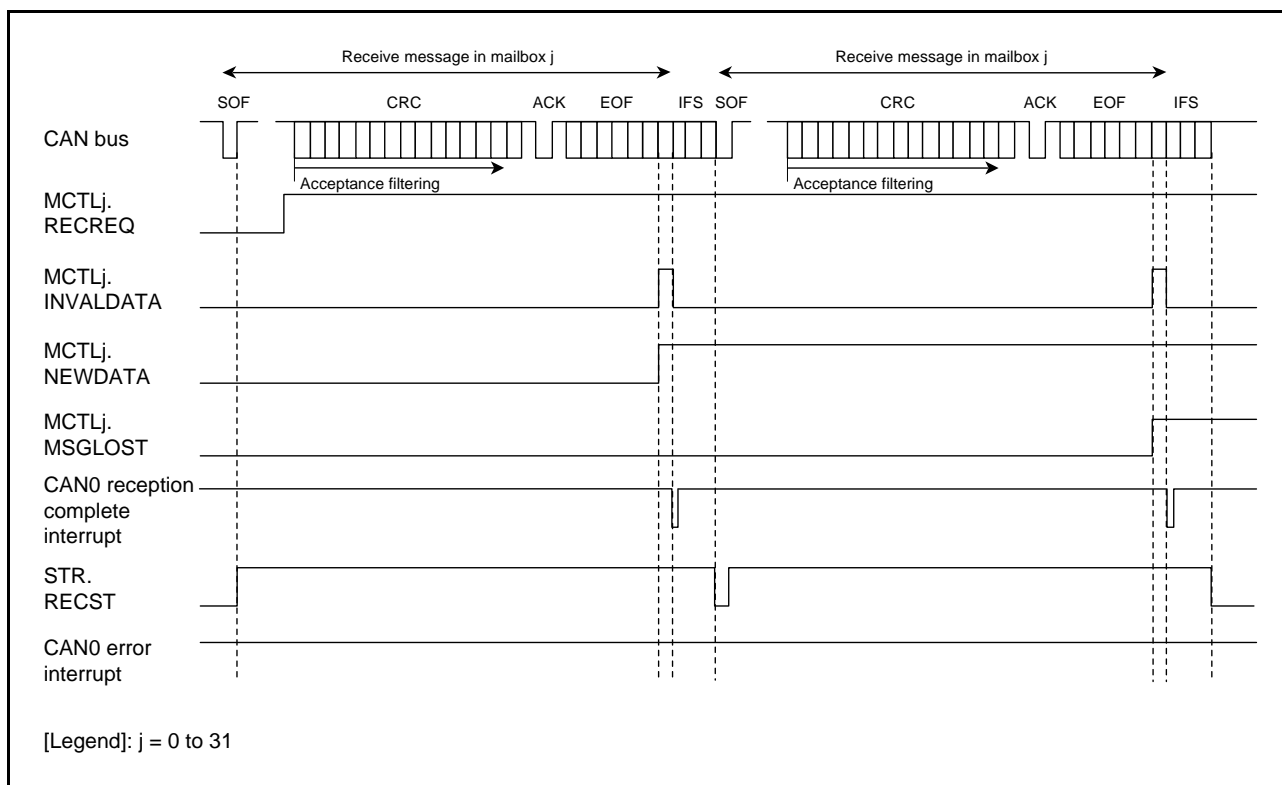


Figure 32.18 Operation Example of Data Frame Reception in Overwrite Mode

1. When an SOF is detected on the CAN bus, the RECST bit in STR is set to 1 (reception in progress) immediately if the CAN module has no message ready to start transmission.
2. The acceptance filter processing starts at the beginning of the CRC field to select the receive mailbox.
3. After a message has been received, the NEWDATA bit in MCTLj for the receive mailbox is set to 1 (new message is being stored or has been stored to the mailbox). The INVALIDATA bit in MCTLj is set to 1 (message is being updated) at the same time, and then the INVALIDATA bit is set to 0 (message valid) again after the complete message is transferred to the mailbox.
4. When the interrupt enable bit in MIER for the receive mailbox is 1 (interrupt enabled), the CAN0 reception complete interrupt request is generated. This interrupt (CAN0 reception complete interrupt) is generated when the INVALIDATA bit is set to 0.
5. After reading the message from the mailbox, the NEWDATA bit needs to be set to 0 by a program.
6. In overwrite mode, if the next CAN message has been received into a mailbox whose NEWDATA bit is still set to 1, the MSGLOST bit in MCTLj is set to 1 (message has been overwritten). The new received message is transferred to the mailbox. The CAN0 reception complete interrupt request is generated the same as in 4.

Figure 32.19 shows the operation example of data frame reception in overrun mode.

This example shows the operation of overrunning the second message when the CAN module receives two consecutive CAN messages which match the receiving conditions of MCTLj (j = 0 to 31).

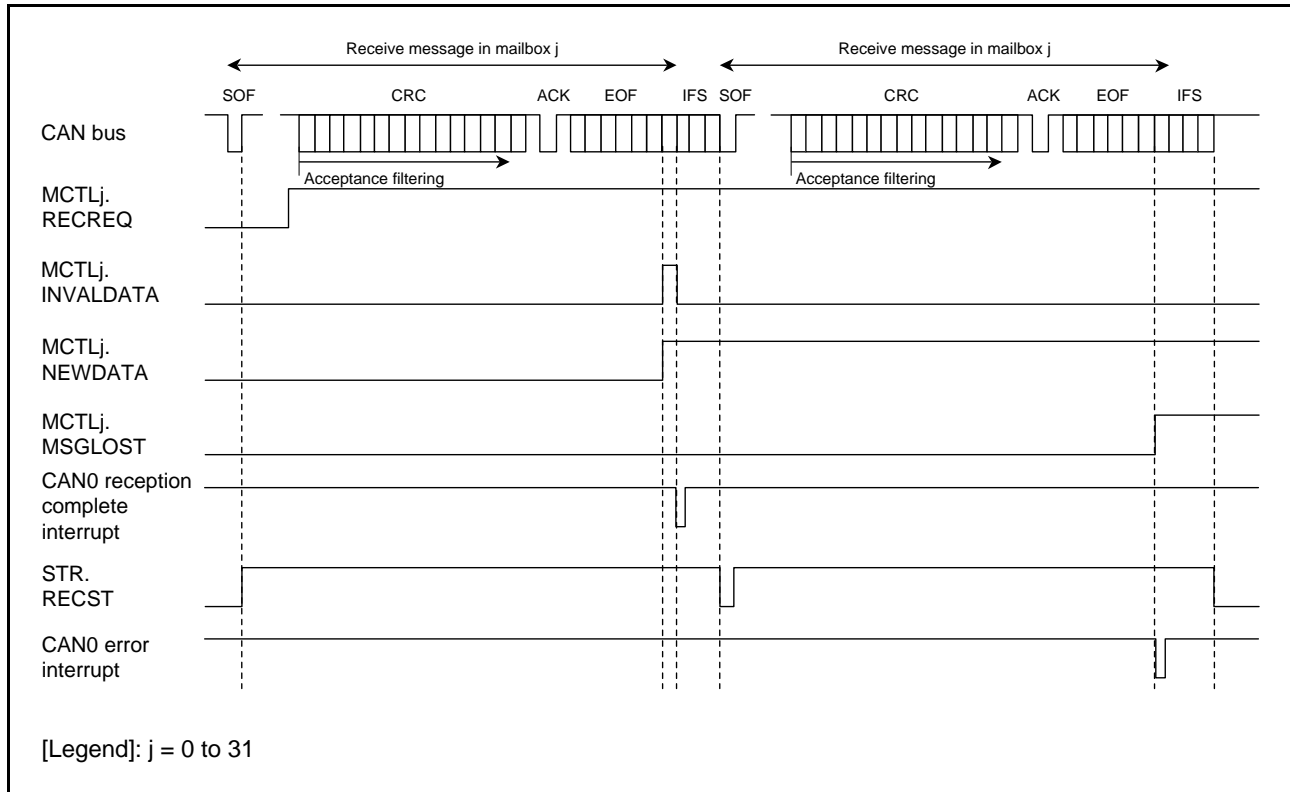


Figure 32.19 Operation Example of Data Frame Reception in Overrun Mode

1. to 5. are the same as in overwrite mode.
6. In overrun mode, if the next CAN message has been received before the NEWDATA bit in MCTLj is set to 0, the MSGLOST bit in MCTLj is set to 1 (message has been overrun). The new received message is discarded and a CAN0 error interrupt request is generated if the corresponding interrupt enable bit in EIER is set to 1 (interrupt enabled).

32.7.2 Transmission

Figure 32.20 shows an operation example of data frame transmission.

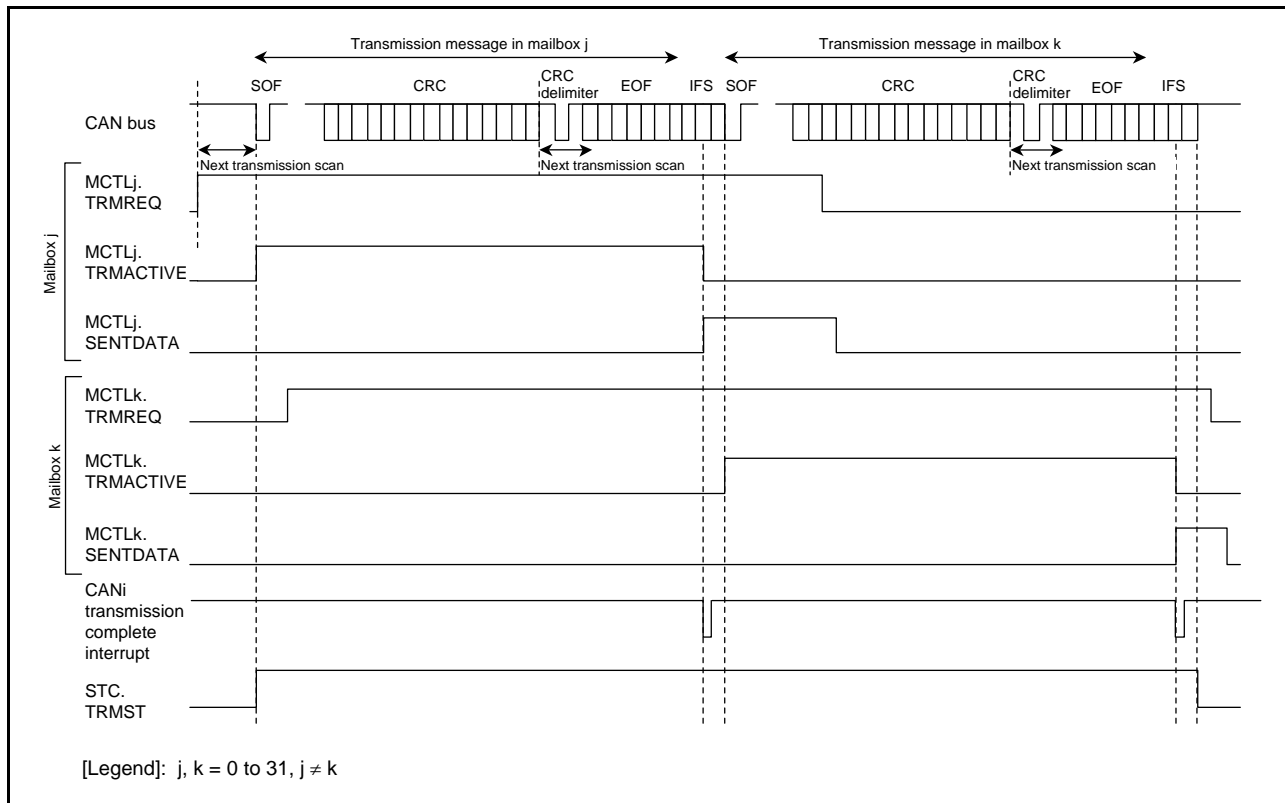


Figure 32.20 Operation Example of Data Frame Transmission

1. When a TRMREQ bit in MCTLj (j = 0 to 31) is set to 1 (transmit mailbox) in the bus-idle state, the mailbox scan processing starts to decide the highest-priority mailbox for transmission. Once the transmit mailbox is decided, the TRMACTIVE bit in MCTLj is set to 1 (from acceptance of transmission request to completion of transmission, or error/arbitration lost), the TRMST bit in STR is set to 1 (transmission in progress), and the CAN module starts transmission.*
2. If other TRMREQ bits are set, the transmission scan processing starts with the CRC delimiter for the next transmission.
3. If transmission is completed without losing arbitration, the SENDDATA bit in MCTLj is set to 1 (transmission completed) and the TRMACTIVE bit is set to 0 (transmission is pending or transmission is not requested). If the interrupt enable bit in MIER is 1 (interrupt enabled), the CAN0 transmission complete interrupt request is generated.
4. When requesting the next transmission from the same mailbox, set bits SENDTDATA and TRMREQ to 0, then set the TRMREQ bit to 1 after checking that bits SENDTDATA and TRMREQ have been set to 0.

Note: * If arbitration is lost after the CAN module starts transmission, the TRMACTIVE bit is set to 0. The transmission scan processing is performed again to search for the highest-priority transmit mailbox from the beginning of the CRC delimiter. If an error occurs either during transmission or following the arbitration lost, the transmission scan processing is performed again to search for the highest-priority transmit mailbox from the start of the error delimiter.

32.8 CAN Interrupt

The CAN module provides the following CAN interrupts for each channel. Table 32.12 lists CAN interrupts.

- CAN0 reception complete interrupt (mailboxes 0 to 31) [RXM0]
- CAN0 transmission complete interrupt (mailboxes 0 to 31) [TXM0]
- CAN0 receive FIFO interrupt [RXF0]
- CAN0 transmit FIFO interrupt [TXF0]
- CAN0 error interrupt [ERS0]

There are eight types of interrupt sources for the CAN0 error interrupts. These sources can be determined by checking EIFR.

- Bus error
- Error-warning
- Error-passive
- Bus-off entry
- Bus-off recovery
- Receive overrun
- Overload frame transmission
- Bus lock

Table 32.12 CAN Interrupts

Module	Interrupt Symbol	Interrupt Source	Source Flag
CAN0	ERS0	Bus lock detected	EIFR.BLIF
		Overload frame transmission detected	EIFR.OLIF
		Overrun detected	EIFR.ORIF
		Bus-off recovery detected	EIFR.BORIF
		Bus-off entry detected	EIFR.BOEIF
		Error-passive detected	EIFR.EPIF
		Error-warning detected	EIFR.EWIF
		Bus error detected	EIFR.BEIF
RXF0	RXF0	Receive FIFO message received (MIER[29] = 0)	—
		Receive FIFO warning (MIER[29] = 1)	—
TXF0	TXF0	Transmit FIFO message transmission completed (MIER[25] = 0)	—
		FIFO last message transmission completed (MIER[25] = 1)	—
RXM0	RXM0	Mailbox 0 to 31 message received	MCTL0.NEWDATA to MCTL31.NEWDATA
TXM0	TXM0	Mailbox 0 to 31 message transmission completed	MCTL0.SENTDATA to MCTL31.SENTDATA

33. Serial Peripheral Interface (RSPI)

33.1 Overview

The RX62N/RX621 Group includes two independent channels of Serial Peripheral Interface (RSPI).

The RSPI channels are capable of full-duplex high-speed serial communications with multiple processors and peripheral devices.

Table 33.1 shows the specifications of the RSPI, and Figure 33.1 shows a block diagram of the RSPI.

Table 33.1 Specifications of RSPI

Item	Description
Number of channels	Two channels
RSPI transfer functions	<ul style="list-style-type: none"> Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (four-wire method) or clock synchronous operation (three-wire method). Transmit-only operation is available. Capable of serial communications in master/slave mode Switching of the polarity of the serial transfer clock Switching of the phase of the serial transfer clock
Data format	<ul style="list-style-type: none"> MSB-first/LSB-first selectable Transfer bit length is selectable as 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits. 128-bit transmit/receive buffers Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits).
Buffer configuration	<ul style="list-style-type: none"> Double buffer configuration for the transmit/receive buffers
Error detection	<ul style="list-style-type: none"> Mode fault error detection Overrun error detection Parity error detection
SSL control function	<ul style="list-style-type: none"> Four SSL signals (SSLn0 to SSLn3) for each channel In single-master mode, SSL0 to SSL3 signals are output. In multi-master mode: <ul style="list-style-type: none"> SSL0 signal for input, and SSL1 to SSL3 signals for either output or high-impedance. In slave mode: <ul style="list-style-type: none"> SSL0 signal for input, and SSL1 to SSL3 signals for high-impedance. Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) Controllable delay from RSPCK stoppage to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) Function for changing SSL polarity
Control in master transfer	<ul style="list-style-type: none"> A transfer of up to eight commands can be executed sequentially in looped execution. For each command, the following can be set: <ul style="list-style-type: none"> SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, LSB/MSB-first, burst, RSPCK delay, SSL negation delay, and next-access delay A transfer can be initiated by writing to the transmit buffer. MOSI signal value specifiable in SSL negation
Interrupt sources	<ul style="list-style-type: none"> Maskable interrupt sources <ul style="list-style-type: none"> RSPI receive interrupt (receive buffer full) RSPI transmit interrupt (transmit buffer empty) RSPI error interrupt (mode fault, overrun, parity error) RSPI idle interrupt (RSPI idle)
Others	<ul style="list-style-type: none"> Function for switching between CMOS output and open-drain output Function for disabling (initializing) the RSPI Loopback mode

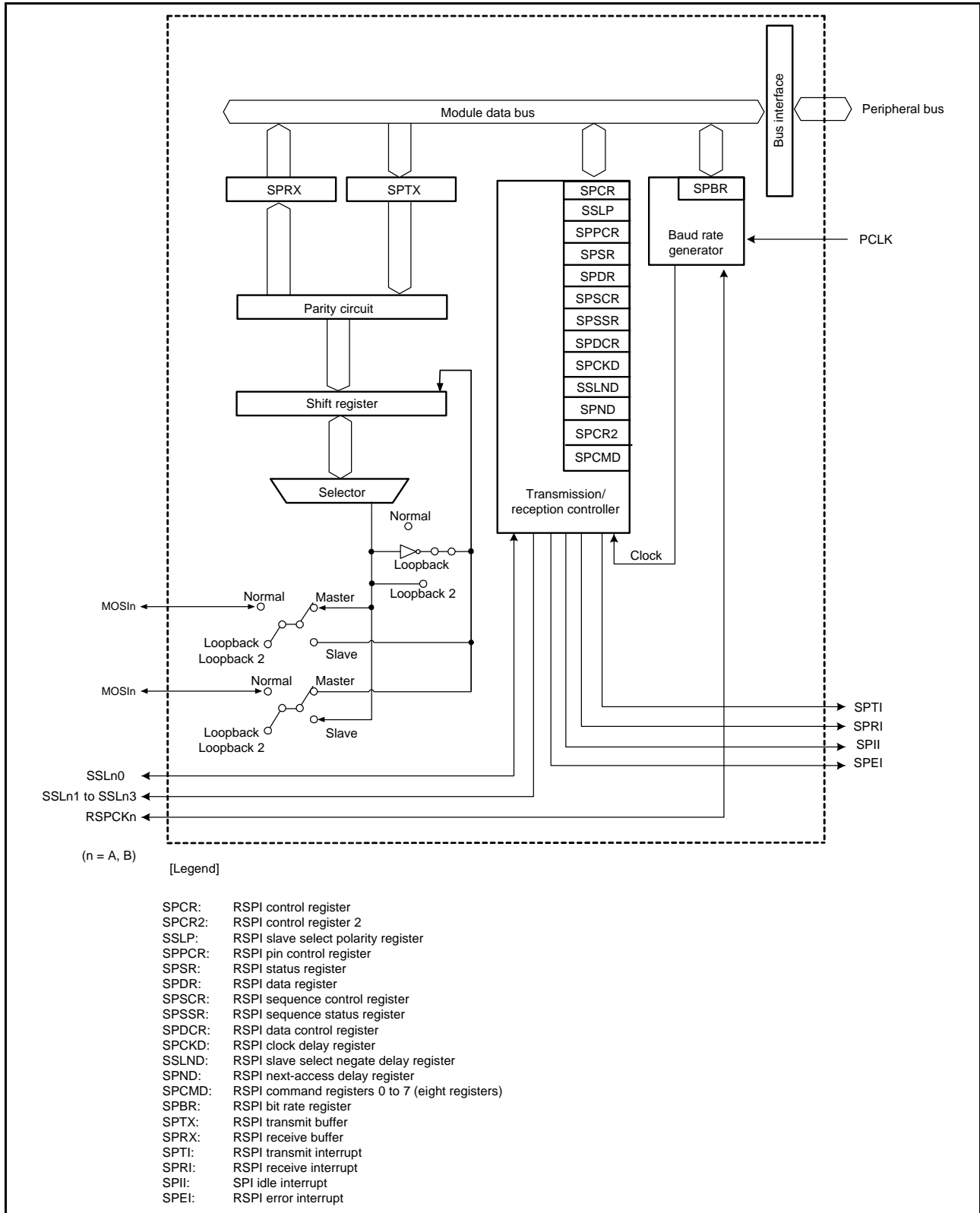


Figure 33.1 Block Diagram of RSPI

Table 33.2 shows the input and output pins used in the RSPI.

The RSPI automatically switches the input/output direction of the SSLn0 (n = A, B) pin. SSL0 is set as an output when the RSPI is a single master and as an input when the RSPI is a multi-master or a slave. Pins RSPCKn, MOSIn, and MISOn (n = A, B) are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLn0 pin (see section 33.3.2, Controlling RSPI Pins).

Table 33.2 RSPI Pin Configuration

Channel	Pin Name	I/O	Function
RSPI0	RSPCKA	I/O	Clock input/output pin
	MOSIA	I/O	Master transmit data input/output pin
	MISOA	I/O	Slave transmit data input/output pin
	SSLA0	I/O	Slave selection input/output pin
	SSLA1	Output	Slave selection output pin
	SSLA2	Output	Slave selection output pin
	SSLA3	Output	Slave selection output pin
RSPI1	RSPCKB	I/O	Clock input/output pin
	MOSIB	I/O	Master transmit data input/output pin
	MISOB	I/O	Slave transmit data input/output pin
	SSLB0	I/O	Slave selection input/output pin
	SSLB1	Output	Slave selection output pin
	SSLB2	Output	Slave selection output pin
	SSLB3	Output	Slave selection output pin

Note: In the description of the pins, the channel number is omitted and pin names are described as RSPCK, MOSI, MISO, and SSL0 to SSL3.

33.2 Register Descriptions

Table 33.3 shows a list of the RSPI registers. These registers enable the RSPI to perform the following controls: specifying master/slave modes, specifying a transfer format, and controlling the transmitter and receiver.

Table 33.3 Registers of RSPI (1 / 2)

Channel	Register Name	Symbol	Value after Reset	Address	Access Size
RSPI0	RSPI control register	SPCR	00h	0008 8380h	8
	RSPI slave select polarity register	SSLP	00h	0008 8381h	8
	RSPI pin control register	SPPCR	00h	0008 8382h	8
	RSPI status register	SPSR	20h	0008 8383h	8
	RSPI data register	SPDR	00000000h	0008 8384h	16, 32
	RSPI sequence control register	SPSCR	00h	0008 8388h	8
	RSPI sequence status register	SPSSR	00h	0008 8389h	8
	RSPI bit rate register	SPBR	FFh	0008 838Ah	8
	RSPI data control register	SPDCR	00h	0008 838Bh	8
	RSPI clock delay register	SPCKD	00h	0008 838Ch	8
	RSPI slave select negation delay register	SSLND	00h	0008 838Dh	8
	RSPI next-access delay register	SPND	00h	0008 838Eh	8
	RSPI control register 2	SPCR2	00h	0008 838Fh	8
	RSPI command register 0	SPCMD0	070Dh	0008 8390h	16
	RSPI command register 1	SPCMD1	070Dh	0008 8392h	16
	RSPI command register 2	SPCMD2	070Dh	0008 8394h	16
	RSPI command register 3	SPCMD3	070Dh	0008 8396h	16
	RSPI command register 4	SPCMD4	070Dh	0008 8398h	16
	RSPI command register 5	SPCMD5	070Dh	0008 839Ah	16
	RSPI command register 6	SPCMD6	070Dh	0008 839Ch	16
RSPI command register 7	SPCMD7	070Dh	0008 839Eh	16	
RSPI1	RSPI control register	SPCR	00h	0008 83A0h	8
	RSPI slave select polarity register	SSLP	00h	0008 83A1h	8
	RSPI pin control register	SPPCR	00h	0008 83A2h	8
	RSPI status register	SPSR	20h	0008 83A3h	8
	RSPI data register	SPDR	00000000h	0008 83A4h	16, 32
	RSPI sequence control register	SPSCR	00h	0008 83A8h	8
	RSPI sequence status register	SPSSR	00h	0008 83A9h	8
	RSPI bit rate register	SPBR	FFh	0008 83AAh	8
	RSPI data control register	SPDCR	00h	0008 83ABh	8
	RSPI clock delay register	SPCKD	00h	0008 83ACh	8
	RSPI slave select negation delay register	SSLND	00h	0008 83ADh	8
	RSPI next-access delay register	SPND	00h	0008 83AEh	8
	RSPI control register 2	SPCR2	00h	0008 83AFh	8

Table 33.3 Registers of RSPI (2 / 2)

Channel	Register Name	Symbol	Value after Reset	Address	Access Size
RSPI1	RSPI command register 0	SPCMD0	070Dh	0008 83B0h	16
	RSPI command register 1	SPCMD1	070Dh	0008 83B2h	16
	RSPI command register 2	SPCMD2	070Dh	0008 83B4h	16
	RSPI command register 3	SPCMD3	070Dh	0008 83B6h	16
	RSPI command register 4	SPCMD4	070Dh	0008 83B8h	16
	RSPI command register 5	SPCMD5	070Dh	0008 83BAh	16
	RSPI command register 6	SPCMD6	070Dh	0008 83BCh	16
	RSPI command register 7	SPCMD7	070Dh	0008 83BEh	16

[Legend] x: Undefined

Note: In the description of the register names, the channel is omitted.

33.2.1 RSPI Control Register (SPCR)

Addresses: RSPI0.SPCR 0008 8380h, RSPI1.SPCR 0008 83A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPMS	RSPI Mode Select	0: SPI operation (four-wire method) 1: Clock synchronous operation (three-wire method)	R/W
b1	TXMD	Communications Operating Mode Select	0: Full-duplex synchronous serial communications 1: Serial communications consisting of only transmit operations	R/W
b2	MODFEN	Mode Fault Error Detection Enable	0: Disables the detection of mode fault error 1: Enables the detection of mode fault error	R/W
b3	MSTR	RSPI Master/Slave Mode Select	0: Slave mode 1: Master mode	R/W
b4	SPEIE	RSPI Error Interrupt Enable	0: Disables the generation of RSPI error interrupt requests 1: Enables the generation of RSPI error interrupt requests	R/W
b5	SPTIE	RSPI Transmit Interrupt Enable	0: Disables the generation of RSPI transmit interrupt requests 1: Enables the generation of RSPI transmit interrupt requests	R/W
b6	SPE	RSPI Function Enable	0: Disables the RSPI function 1: Enables the RSPI function	R/W
b7	SPRIE	RSPI Receive Interrupt Enable	0: Disables the generation of RSPI receive interrupt requests 1: Enables the generation of RSPI receive interrupt requests	R/W

SPCR sets the operating mode of the RSPI. If the SPCR.MSTR, SPCR.MODFEN, and SPCR.TXMD bits are changed while the RSPI function is enabled (SPCR.SPE = 1), subsequent operations cannot be guaranteed.

SPMS Bit (RSPI Mode Select)

The SPMS bit selects SPI operation (four-wire method) or clock synchronous operation (three-wire method).

The SSL0 to SSL3 pins are not used in clock synchronous operation. The three pins RSPCK, MOSI, and MISO handle communications. If clock-synchronous operation is to proceed in master mode (SPCR.MSTR = 1), the CPHA bit in the RSPI command register (SPCMD) can be set to either 0 or 1. Set the CPHA bit to 1 if clock-synchronous operation is to proceed in slave mode (SPCR.MSTR = 0). Operation is not guaranteed if the CPHA bit is set to 0 when clock-synchronous operation is to proceed in slave mode (SPCR.MSTR = 0). (n = A, B)

TXMD Bit (Communications Operating Mode Select)

The TXMD bit selects full-duplex synchronous serial communications or transmit operations only.

When performing communications with the TXMD bit set to 1, the RSPI performs only transmit operations and not receive operations (see section 33.3.6, Communications Operating Mode).

When the TXMD bit is set to 1, receive buffer full interrupt requests cannot be used.

MODFEN Bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault error (see section 33.3.8, Error Detection). In addition, the RSPI determines the input/output direction of the SSL0 to SSL3 pins based on combinations of the MODFEN and MSTR bits (see section 33.3.2, Controlling RSPI Pins).

MSTR Bit (RSPI Master/Slave Mode Select)

The MSTR bit selects master/slave mode of the RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCKn, MOSIn, MISOn, and SSLn. (n = A, B)

SPEIE Bit (RSPI Receive 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 bit to 1, or when the RSPI detects an overrun error and sets the SPSR.OVRF bit to 1 (see section 33.3.8, Error Detection).

SPTIE Bit (RSPI Transmit Interrupt Enable)

The SPTIE bit enables or disables the generation of RSPI transmit interrupt requests when the RSPI detects transmit buffer empty.

At the beginning of transmission, the transmit interrupt requests are generated by setting the SPE bit to 1 at the same time or after the SPTIE bit has been set to 1.

Therefore, note that even while the RSPI function is disabled (SPE bit is 0), setting the SPTIE bit to 1 will generate an RSPI transmit interrupt request.

SPE Bit (RSPI Function Enable)

The SPE bit enables or disables the RSPI function. When the SPSR.MODF bit is 1, the SPE bit cannot be set to 1. For details, refer to section 33.3.8, Error Detection.

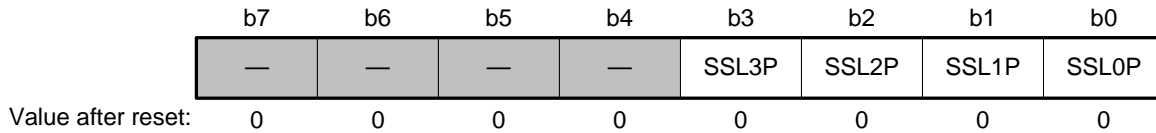
Setting the SPE bit to 0 disables the RSPI function, and initializes a part of the module function. For details, refer to section 33.3.9, Initializing RSPI.

SPRIE Bit (RSPI Receive Interrupt Enable)

If the RSPI has detected a receive buffer write after completion of a serial transfer, the SPRIE bit enables or disables the generation of an RSPI receive interrupt request.

33.2.2 RSPI Slave Select Polarity Register (SSLP)

Addresses: RSPI0.SSLP 0008 8381h, RSPI1.SSLP 0008 83A1h



Bit	Symbol	Bit Name	Description	R/W
b0	SSL0P	SSL0 Signal Polarity Setting	0: SSL0 signal is 0-active 1: SSL0 signal is 1-active	R/W
b1	SSL1P	SSL1 Signal Polarity Setting	0: SSL1 signal is 0-active 1: SSL1 signal is 1-active	R/W
b2	SSL2P	SSL2 Signal Polarity Setting	0: SSL2 signal is 0-active 1: SSL2 signal is 1-active	R/W
b3	SSL3P	SSL3 Signal Polarity Setting	0: SSL3 signal is 0-active 1: SSL3 signal is 1-active	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

SSLP sets the polarity of the SSL0 to SSL3 signals of the RSPI.

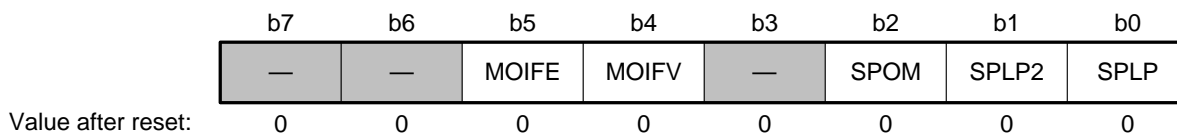
If the contents of SSLP are changed by the CPU while the RSPI function is enabled (SPCR.SPE = 1), subsequent operations are not guaranteed.

SSLiP Bit (SSL Signal Polarity Setting)

These bits set the polarity of the SSLi signals. The setting of SSLiP (i = 3 to 0) indicates the active polarity of the SSLi signal.

33.2.3 RSPI Pin Control Register (SPPCR)

Addresses: RSPI0.SPPCR 0008 8382h, RSPIB.SPPCR 0008 83A2h



Bit	Symbol	Bit Name	Description	R/W
b0	SPLP	RSPI Loopback	0: Normal mode 1: Loopback mode (reversed transmit data = receive data)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (transmit data = receive data)	R/W
b2	SPOM	RSPI Output Pin Mode	0: CMOS output 1: Open-drain output	R/W
b3	—	Reserved	This bit is always read as 0. The write value should always be 0.	R
b4	MOIFV	MOSI Idle Fixed Value	0: MOSI Idle fixed value equals 0 1: MOSI Idle fixed value equals 1	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit	R/W
b7, b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R

SPPCR sets the modes of the RSPI pins. If the contents of SPPCR are changed by the CPU while the RSPI function is enabled (SPCR.SPE = 1), subsequent operations are not guaranteed.

SPLP Bit (RSPI Loopback)

The SPLP bit selects the mode of the RSPI pins.

When the SPLP bit is set to 1, the RSPI shuts off the path between the MISO pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI pin and the shift register if the SPCR.MSTR bit is 0, and connects (reverses) the input path and output path for the shift register (loopback mode).

For details, see section 33.3.15, Loopback Mode.

SPLP2 Bit (RSPI Loopback 2)

The SPLP2 bit selects the mode of the RSPI pins.

When the SPLP2 bit is set to 1, the RSPI shuts off the path between the MISO pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path for the shift register (loopback mode 2). (n = A, B)

For details, see section 33.3.15, Loopback Mode.

SPOM Bit (RSPI Output Pin Mode)

The SPOM bit sets the RSPI output pins to function as CMOS output pins or open drain output pins.

MOIFV Bit (MOSI Idle Fixed Value)

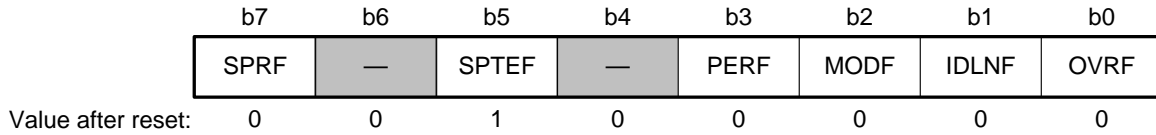
If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSI pin output value during the SSL negation period (including the SSL retention period during a burst transfer). (n = A, B)

MOIFE Bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSI 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. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSI bit.

33.2.4 RSPI Status Register (SPSR)

Addresses: RSPI0.SPSR 0008 8383h, RSPI1.SPSR 0008 83A3h



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 always read as 0. The write value should always be 0.	R/W
b5	SPTEF	Transmit Buffer Empty Flag	0: When data is transferred to SPDR (Transmit buffer is full) 1: When data is transferred from SPDR to shift register (Transmit buffer is empty)	R/(W)*2
b6	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b7	SPRF	Receive Buffer Full Flag	0: When data is transferred from SPDR (SPDR has no valid received data) 1: When data has been received normally and transferred from shift register to SPDR (SPDR has valid received data)	R/(W)*2

Note 1. Only 0 can be written to clear the flag after reading 1.

Note 2. The write value should always be 1.

SPSR indicates the operating status of the RSPI. SPSR can always be read by the CPU. Writing to SPSR can only be performed by the CPU under certain conditions.

OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error.

[Setting condition]

- If a serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer holds data that has not yet been read out

[Clearing condition]

- The CPU reads SPSR when the OVRF bit is 1, and then writes the value 0 to the OVRF bit.

IDLNF Flag (RSPI Idle Flag)

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

<Master mode>

- Even one condition among the following clearing conditions for master mode is not satisfied

<Slave mode>

- The SPCR.SPE bit is 1 (RSPI function is enabled)

[Clearing conditions]

<Master mode>

- 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 000 (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)

The flag is cleared to 0 when the above first clearing condition is satisfied or all of the second to fourth clearing conditions are satisfied.

<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 SSL_n 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 SSL_n 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 SSL signal is determined by the SSLP.SSLiP bit (SSL signal polarity setting bit). (i = 0 to 3)

[Clearing condition]

- The CPU reads SPSR when the MODF bit is 1, and then writes the value 0 to the MODF bit

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]

- The CPU reads SPSR when the PERF bit is 1, and then writes the value 0 to the PERF bit

SPTEF Bit (Transmit Buffer Empty Flag)

Indicates whether the transmit buffer is empty.

[Setting condition]

- When data is transferred from SPDR to shift register

[Clearing condition]

- When data is transferred to SPDR

SPRF Bit (Receive Buffer Full Flag)

Indicates whether the receive buffer is full.

[Setting condition]

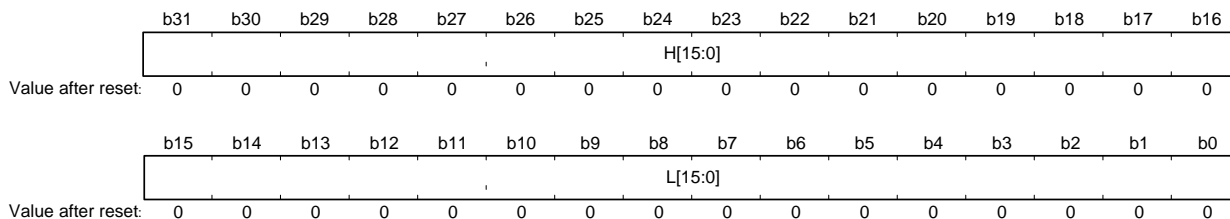
- When data has been received normally and transferred from shift register to SPDR

[Clearing condition]

- When data is transferred from SPDR

33.2.5 RSPI Data Register (SPDR)

Addresses: RSPI0.SPDR 0008 8384h, RSPI1.SPDR 0008 83A4h



SPDR which can always be read from or written to by the CPU is a buffer that holds data for transmission and reception by the RSPI.

The transmit buffer (SPTX) and receive buffer (SPRX) are independent and are mapped to SPDR.

Reading from or writing to SPDR should be performed in word or longword units according to the SPDCR.SPLW bit setting. When the SPLW bit is 0, SPDR is a 64-bit buffer that consists of up to four 16-bit frames. When the SPLW bit is 1, SPDR is a 128-bit buffer that consists of up to four 32-bit frames.

The frame length used by SPDR is determined by the number of frames specification bits (SPDCR.SPFC[1:0]), and the bit length to be used is determined by the RSPI data length specification bits (SPCMD.SPB[3:0]).

If the transmit buffer (SPTX) is empty (data for the next transfer is not set) when data is written to SPDR, the RSPI allows writing of the data to the transmit buffer of SPDR. If the transmit buffer holds data that has not yet been transmitted, the RSPI does not allow updating of the transmit buffer of SPDR.

If the SPDCR.SPRDTD bit is 0 when data is read from SPDR, the RSPI allows reading of the receive buffer. If the SPDCR.SPRDTD bit is 1, the RSPI allows reading of the transmit buffer.

When reading from the transmit buffer, the value written to the buffer immediately before the read operation is read. If the transmit buffer holds data that has not yet been transmitted, the data read out will be 0.

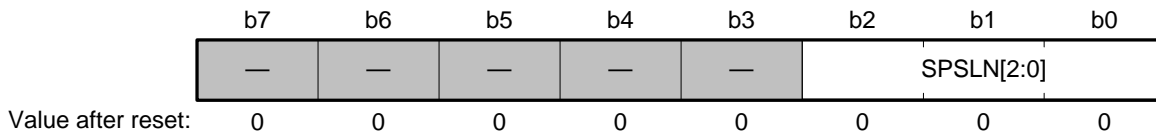
In the normal operation method, data is read from the receive buffer when the SPRDTD bit is 0 and a receive buffer full interrupt occurs. When the receive buffer holds data that has not yet been read out or the SPSR.OVRF flag is 1, the RSPI does not update the receive buffer of SPDR at the end of a serial transfer.

To read from or write to the SPDR register in words or longwords, access the following addresses. Otherwise, the data is not guaranteed.

- Longwords: RSPI0.SPDR 0008 8384h
RSPI1.SPDR 0008 83A4h
- Words: RSPI0.SPDR 0008 8384h
RSPI1.SPDR 0008 83A4h

33.2.6 RSPI Sequence Control Register (SPSCR)

Addresses: RSPI0.SPSCR 0008 8388h, RSPI1.SPSCR 0008 83A8h



Bit	Symbol	Bit Name	Description	R/W																											
b2 to b0	SPSLN[2:0]	RSPI Sequence Length Specification	<table border="0"> <tr> <td>b2 b1 b0</td> <td>Sequence Length</td> <td>Referenced SPCMD0 to SPCMD7 (No.)</td> </tr> <tr> <td>0 0 0:</td> <td>1</td> <td>0 → 0 → ...</td> </tr> <tr> <td>0 0 1:</td> <td>2</td> <td>0 → 1 → 0 → ...</td> </tr> <tr> <td>0 1 0:</td> <td>3</td> <td>0 → 1 → 2 → 0 → ...</td> </tr> <tr> <td>0 1 1:</td> <td>4</td> <td>0 → 1 → 2 → 3 → 0 → ...</td> </tr> <tr> <td>1 0 0:</td> <td>5</td> <td>0 → 1 → 2 → 3 → 4 → 0 → ...</td> </tr> <tr> <td>1 0 1:</td> <td>6</td> <td>0 → 1 → 2 → 3 → 4 → 5 → 0 → ...</td> </tr> <tr> <td>1 1 0:</td> <td>7</td> <td>0 → 1 → 2 → 3 → 4 → 5 → 6 → 0 → ...</td> </tr> <tr> <td>1 1 1:</td> <td>8</td> <td>0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 0 → ...</td> </tr> </table> <p>SPCMD0 to SPCMD7 to be referenced and the order in which they are referenced are changed according to the sequence length that is set in these bits. The relationship among the setting of these bits, sequence length, and SPCMD0 to SPCMD7 referenced by the RSPI is shown below. However, the RSPI in slave mode always 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 always read as 0. The write value should always be 0.	R/W																											

SPSCR sets the sequence control method when the RSPI operates in master mode. When changing the SPSLN[2:0] bits in SPSCR while the SPCR.MSTR and SPCR.SPE bits are 1 with the RSPI function in master mode enabled, the bits should be changed while the SPSR.IDLNF flag is 0.

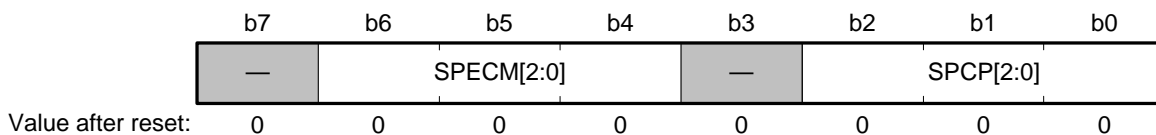
SPSLN[2:0] Bits (RSPI Sequence Length Specification)

The SPSLN[2:0] bits specify a sequence length when the RSPI in master mode performs sequential operations. The RSPI in master mode changes RSPI command registers 0 to 7 (SPCMD0 to SPCMD7) to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN[2:0] bits.

The RSPI in slave mode always references SPCMD0.

33.2.7 RSPI Sequence Status Register (SPSSR)

Addresses: RSPI0.SPSSR 0008 8389h, RSPI1.SPSSR 0008 83A9h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPCP[2:0]	RSPI Command Pointer	b2 b1 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 always read as 0 and cannot be modified.	R/W
b6 to b4	SPECM[2:0]	RSPI Error Command	b6 b5 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 always read as 0 and cannot be modified.	R/W

SPSSR indicates the sequence control status when the RSPI operates in master mode.

Any writing to SPSSR by the CPU is ignored.

SPCP[2:0] Bits (RSPI Command Pointer)

The SPCP[2:0] bits indicate SPCMD0 to SPCMD7 that is currently pointed to by the pointer during sequence control by the RSPI.

For the RSPI's sequence control, see section 33.3.10.1, Master Mode Operation.

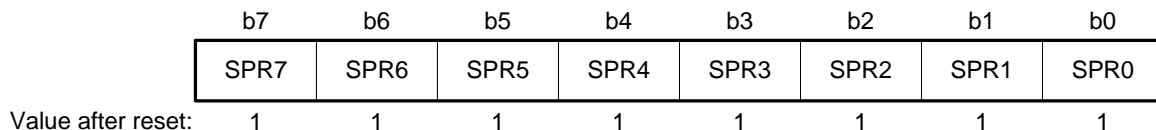
SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate SPCMD0 to SPCMD7 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 all of the SPSR.OVRF, MODF, and PERF 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, see section 33.3.8, Error Detection. For the RSPI's sequence control, see section 33.3.10.1, Master Mode Operation.

33.2.8 RSPI Bit Rate Register (SPBR)

Addresses: RSPI0.SPBR 0008 838Ah, RSPI1.SPBR 0008 83AAh



SPBR which can be read from or written to by the CPU sets the bit rate in master mode. If the contents of SPBR are changed by the CPU while both the SPCR.MSTR and SPCR.SPE bits are 1 with the RSPI function in master mode enabled, subsequent operations cannot be guaranteed.

When the RSPI is used in slave mode, the bit rate depends on the bit rate of the input clock (bit rate satisfying the electrical characteristics should be used) regardless of the settings of SPBR and the SPCMDm.BRDV[1:0] bits (bit rate division setting bits).

The bit rate is determined by combinations of the SPBR setting and the BRDV[1:0] bit setting in SPCMD0 to SPCMD7. 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).

$$\text{Bit rate} = \frac{f(R\phi)}{2 \times (n+1) 2^N}$$

Table 33.4 shows examples of the relationship between the SPBR register and BRDV[1:0] bit settings.

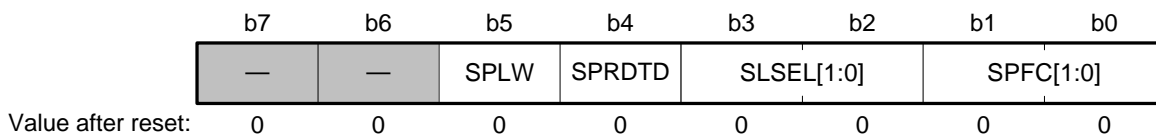
Table 33.4 Relationship between SPBR and BRDV[1:0] Bit Settings

SPBR (n)	BRDV[1:0]		Bit Rate			
	Bits (N)	Division Ratio	PCLK = 32 MHz	PCLK = 36 MHz	PCLK = 40 MHz	PCLK = 50 MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps*	25.0 Mbps*
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps
5	2	48	667 kbps	750 kbps	833 kbps	1.04 Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps

Note: * Can be set in this LSI but bit rates satisfying the electrical characteristics should be used.

33.2.9 RSPI Data Control Register (SPDCR)

Addresses: RSPI0.SPDCR 0008 838Bh, RSPI1.SPDCR 0008 83ABh



Bit	Symbol	Bit Name	Description	R/W																											
b1, b0	SPFC[1:0]	Number of Frames Specification	These bits specify the number of frames that can be stored in SPDR. Table 33.5 and Figure 33.2 show the frame configurations that can be stored in SPDR and examples of combinations of settings for transmission and reception. If combinations of settings other than those shown in the examples are made, subsequent operations cannot be guaranteed.	R/W																											
b3, b2	SLSEL[1:0]	SSI Pin Output Selection	<table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th></th> <th>SLSEL</th> <th>SLSEL</th> <th>SLSEL</th> <th>SLSEL</th> </tr> <tr> <th></th> <th>[1:0] = 00b</th> <th>[1:0] = 01b</th> <th>[1:0] = 10b</th> <th>[1:0] = 11b</th> </tr> </thead> <tbody> <tr> <td>SSL3</td> <td>Output</td> <td>IO</td> <td>IO</td> <td rowspan="4" style="text-align: center; vertical-align: middle;">Setting prohibited</td> </tr> <tr> <td>SSL2</td> <td>Output</td> <td>IO</td> <td>IO</td> </tr> <tr> <td>SSL1</td> <td>Output</td> <td>IO</td> <td>Output</td> </tr> <tr> <td>SSL0</td> <td>Output</td> <td>Output</td> <td>Output</td> </tr> </tbody> </table> <p style="margin-top: 5px; font-size: small;">Operation is not guaranteed if SLSEL[1:0] = 11b is set.</p>		SLSEL	SLSEL	SLSEL	SLSEL		[1:0] = 00b	[1:0] = 01b	[1:0] = 10b	[1:0] = 11b	SSL3	Output	IO	IO	Setting prohibited	SSL2	Output	IO	IO	SSL1	Output	IO	Output	SSL0	Output	Output	Output	R/W
	SLSEL	SLSEL	SLSEL	SLSEL																											
	[1:0] = 00b	[1:0] = 01b	[1:0] = 10b	[1:0] = 11b																											
SSL3	Output	IO	IO	Setting prohibited																											
SSL2	Output	IO	IO																												
SSL1	Output	IO	Output																												
SSL0	Output	Output	Output																												
b4	SPRDTD	RSPI Receive/Transmit Data Selection	0: SPDR values are read from the receive buffer 1: SPDR values are read from the transmit buffer (but only if the transmit buffer is empty)	R/W																											
b5	SPLW	RSPI Longword Access/Word Access Specification	0: SPDR is accessed in words 1: SPDR is accessed in longwords	R/W																											
b7, b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W																											

SPDCR is a register used to set the number of frames that can be stored in SPDR, control the SSLn pin output (n = A, B) and reading from SPDR, and select the width (longword or word) for access to SPDR.

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 RSPI data length specification bits (SPB[3:0]) in the RSPI command register (SPCMD), the RSPI sequence length specification bits (SPSLN[2:0]) in the RSPI sequence control register (SPSCR), and the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR).

When changing the SPFC[1:0] bits in SPDCR while the SPE bit in the RSPI control register (SPCR) is 1 with the RSPI function enabled, 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. 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 RSPI data length specification bits (SPB[3:0]) in the RSPI command register (SPCMD), the RSPI sequence length specification bits (SPSLN[2:0]) in the RSPI sequence control register (SPSCR), and the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR). Also, the SPFC[1:0] bits specify the number of received data at which the RSPI receive buffer full interrupt is requested. Table 33.5 and Figure 33.2 show the frame configurations that can be stored in SPDR and examples of combinations of settings for transmission and reception. If combinations of settings other than those shown in the examples are made, subsequent operations are not guaranteed.

SLSEL[1:0] Bits (SSI Pin Output Selection)

The SLSEL[1:0] bits control the SSLn pin output in master mode. (n = A, B)

SPRDTD Bit (RSPI Receive/Transmit Data Selection)

The SPRDTD bit selects whether the RSPI data register (SPDR) reads values from the receive buffer or from the transmit buffer.

If reading is from the transmit buffer, the value written to SPDR immediately beforehand is read.

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 SPCMD.SPB[3:0] bits (RSPI data length specification bits) to 8 to 16 bits. When 20, 24, or 32 bits is specified, operation is not guaranteed.

Table 33.5 Data Length Settable by SPB[3:0] Bits

Setting	SPB[3:0]	SPSLN[2:0]	SPFC[1:0]	Number of Frames for Transfer	Number of Frames at which Receive Buffer Full Interrupt Occurs or Transmit Buffer Holding Data is Recognized
1-1	N	000	00	1	1
1-2	N	000	01	2	2
1-3	N	000	10	3	3
1-4	N	000	11	4	4
2-1	N, M	001	01	2	2
2-2	N, M	001	11	4	4
3	N, M, O	010	10	3	3
4	N, M, O, P	011	11	4	4
5	N, M, O, P, Q	100	00	5	1
6	N, M, O, P, Q, R	101	00	6	1
7	N, M, O, P, Q, R, S	110	00	7	1
8	N, M, O, P, Q, R, S, T	111	00	8	1

[Legend]

N, M, O, P, Q, R, S, T: Data length that can be specified by the SPB[3:0] bits

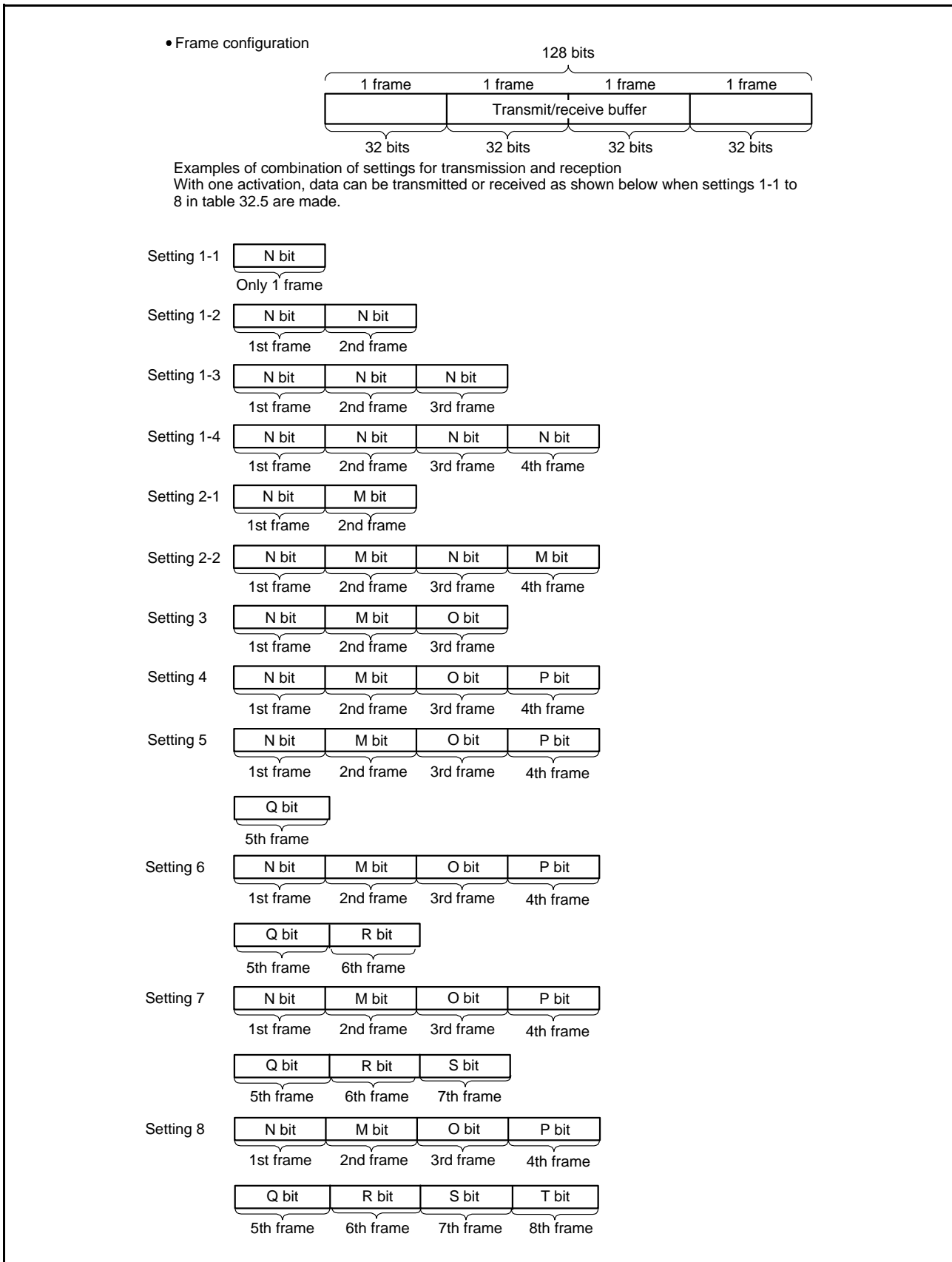
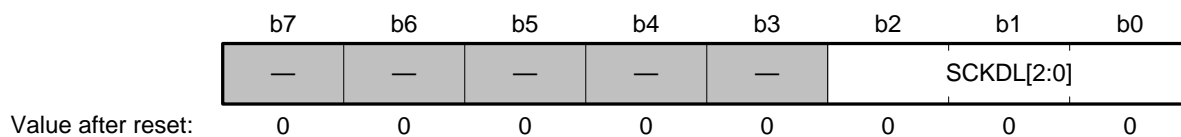


Figure 33.2 Frame Configurations and Examples of Combinations of Transmission and Reception Settings

33.2.10 RSPI Clock Delay Register (SPCKD)

Addresses: RSPI0.SPCKD 0008 838Ch, RSPI1.SPCKD 0008 83ACh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	b2 b1 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

SPCKD sets a period from the beginning of SSL signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMD.SCKDEN bit is 1. If the contents of SPCKD are changed by the CPU while both the SPCR.MSTR and SPCR.SPE bits are 1 with the RSPI function in master mode enabled, subsequent operations cannot be guaranteed. When using the RSPI in slave mode, set the SCKDL[2:0] bits to 000.

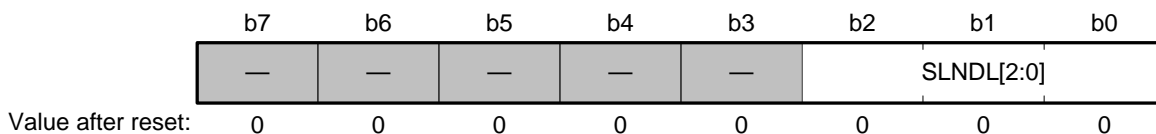
SCKDL[2:0] Bits (RSPCK Delay Setting)

The SCKDL[2:0] bits set an RSPCK delay value when the SPCMD.SCKDEN bit is 1.

When using the RSPI in slave mode, set the SCKDL[2:0] bits to 000.

33.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Addresses: RSPI0.SSLND 0008 838Dh, RSPI1.SSLND 0008 83ADh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SLNDL[2:0]	SSL Negation Delay Setting	b2 b1 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSL signal during a serial transfer by the RSPI in master mode. If the contents of SSLND are changed by the CPU while both the SPCR.MSTR and SPE bits are 1 with the RSPI function in master mode enabled, subsequent operations cannot be guaranteed.

When using the RSPI in slave mode, set the SLNDL[2:0] bits to 000.

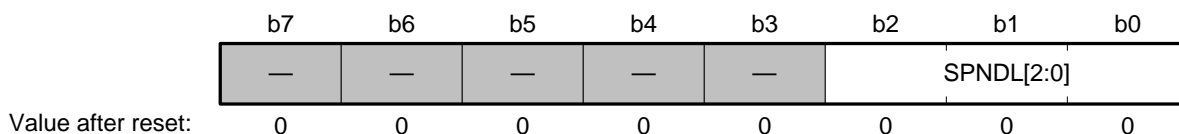
SLNDL[2:0] Bits (SSL Negation Delay Setting)

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 000.

33.2.12 RSPI Next-Access Delay Register (SPND)

Addresses: RSPI0.SPND 0008 838Eh, RSPI1.SPND 0008 83AEh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPNDL[2:0]	RSPI Next-Access Delay Setting	b2 b1 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 always read as 0. The write value should always be 0.	R/W

SPND sets a non-active period (next-access delay) after termination of a serial transfer when the SPCMD.SPNDEN bit is 1. If the contents of SPND are changed by the CPU while both the SPCR.MSTR and SPE bits are 1 with the RSPI function in master mode enabled, subsequent operations cannot be guaranteed.

When using the RSPI in slave mode, set the SPNDL[2:0] bits to 000.

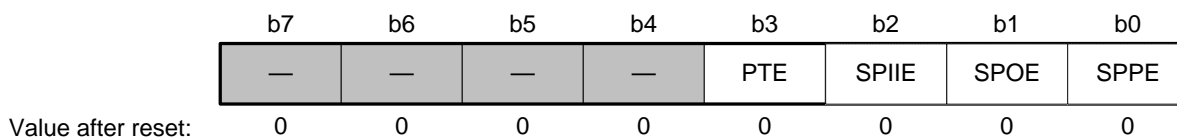
SPNDL[2:0] Bits (RSPI Next-Access Delay Setting)

The SPNDL[2:0] bits set a next-access delay when the SPCMD.SPNDEN bit is 1.

When using the RSPI in slave mode, set the SPNDL[2:0] bits to 000.

33.2.13 RSPI Control Register 2 (SPCR2)

Addresses: RSPI0.SPCR2 0008 838Fh, RSPI1.SPCR2 0008 83AFh



Bit	Symbol	Bit Name	Description	R/W
b0	SPPE	Parity Enable	0: Does not add the parity bit to transmit data and does not check the parity bit of receive data 1: Adds the parity bit to transmit data and checks the parity bit of receive data (when SPCR.TXMD = 0) Adds the parity bit to transmit data but does not check the parity bit of receive data (when SPCR.TXMD = 1)	R/W
b1	SPOE	Parity Mode	0: Selects even parity for use in transmission and reception 1: Selects odd parity for use in transmission and reception	R/W
b2	SPIIE	RSPI Idle Interrupt Enable	0: Disables the generation of idle interrupt requests 1: Enables the generation of idle interrupt requests	R/W
b3	PTE	Parity Self-Testing	0: Disables the self-diagnosis function of the parity circuit 1: Enables the self-diagnosis function of the parity circuit	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

SPCR2 sets the operating mode of the RSPI. If the SPPE or SPOE bit in SPCR2 is changed while the SPCR.SPE bit is 1 with the RSPI function enabled, subsequent operations cannot be guaranteed.

SPPE Bit (Parity Enable)

The SPPE bit enables or disables the parity function.

The parity bit is added to transmit data and parity checking is performed for receive data when the communications operating mode select bit (TXMD) in the RSPI control register (SPCR) is 0 and the SPPE bit in SPCR2 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 SPPE bit in SPCR2 is 1.

SPOE Bit (Parity Mode)

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 IDLNF flag in the RSPI status register (SPSR) is cleared to 0.

PTE Bit (Parity Self-Testing)

The PTE bit enables the self-diagnosis function of the parity circuit in order to check whether the parity function is operating correctly.

33.2.14 RSPI Command Registers 0 to 7 (SPCMD0 to SPCMD7)

Addresses: RSPI0.SPCMD0 0008 8390h, RSPI0.SPCMD1 0008 8392h, RSPI0.SPCMD2 0008 8394h, RSPI0.SPCMD3 0008 8396h, RSPI0.SPCMD4 0008 8398h, RSPI0.SPCMD5 0008 839Ah, RSPI0.SPCMD6 0008 839Ch, RSPI0.SPCMD7 0008 839Eh, RSPI1.SPCMD0 0008 83B0h, RSPI1.SPCMD1 0008 83B2h, RSPI1.SPCMD2 0008 83B4h, RSPI1.SPCMD3 0008 83B6h, RSPI1.SPCMD4 0008 83B8h, RSPI1.SPCMD5 0008 83BAh, RSPI1.SPCMD6 0008 83BCh, RSPI1.SPCMD7 0008 83BEh

b15	b14	b13	b12	b11	b10	b9	b8
SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
Value after reset:	0	0	0	0	1	1	1
b7	b6	b5	b4	b3	b2	b1	b0
SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
Value after reset:	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 = 0 when idle 1: RSPCK = 1 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 b5 b4 0 0 0: SSL0 0 0 1: SSL1 0 1 0: SSL2 0 1 1: SSL3 1 x x: — (Setting prohibited) [Legend] 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 b10 b9 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 Setting Enable	0: A next-access delay of 1 RSPCK + 2 PCLK 1: A next-access delay is equal to the setting of the RSPI next-access delay register (SPND)	R/W
b14	SLNDEN	SSL Negation Delay Setting Enable	0: An SSL negation delay of 1 RSPCK 1: An SSL negation delay is equal to the setting of the RSPI slave select negation delay register (SSLND)	R/W

Bit	Symbol	Bit Name	Description	R/W
b15	SCKDEN	RSPCK Delay Setting Enable	0:An RSPCK delay of 1 RSPCK 1:An RSPCK delay is equal to the setting of the RSPI clock delay register (SPCKD)	R/W

SPCMD0 to SPCMD7 are 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 are used to set a transfer mode for the RSPI in slave mode. The RSPI in master mode sequentially references SPCMD0 to SPCMD7 according to the settings in the SPSLN[2:0] bits in the RSPI sequence control register (SPSCR), and executes the serial transfer that is set in the referenced SPCMD.

SPCMD 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 SPCMD is referenced.

SPCMD that is referenced by the RSPI in master mode can be checked by means of the SPCP[2:0] bits in the RSPI sequence status register (SPSSR). When the RSPI function in slave mode is enabled, subsequent operations cannot be guaranteed if the contents of SPCMD are changed by the CPU.

CPHA Bit (RSPCK Phase Setting)

The CPHA bit sets an RSPCK phase of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK phase setting between the modules.

CPOL Bit (RSPCK Polarity Setting)

The CPOL bit sets an RSPCK polarity of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK polarity setting between the modules.

BRDV[1:0] Bits (Bit Rate Division Setting)

The BRDV[1:0] bits are used to determine the bit rate. A bit rate is determined by combinations of the settings in the BRDV[1:0] bits and SPBR (see section 33.2.8, 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 SPCMD0 to SPCMD7, different BRDV[1:0] bit settings can be specified. This permits the execution of serial transfers at a different bit rate for each command.

SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSL signal assertion when the RSPI performs serial transfers in master mode. Setting the SSLA[2:0] bits controls the assertion for the SSL0 to SSL3 signals. When an SSL signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000 in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLn0 pin acts as input).

When using the RSPI in slave mode, set the SSLA[2:0] bits to 000.

SSLKP Bit (SSL Signal Level Keeping)

When the RSPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSL signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command.

When using the RSPI in slave mode, the SSLKP bit should be set to 0.

SPB[3:0] Bits (RSPI Data Length Setting)

The SPB[3:0] bits set a transfer data length for the RSPI in master mode or slave mode.

LSBF Bit (RSPI LSB First)

The LSBF bit sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first.

SPNDEN Bit (RSPI Next-Access Delay Setting Enable)

The SPNDEN bit sets the period from the time the RSPI in master mode terminates a serial transfer and sets the SSL signal inactive until the RSPI enables the SSL signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPI sets the next-access delay to 1 RSPCK. 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 (SSL negation delay) from the time the master mode RSPI stops RSPCK oscillation until the RSPI sets the SSL signal inactive. If the SLNDEN bit is 0, the RSPI sets the SSL negation delay to 1 RSPCK + 2 PCLK. 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 SSL signal until the RSPCK starts oscillation (RSPI clock delay). If the SCKDEN bit is 0, the RSPI sets the RSPCK delay to 1 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.

33.3 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.

33.3.1 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 33.6 gives the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 33.6 Relationship between RSPI Modes and SPCR and Description of Each Mode (1 / 2)

Mode	Slave (SPI Operation)	Single-Master (SPI Operation)	Multi-Master (SPI Operation)	Slave (Clock Synchronous Operation)	Master (Clock Synchronous Operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
RSPCK signal	Input	Output	Output/Hi-Z	Input	Output
MOSI signal	Input	Output	Output/Hi-Z	Input	Output
MISO signal	Output/Hi-Z	Input	Input	Output	Input
SSL0 signal	Input	Output	Input	Hi-Z	Hi-Z
SSL1 to SSL3 signals	Hi-Z	Output/Hi-Z	Output/Hi-Z	Hi-Z	Hi-Z
Output pin mode	CMOS/ open-drain	CMOS/ open-drain	CMOS/ open-drain	CMOS/ open-drain	CMOS/ open-drain
SSL polarity modification function	Supported	Supported	Supported	—	—
Transfer rate	Up to PCLK/8	Up to PCLK/2	Up to PCLK/2	Up to PCLK/8	Up to PCLK/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two	Two	Two	Two	Two
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
Transfer data length	8 to 32 bits	8 to 32 bits	8 to 32 bits	8 to 32 bits	8 to 32 bits
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0, 1)	Possible (CPHA = 0, 1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer is written to at generation of a transmit buffer empty interrupt request	Transmit buffer is written to at generation of a transmit buffer empty interrupt request	RSPCK oscillation	Transmit buffer is written to at generation of a transmit buffer empty interrupt request
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported	Supported	Supported	Supported	Supported

Table 33.6 Relationship between RSPI Modes and SPCR and Description of Each Mode (2 / 2)

Mode	Slave (SPI Operation)	Single-Master (SPI Operation)	Multi-Master (SPI Operation)	Slave (Clock Synchronous Operation)	Master (Clock Synchronous Operation)
Receive buffer full detection	Supported*1	Supported*1	Supported*1	Supported*1	Supported*1
Overrun error detection	Supported*1	Supported*1	Supported*1	Supported*1	Supported*1
Parity error detection	Supported*1*2	Supported*1*2	Supported*1*2	Supported*1*2	Supported*1*2
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported

Note 1. When the SPCR.TXMD bit is 1, receiver buffer full detection, overrun error detection, and parity error detection are not performed.

Note 2. When the SPCR2.SPPE bit is 0, parity error detection is not performed.

33.3.2 Controlling RSPI Pins

According to the MSTR, MODFEN, and SPMS bits in SPCR and the SPOM bit in SPPCR, the RSPI can automatically switch pin directions and output modes. Table 33.7 shows the relationship between pin states and bit settings.

Table 33.7 Relationship between Pin States and Bit Settings

Mode	Pin	Pin State*1	
		SPPCR.SPOM = 0	SPPCR.SPOM = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCK	CMOS output	Open-drain output
	SSL0 to SSL3	CMOS output	Open-drain output
	MOSI	CMOS output	Open-drain output
	MISO	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCK*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSL0	Input	Input
	SSL1 to SSL3*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSI*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	MISO	Input	Input
	RSPCK	Input	Input
	SSL0	Input	Input
	SSL1 to SSL3	Hi-Z	Hi-Z
Master mode (Clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	MOSI	Input	Input
	MISO*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	RSPCK	CMOS output	Open-drain output
	SSL0 to SSL3*4	Hi-Z	Hi-Z
Slave mode (Clock synchronous operation) (MSTR = 0, SPMS = 1)	MOSI	CMOS output	Open-drain output
	MISO	Input	Input
	RSPCK	Input	Input
	SSL0 to SSL3*4	Hi-Z	Hi-Z
	MOSI	Input	Input
	MISO	CMOS output	Open-drain output

Note 1. RSPI settings are not indicated in the multiplex pins for which the RSPI function is not selected.

Note 2. When SSL0 is at the active level, the pin state is high-impedance.

Note 3. When SSL0 is at the non-active level or the SPE bit in SPCR is cleared (= 0), the pin state is high-impedance.

Note 4. In clock synchronous operation, SSL0 to SSL3 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 shown in Table 33.8.

Table 33.8 MOSI Signal Value Determination during SSL Negation Period

MOIFE Bit	MOIFV Bit	MOSI Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Always 0
1	1	Always 1

33.3.3 RSPI System Configuration Examples

33.3.3.1 Single Master/Single Slave (with This LSI Acting as Master)

Figure 33.3 shows a single-master/single-slave RSPI system configuration example when this LSI is used as a master. In the single-master/single-slave configuration, the SSL0 to SSL3 output of this LSI (master) are not used. The SSL input of the RSPI slave is fixed to the low level, and the RSPI slave is always maintained in a select state.*

This LSI (master) always drives the RSPCK and MOSI. The RSPI slave always drives the MISO.

Note: * In the transfer format corresponding to the case where the SPCMDm.CPHA (m = 0 to 7) 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 SSL output of this LSI should be connected to the SSL input of the slave device.

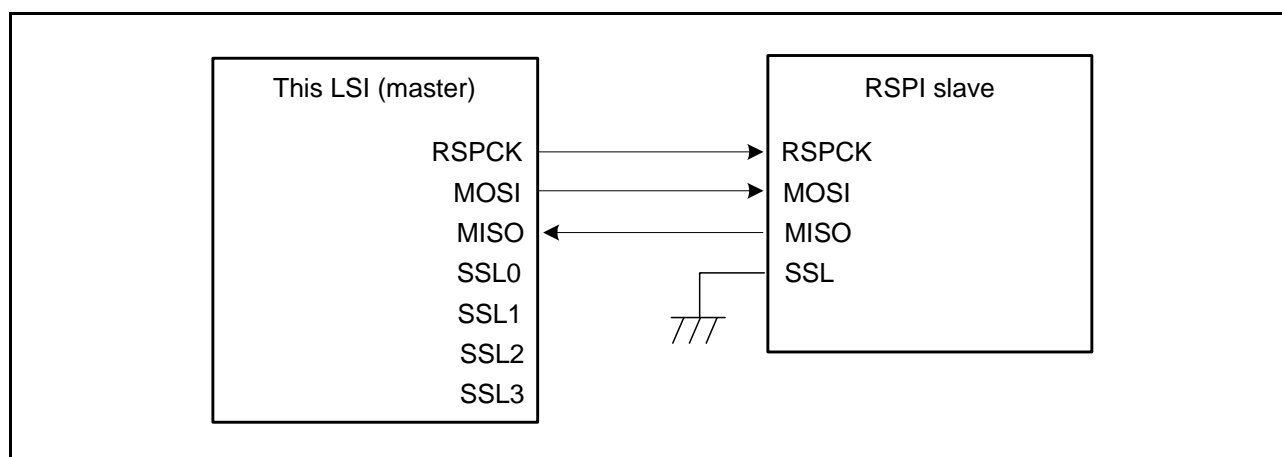


Figure 33.3 Single-Master/Single-Slave Configuration Example (This LSI = Master)

33.3.3.2 Single Master/Single Slave (with This LSI Acting as Slave)

Figure 33.4 shows a single-master/single-slave RSPI system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave, the SSL0 pin is used as SSL input. The RSPI master always drives the RSPCK and MOSI. This LSI (slave) always drives the MISO.*

In the single-slave configuration in which the SPCMD.CPHA bit is set to 1, the SSL0 input of this LSI (slave) is fixed to low, this LSI (slave) is always maintained in a selected state, and in this manner it is possible to execute serial transfer (Figure 33.5).

Note: * When SSL0 is at the non-active level, the pin state is high-impedance.

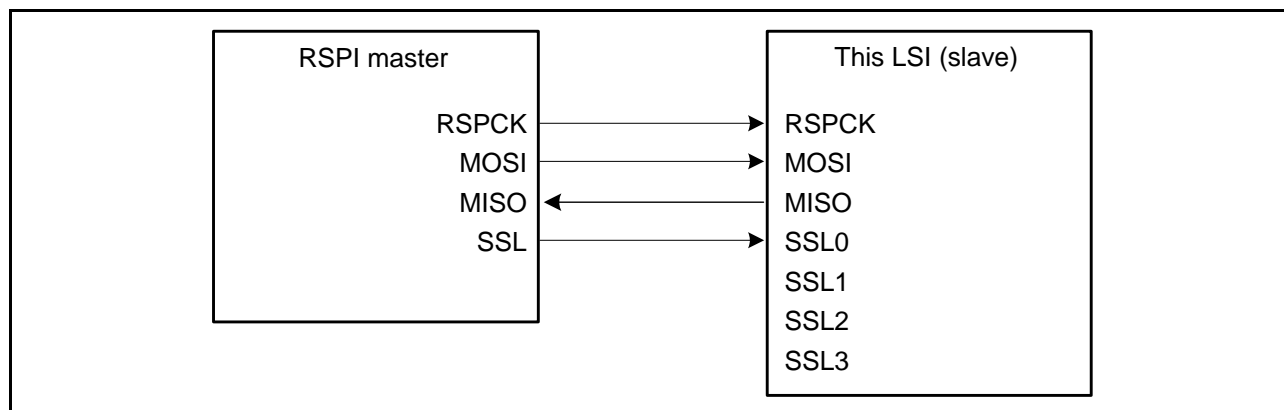


Figure 33.4 Single-Master/Single-Slave Configuration Example (This LSI = Slave)

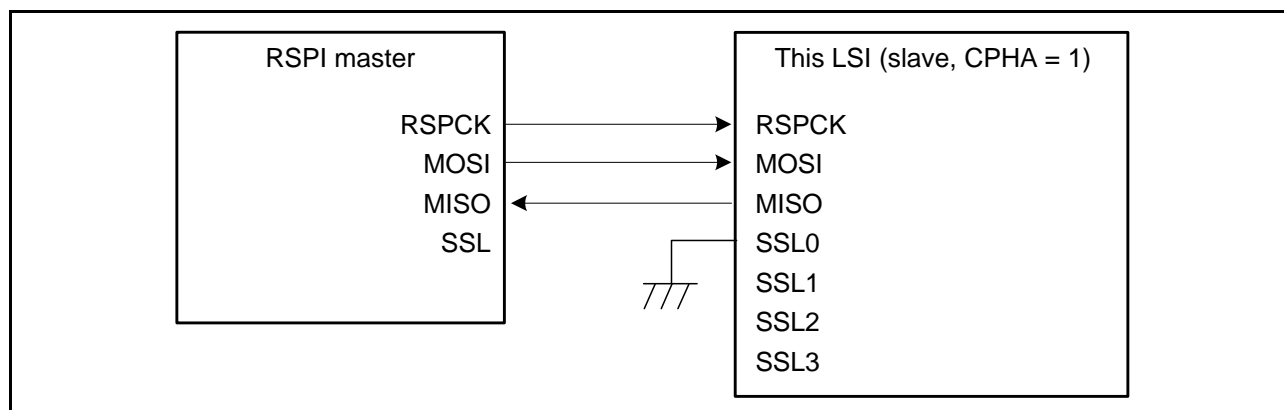


Figure 33.5 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 1)

33.3.3.3 Single Master/Multi-Slave (with This LSI Acting as Master)

Figure 33.6 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of Figure 33.6, the RSPI system is comprised of this LSI (master) and four slaves (RSPI slave 0 to RSPI slave 3).

The RSPCK and MOSI outputs of this LSI (master) are connected to the RSPCK and MOSI inputs of RSPI slave 0 to RSPI slave 3. The MISO outputs of RSPI slave 0 to RSPI slave 3 are all connected to the MISO input of this LSI (master). SSL0 to SSL3 outputs of this LSI (master) are connected to the SSL inputs of RSPI slave 0 to RSPI slave 3, respectively.

This LSI (master) always drives RSPCK, MOSI, and SSL0 to SSL3. Of the RSPI slave 0 to RSPI slave 3, the slave that receives low-input into the SSL input drives MISO.

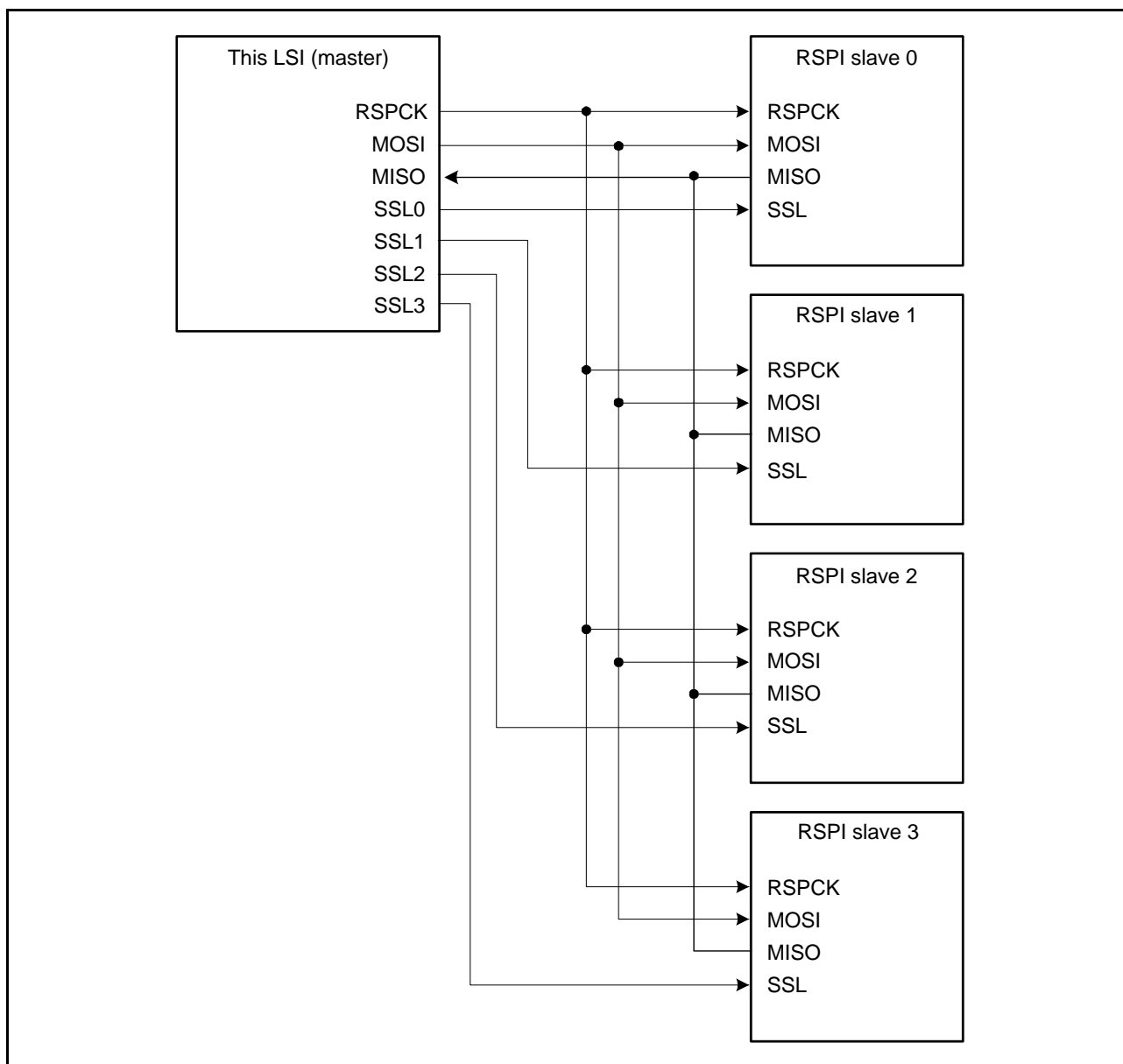


Figure 33.6 Single-Master/Multi-Slave Configuration Example (This LSI = Master)

33.3.3.4 Single Master/Multi-Slave (with This LSI Acting as Slave)

Figure 33.7 shows a single-master/multi-slave RSPi system configuration example when this LSI is used as a slave. In the example of Figure 33.7, the RSPi system is comprised of an RSPi master and two LSIs (slave X and slave Y).

The RSPCK and MOSI outputs of the RSPi master are connected to the RSPCK and MOSI inputs of the LSIs (slave X and slave Y). The MISO outputs of the LSIs (slave X and slave Y) are all connected to the MISO input of the RSPi master. SSLX and SSLY outputs of the RSPi master are connected to the SSL0 inputs of the LSIs (slave X and slave Y), respectively.

The RSPi master always drives RSPCK, MOSI, SSLX, and SSLY. Of the LSIs (slave X and slave Y), the slave that receives low-input into the SSL0 input drives MISO.

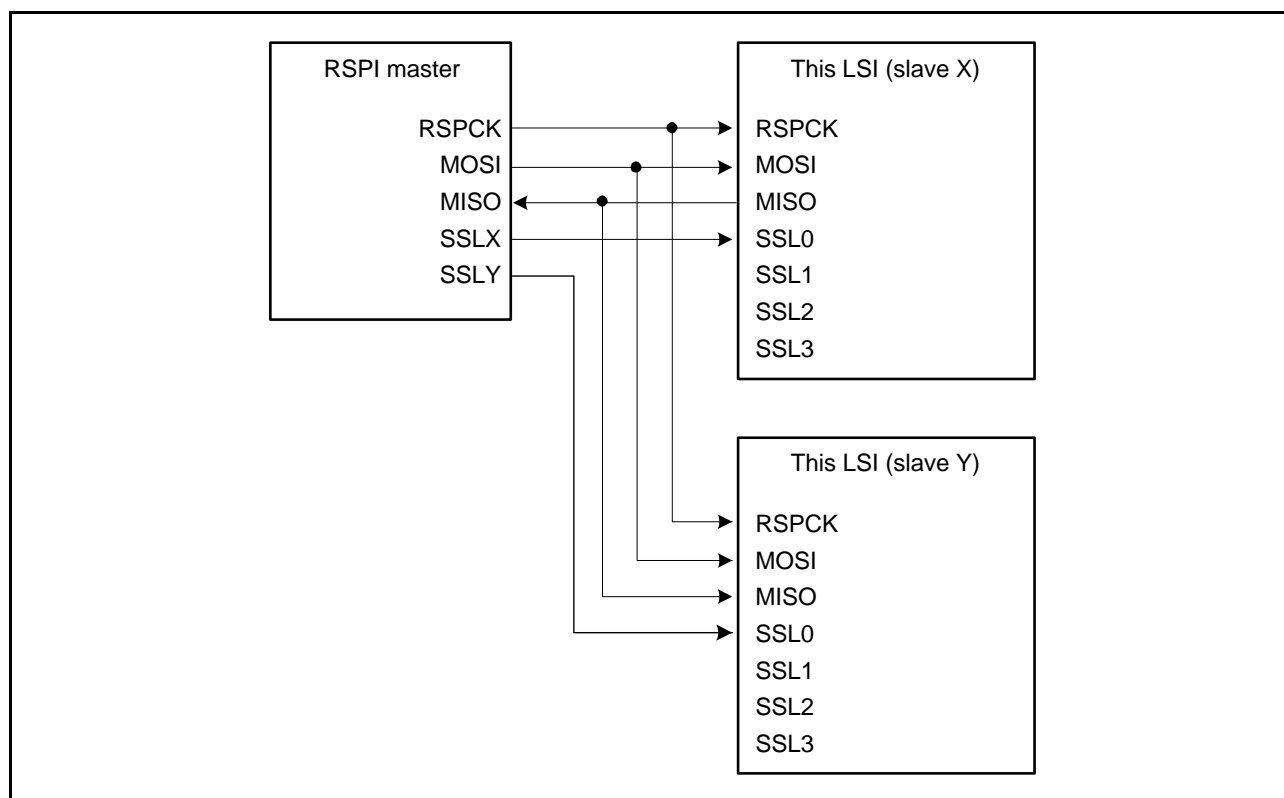


Figure 33.7 Single-Master/Multi-Slave Configuration Example (This LSI = Slave)

33.3.3.5 Multi-Master/Multi-Slave (with This LSI Acting as Master)

Figure 33.8 shows a multi-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of Figure 33.8, the RSPI system is comprised of two LSIs (master X and master Y) and two RSPI slaves (RSPI slave 1 and RSPI slave 2).

The RSPCK and MOSI outputs of the LSIs (master X and master Y) are connected to the RSPCK and MOSI inputs of RSPI slaves 1 and 2. The MISO outputs of RSPI slaves 1 and 2 are connected to the MISO inputs of the LSIs (master X and master Y). Any generic port Y output from this LSI (master X) is connected to the SSL0 input of this LSI (master Y). Any generic port X output of this LSI (master Y) is connected to the SSL0 input of this LSI (master X). The SSL1 and SSL2 outputs of the LSIs (master X and master Y) are connected to the SSL inputs of the RSPI slaves 1 and 2. In this configuration example, because the system can be comprised solely of SSL0 input, and SSL1 and SSL2 outputs for slave connections, the SSL3 output of this LSI is not required.

This LSI drives RSPCK, MOSI, SSL1, and SSL2 when the SSL0 input level is 1. When the SSL0 input level is 0, this LSI detects a mode fault error, sets RSPCK, MOSI, SSL1, and SSL2 to high-impedance, and releases the RSPI bus right to the other master. Of the RSPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

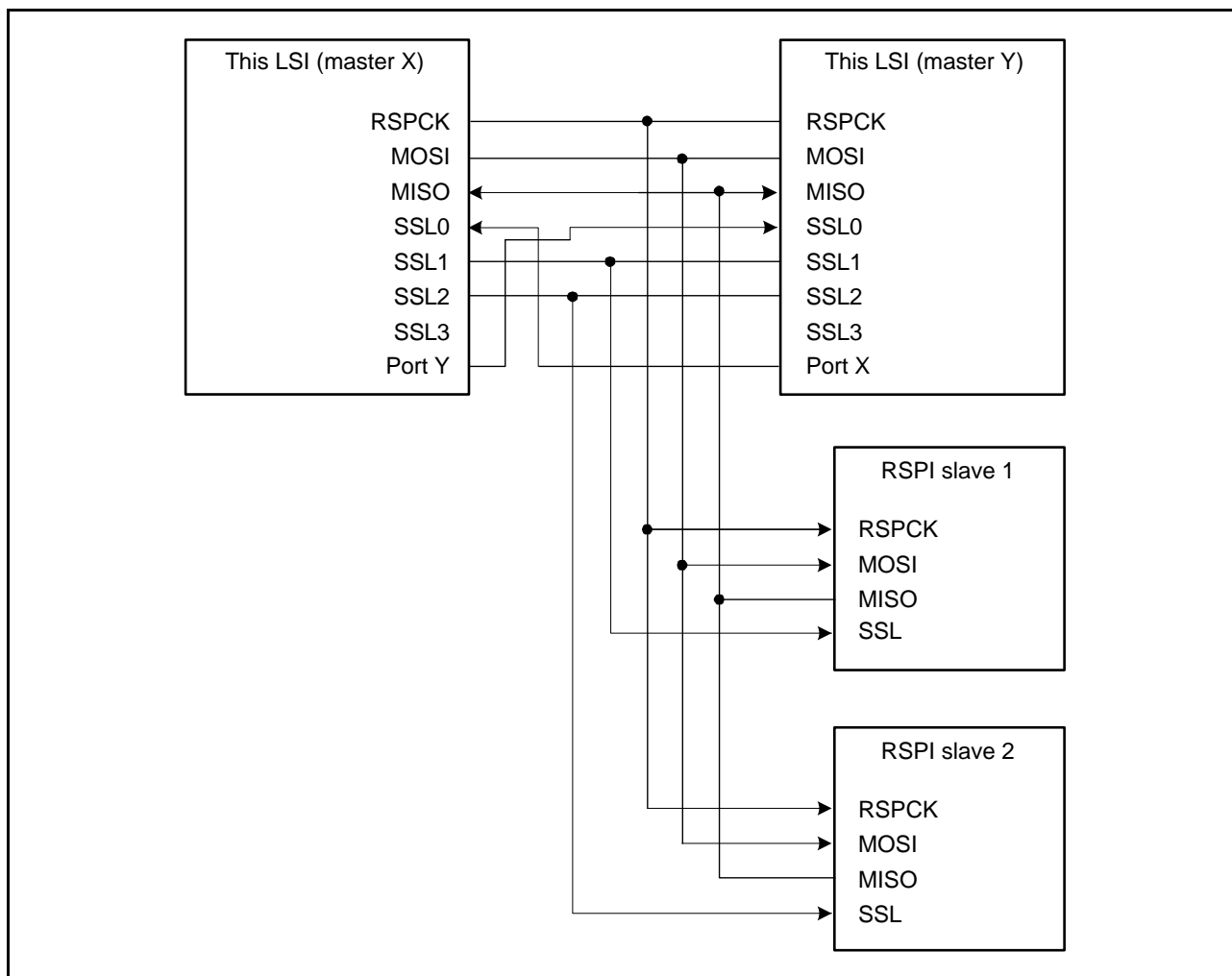


Figure 33.8 Multi-Master/Multi-Slave Configuration Example (This LSI = Master)

33.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Master)

Figure 33.9 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this LSI is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSL0 to SSL3 of this LSI (master) are not used.

This LSI (master) always drives the RSPCK and MOSI. The RSPI slave always drives the MISO.

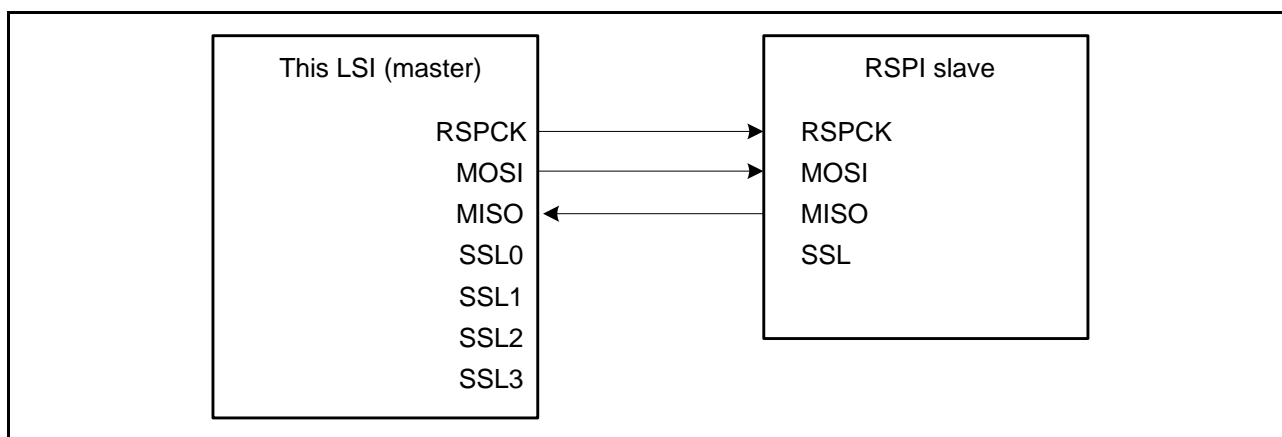


Figure 33.9 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This LSI = Master)

33.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Slave)

Figure 33.10 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave (clock synchronous operation), this LSI (slave) always drives the MISO and the RSPI master always drives the RSPCK and MOSI. In addition, SSL0 to SSL3 of this LSI (slave) are not used.

Only in the single-slave configuration in which the SPCMD.CPHA bit is set to 1, this LSI (slave) can execute serial transfer.

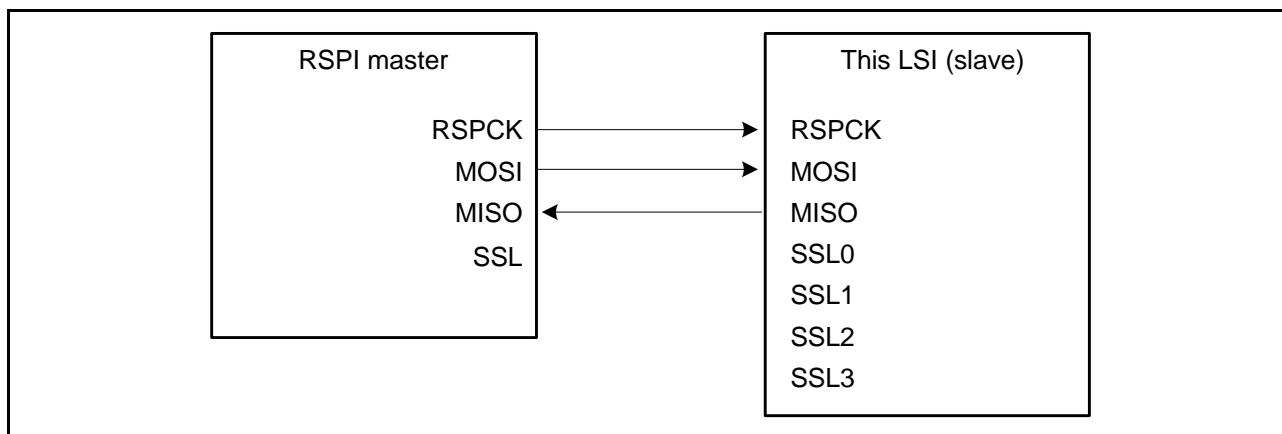


Figure 33.10 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This LSI = Slave, CPHA = 1)

33.3.4 Transfer Format

33.3.4.1 CPHA = 0

Figure 33.11 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA (m = 0 to 7) bit is 0. Note that clock synchronous operation (the SPCR.SPMS bit is 1) is not guaranteed when the RSPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 33.11, RSPCK (CPOL = 0) indicates the RSPCK signal waveform when the CPOL bit in SPCMD is 0; RSPCK (CPOL = 1) indicates the RSPCK signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The input/output directions of the signals depend on the RSPI settings. For details, see section 33.3.2, Controlling RSPI Pins

When the CPHA bit is 0, the driving of valid data to the MOSIn and MISO_n signals commences at an SSL_n signal assertion timing. The first RSPCK signal change timing that occurs after the SSL_n signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIn and MISO_n signals is always 1/2 RSPCK cycles after the transfer data fetch timing. The SPCMD.CPOL bit setting does not affect the RSPCK signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSL signal assertion to RSPCK oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCK oscillation to an SSL signal negation (SSL negation delay). t3 denotes a period in which SSL signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this LSI is in master mode, see section 33.3.10.1, Master Mode Operation. (n = A, B)

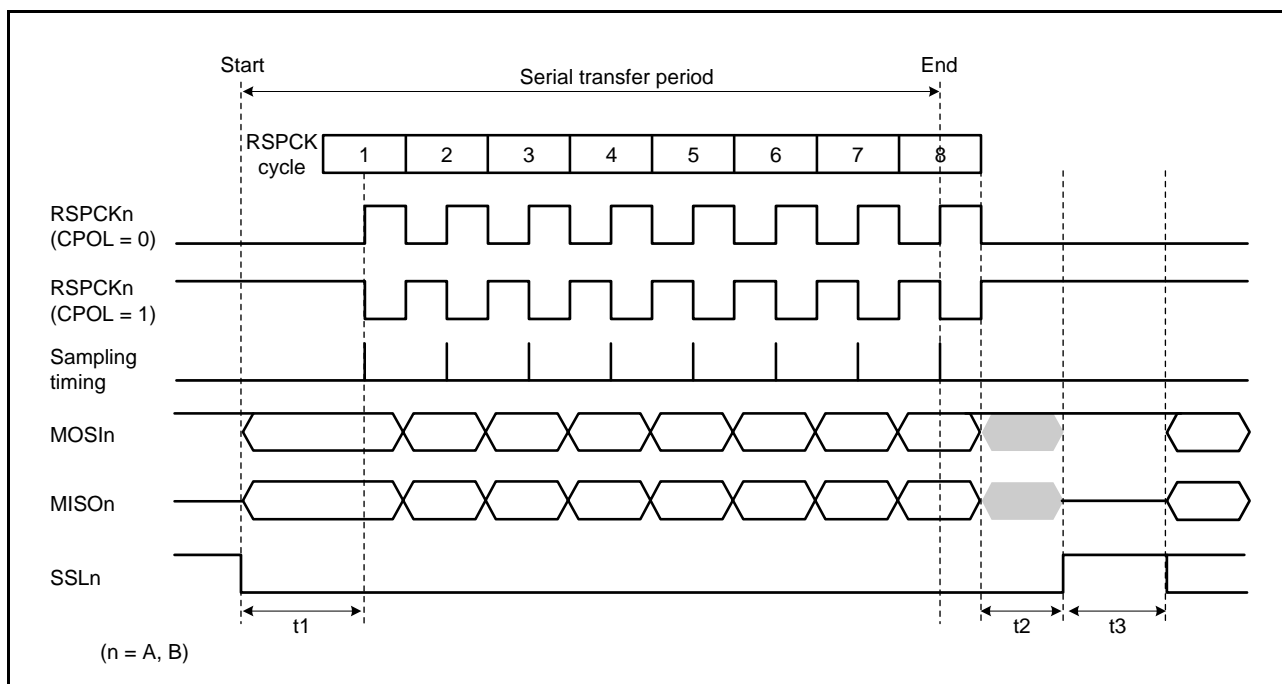


Figure 33.11 RSPI Transfer Format (CPHA = 0)

33.3.4.2 CPHA = 1

Figure 33.12 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA (m = 0 to 7) bit is 1. However, when the SPCR.SPMS bit is 1, the SSL signals are not used, and only the three signals RSPCK, MOSI, and MISO handle communications. In Figure 33.12, RSPCKn (CPOL = 0) indicates the RSPCK signal waveform when the CPOL bit in SPCMD is 0; RSPCKn (CPOL = 1) indicates the RSPCK signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The input/output directions of the signals depend on the RSPI mode (master or slave). For details, see section 33.3.2, Controlling RSPI Pins .

When the CPHA bit is 1, the driving of invalid data to the MISO signal commences at an SSL signal assertion timing. The output of valid data to the MOSI and MISO signals commences at the first RSPCK signal change timing that occurs after the SSL signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is always 1/2 RSPCK cycles after the data update timing. The SPCMD.CPOL bit setting does not affect the RSPCK signal operation timing; it only affects the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this LSI is in master mode, see section 33.3.10.1, Master Mode Operation.

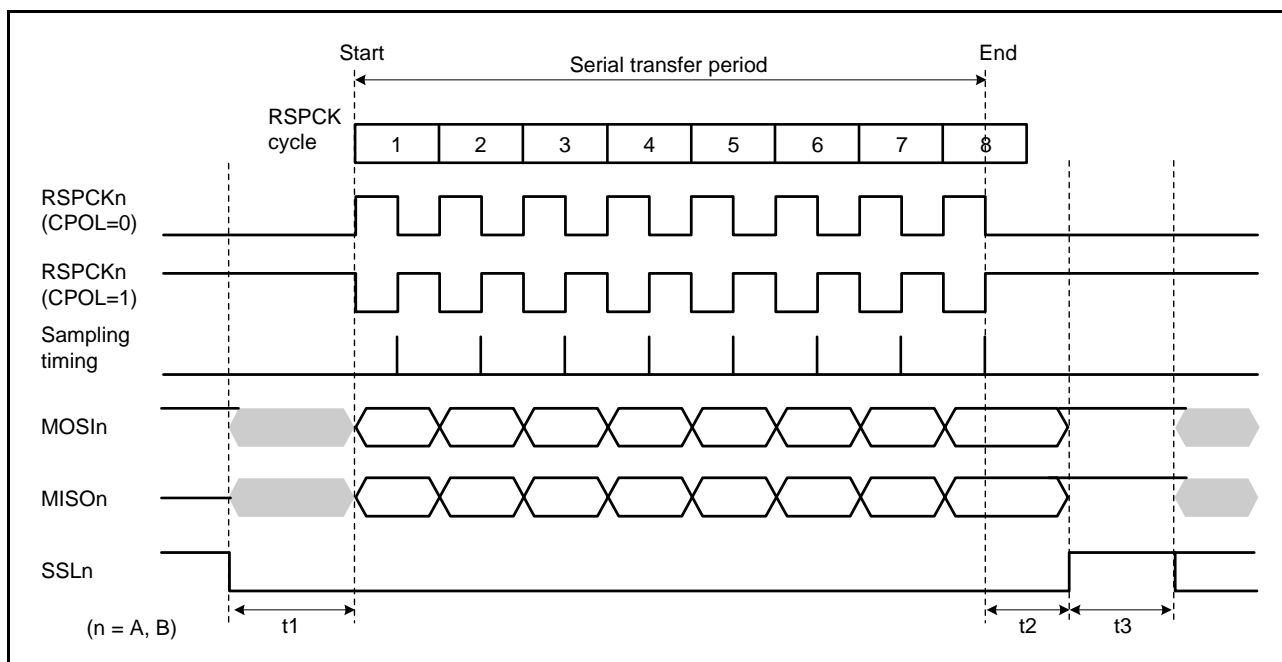


Figure 33.12 RSPI Transfer Format (CPHA = 1)

33.3.5 Data Format

The RSPI's data format depends on the settings in SPCMD and the SPCR2.SPPE bit. Irrespective of MSB/LSB first, the RSPI treats the range from the SPDR.LSB bit to the assigned data length as transfer data.

33.3.5.1 MSB First Transfer (32-Bit Data)

(1) Parity Function is Disabled (SPCR2.SPPE = 0)

Figure 33.13 shows the operation of SPDR and the shift register when the RSPI performs a 32-bit data length MSB-first data transfer with the parity function disabled.

Data T31 to T00 is written to the transmit buffer of SPDR. If the transmit buffer holds data and the shift register is empty, the RSPI copies the data in the transmit buffer of SPDR to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 31) of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycles required for the serial transfer of 32 bits have passed, data R31 to R00 is stored in the shift register. In this state, when performing full-duplex synchronous serial communications (SPCR.TXMD = 0), the RSPI copies the data from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before data is written to the transmit buffer of SPDR, received data R31 to R00 is shifted out from the shift register.

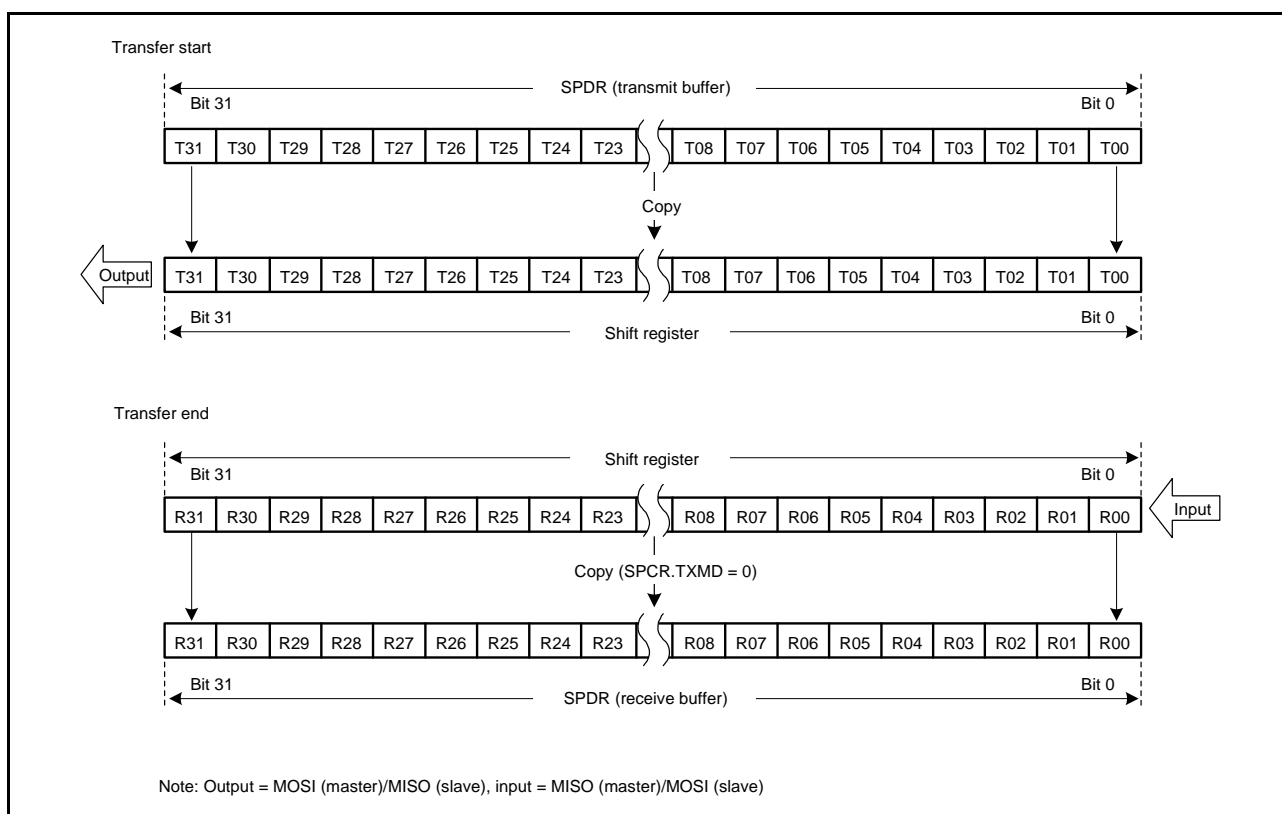


Figure 33.13 MSB First Transfer (1) (32-Bit Data, Parity Function Disabled)

(2) Parity Function is Enabled (SPCR2.SPPE = 1)

Figure 33.13 shows the operation of SPDR and the shift register when the RSPI performs a 32-bit data length MSB-first data transfer with the parity function enabled.

Data T31 to T00 is written to the transmit buffer of SPDR. If the transmit buffer holds data and the shift register is empty, the RSPI converts T00 of the data stored in the transmit buffer of SPDR into the parity bit (P). Then, the RSPI copies the data with the added parity bit (P) to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 31) of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycles required for the serial transfer of 32 bits have passed, data R31 to P is stored in the shift register. In this state, when performing full-duplex synchronous serial communications (SPCR.TXMD = 0), the RSPI copies the data from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before data is written to the transmit buffer of SPDR, received data R31 to P is shifted out from the shift register.

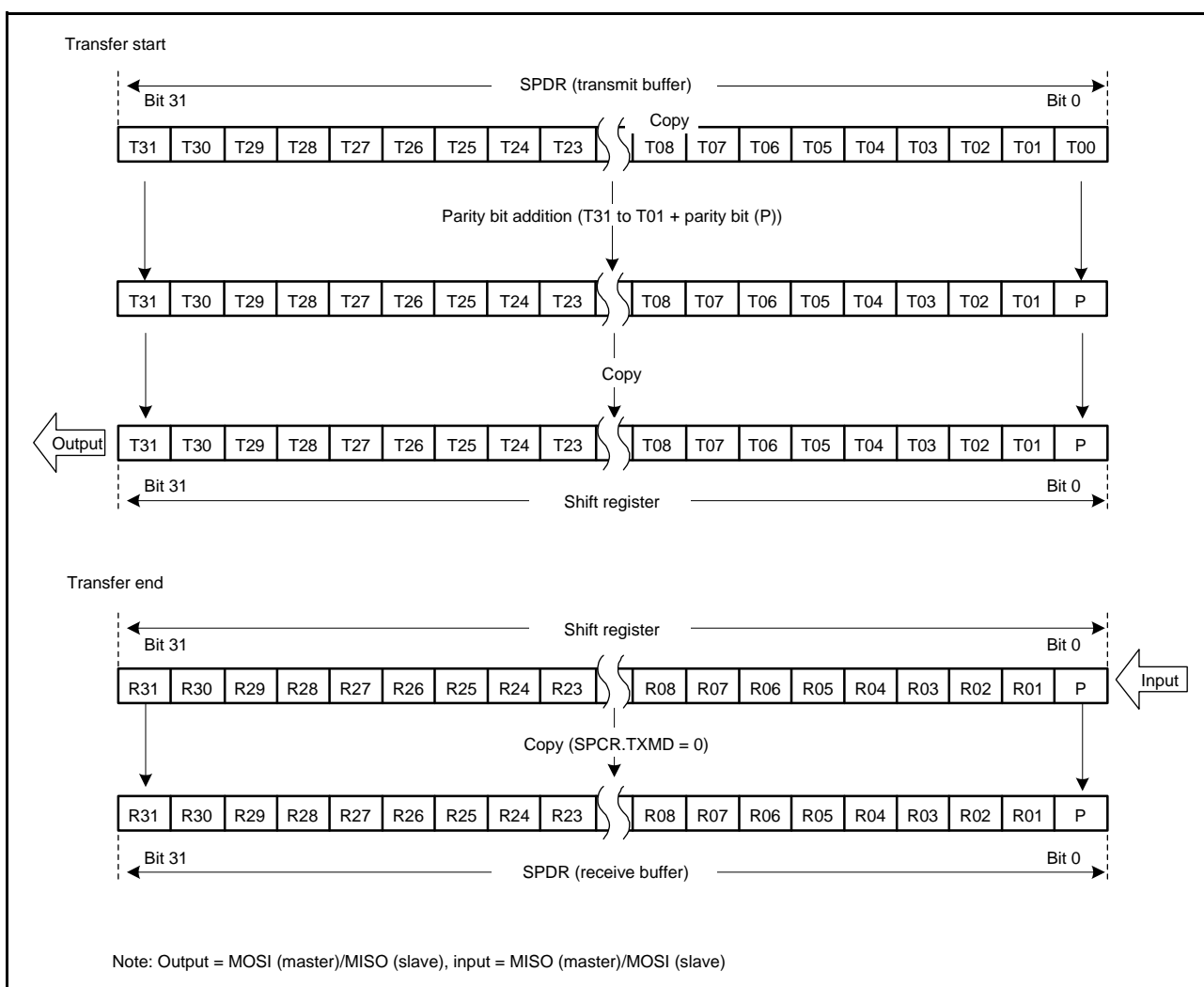


Figure 33.13 MSB First Transfer (2) (32-Bit Data, Parity Function Enabled)

33.3.5.2 MSB First Transfer (24-Bit Data)

(1) Parity Function is Disabled (SPCR2.SPPE = 0)

Figure 33.14 shows the operation of SPDR and the shift register when the RSPI performs a 24-bit data length MSB-first data transfer (as a data transfer example of lengths other than 32 bits) with the parity function disabled.

Data T31 to T00 is written to the transmit buffer of SPDR. If the transmit buffer holds data and the shift register is empty, the RSPI copies the data stored in the transmit buffer of SPDR to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from bit 23 of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycles required for the serial transfer of 24 bits have passed, received data R23 to R00 is stored in bits 23 to 0 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 31 to 24 of the shift register. In this state, when performing full-duplex synchronous serial communications (SPCR.TXMD = 0), the RSPI copies the data from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before data is written to the transmit buffer of SPDR, received data R23 to R00 is shifted out from the shift register.

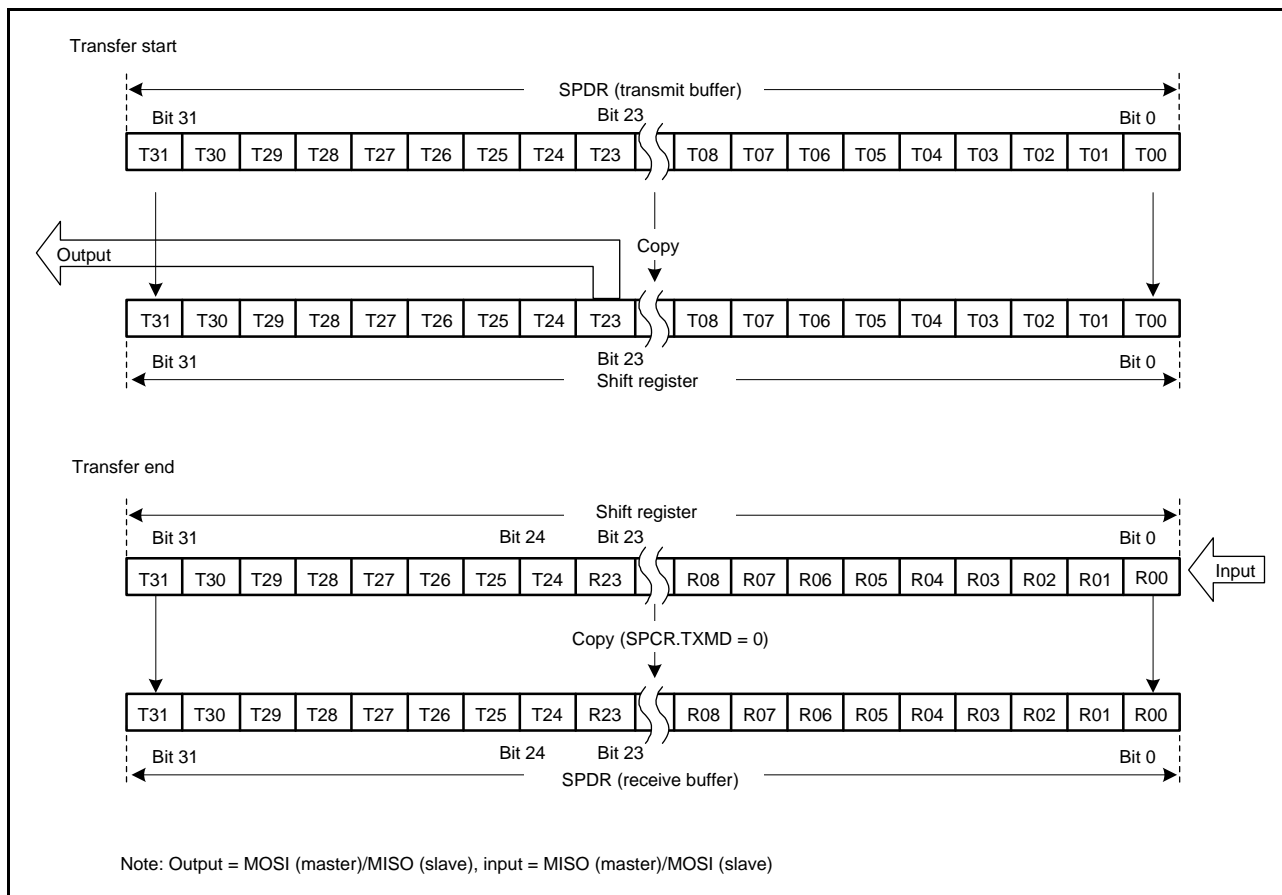


Figure 33.14 MSB First Transfer (1) (24-Bit Data, Parity Function Disabled)

(2) Parity Function is Enabled (SPCR2.SPPE = 1)

Figure 33.14 shows the operation of SPDR and the shift register when the RSPI performs a 24-bit data length MSB-first data transfer (as a data transfer example of lengths other than 32 bits) with the parity function enabled.

Data T31 to T00 is written to the transmit buffer of SPDR. If the transmit buffer holds data and the shift register is empty, the RSPI converts T00 of the data stored in the transmit buffer of SPDR into the parity bit (P). Then, the RSPI copies the data with the added parity bit (P) to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from bit 23 of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycles required for the serial transfer of 24 bits have passed, received data R23 to P is stored in bits 23 to 0 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 31 to 24 of the shift register. In this state, when performing full-duplex synchronous serial communications (SPCR.TXMD = 0), the RSPI copies the data from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before data is written to the transmit buffer of SPDR, received data R23 to P is shifted out from the shift register.

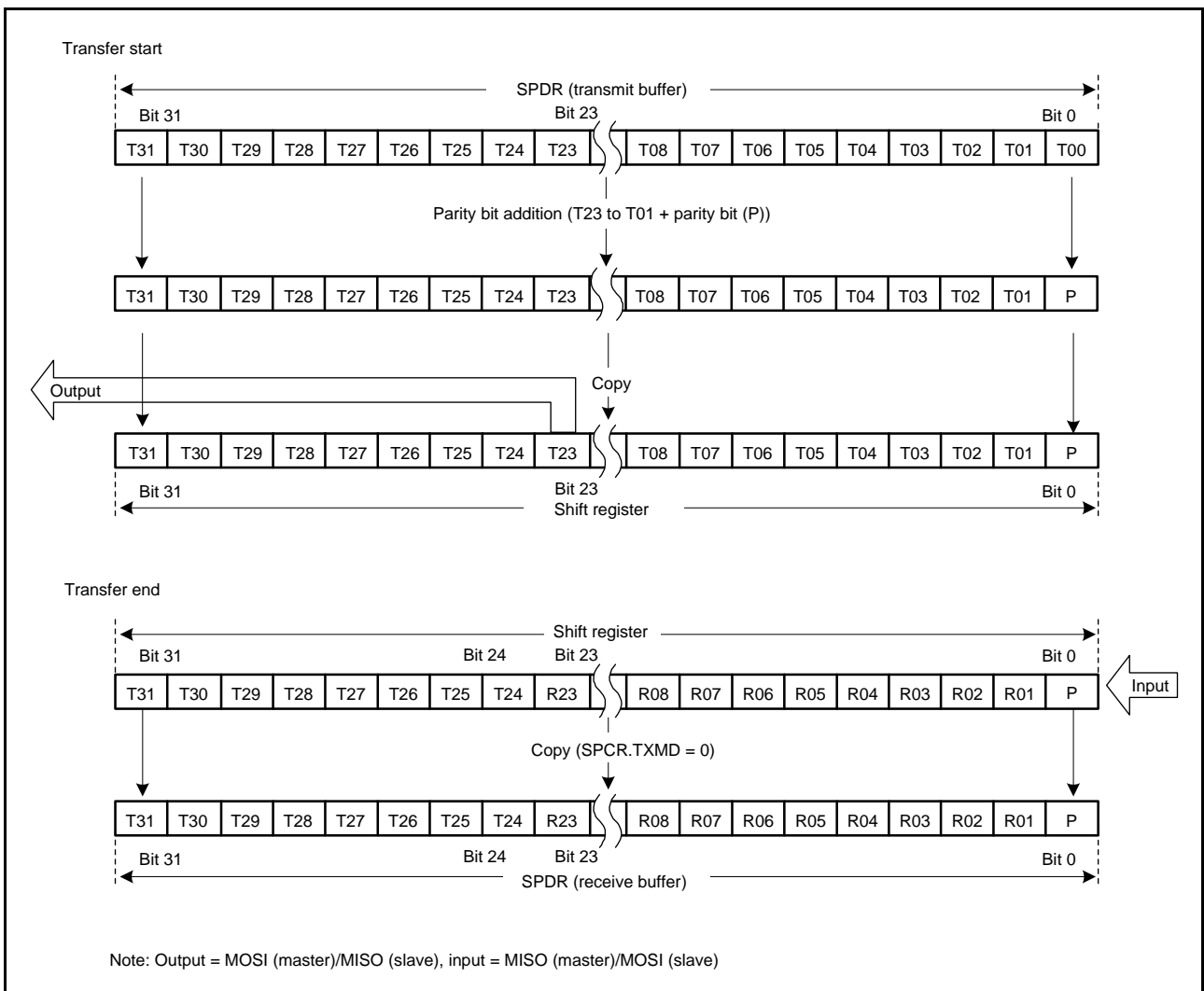


Figure 33.14 MSB First Transfer (2) (24-Bit Data, Parity Function Enabled)

33.3.5.3 LSB First Transfer (32-Bit Data)

(1) Parity Function is Disabled (SPCR2.SPPE = 0)

Figure 33.15 shows the operation of SPDR and the shift register when the RSPI performs a 32-bit data length LSB-first data transfer with the parity function disabled.

Data T31 to T00 is written to the transmit buffer of SPDR. If the transmit buffer holds data and the shift register is empty, the RSPI reverses the order of the bits of the data in the transmit buffer of SPDR, copies it to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 31) of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycles required for the serial transfer of 32 bits have passed, data R00 to R31 is stored in the shift register. In this state, when performing full-duplex synchronous serial communications (SPCR.TXMD = 0), the RSPI copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before data is written to the transmit buffer of SPDR, received data R00 to R31 is shifted out from the shift register.

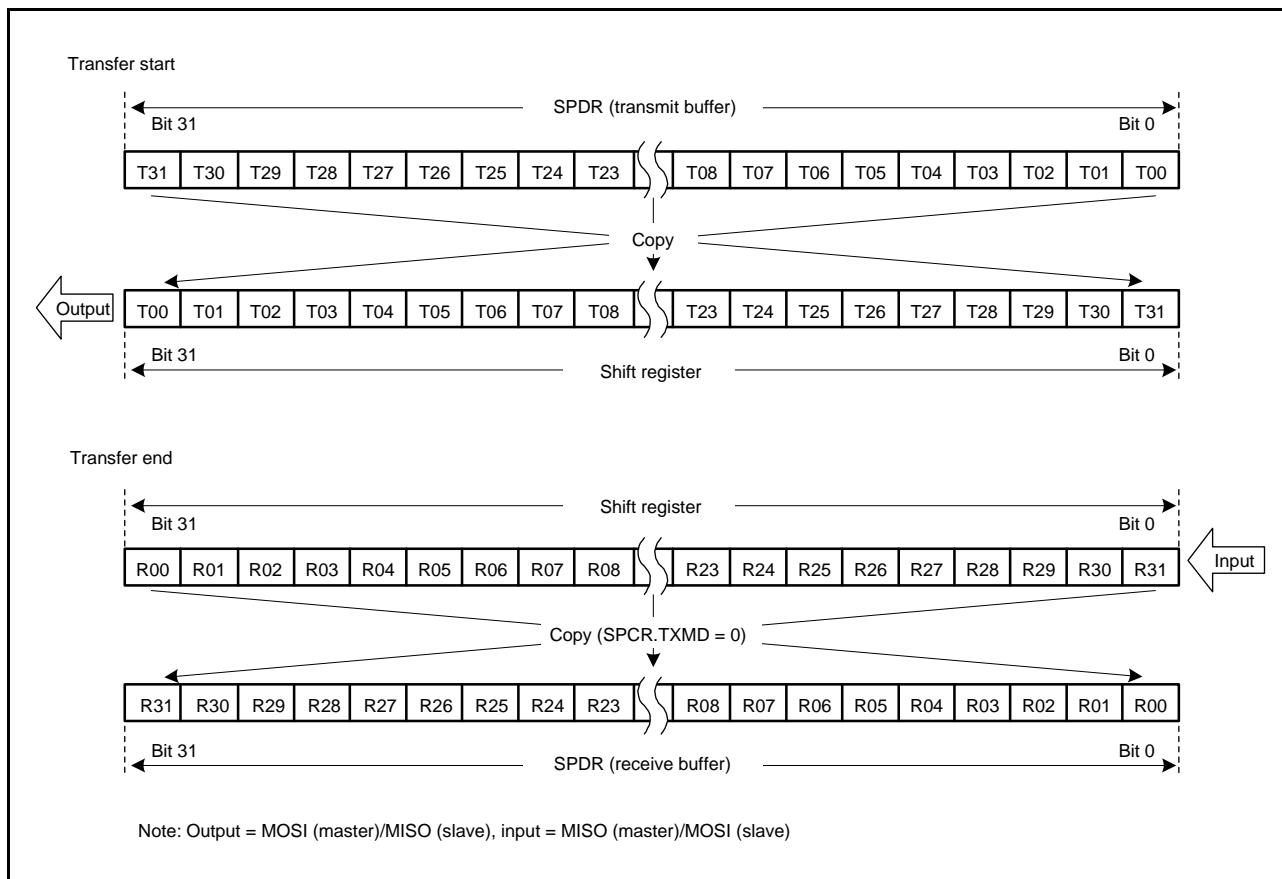


Figure 33.15 LSB First Transfer (1) (32-Bit Data, Parity Function Disabled)

(2) Parity Function is Enabled (SPCR2.SPPE = 1)

Figure 33.15 shows the operation of SPDR and the shift register when the RSPI performs a 32-bit data length LSB-first data transfer with the parity function enabled.

Data T31 to T00 is written to the transmit buffer of SPDR. The RSPI converts T31 of the data stored in the transmit buffer of SPDR into the parity bit (P). If the transmit buffer holds data and the shift register is empty, the RSPI reverses the order of the bits of the data with the added parity bit (P), copies it to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 31) of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycles required for the serial transfer of 32 bits have passed, data R00 to P is stored in the shift register. In this state, when performing full-duplex synchronous serial communications (SPCR.TXMD = 0), the RSPI copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before data is written to the transmit buffer of SPDR, received data R00 to P is shifted out from the shift register.

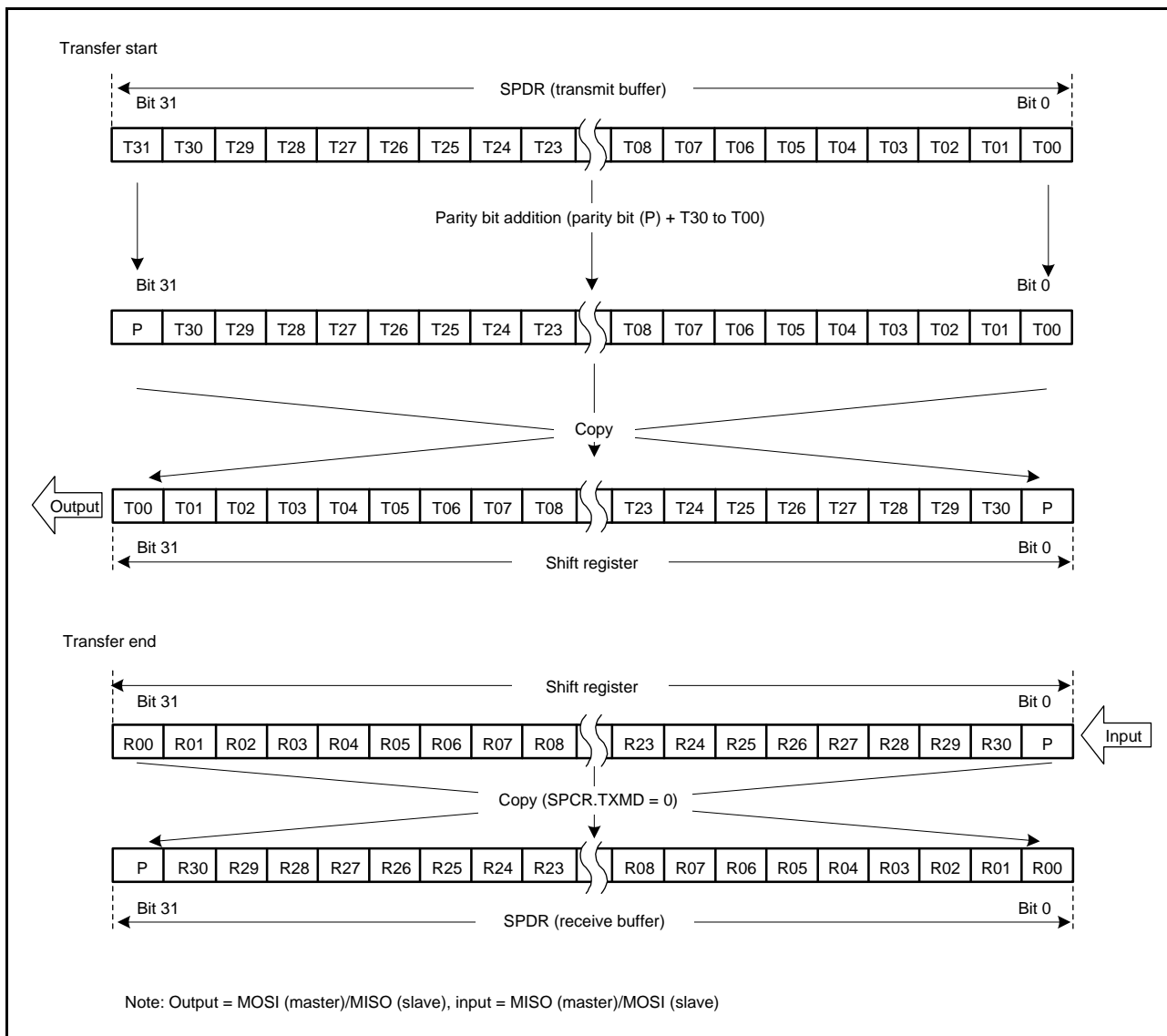


Figure 33.15 LSB First Transfer (2) (32-Bit Data, Parity Function Enabled)

33.3.5.4 LSB First Transfer (24-Bit Data)

(1) Parity Function is Disabled (SPCR2.SPPE = 0)

Figure 33.16 shows the operation of SPDR and the shift register when the RSPI performs a 24-bit data length LSB-first data transfer (as a data transfer example of lengths other than 32 bits) with the parity function disabled.

Data T31 to T00 is written to the transmit buffer of SPDR. If the transmit buffer holds data and the shift register is empty, the RSPI reverses the order of the bits of the data in the transmit buffer of SPDR, copies it to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 31) of the shift register, and shifts in the data from bit 8 of the shift register. When the RSPCK cycles required for the serial transfer of 24 bits have passed, received data R00 to R23 is stored in bits 31 to 8 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 7 to 0 of the shift register. In this state, when performing full-duplex synchronous serial communications (SPCR.TXMD = 0), the RSPI copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before data is written to the transmit buffer of SPDR, received data R00 to R23 is shifted out from the shift register.

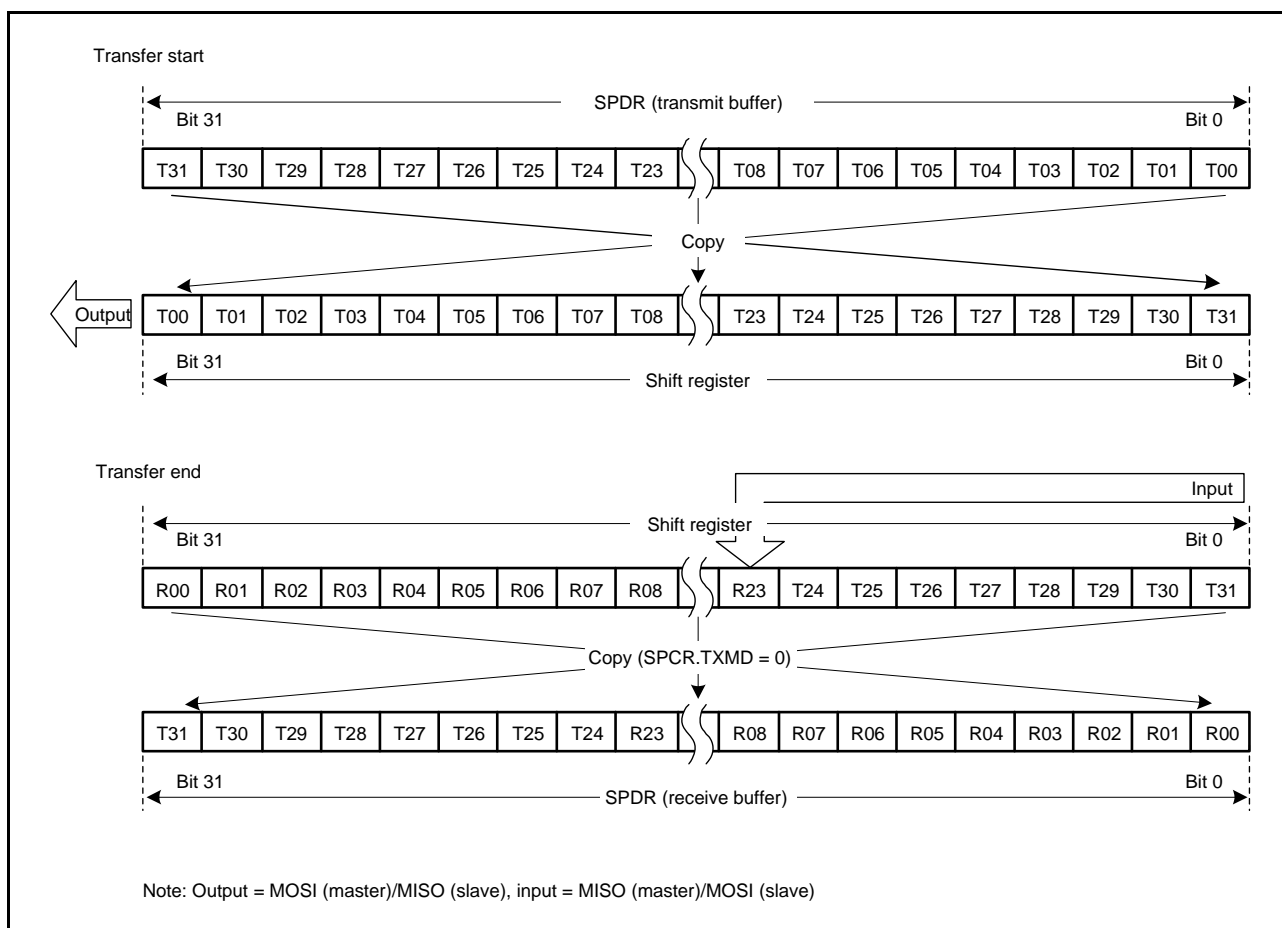


Figure 33.16 LSB First Transfer (1) (24-Bit Data, Parity Function Disabled)

(2) Parity Function is Enabled (SPCR2.SPPE = 1)

Figure 33.16 shows the operation of SPDR and the shift register when the RSPI performs a 24-bit data length LSB-first data transfer (as a data transfer example of lengths other than 32 bits) with the parity function enabled.

Data T31 to T00 is written to the transmit buffer of SPDR. The RSPI converts T23 of the data stored in the transmit buffer of SPDR into the parity bit (P). If the transmit buffer holds data and the shift register is empty, the RSPI reverses the order of the bits of the data with the added parity bit (P), copies it to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 31) of the shift register, and shifts in the data from bit 8 of the shift register. When the RSPCK cycles required for the serial transfer of 24 bits have passed, received data R00 to P is stored in bits 31 to 8 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 7 to 0 of the shift register. In this state, when performing full-duplex synchronous serial communications (SPCR.TXMD = 0), the RSPI copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before data is written to the transmit buffer of SPDR, received data R00 to P is shifted out from the shift register.

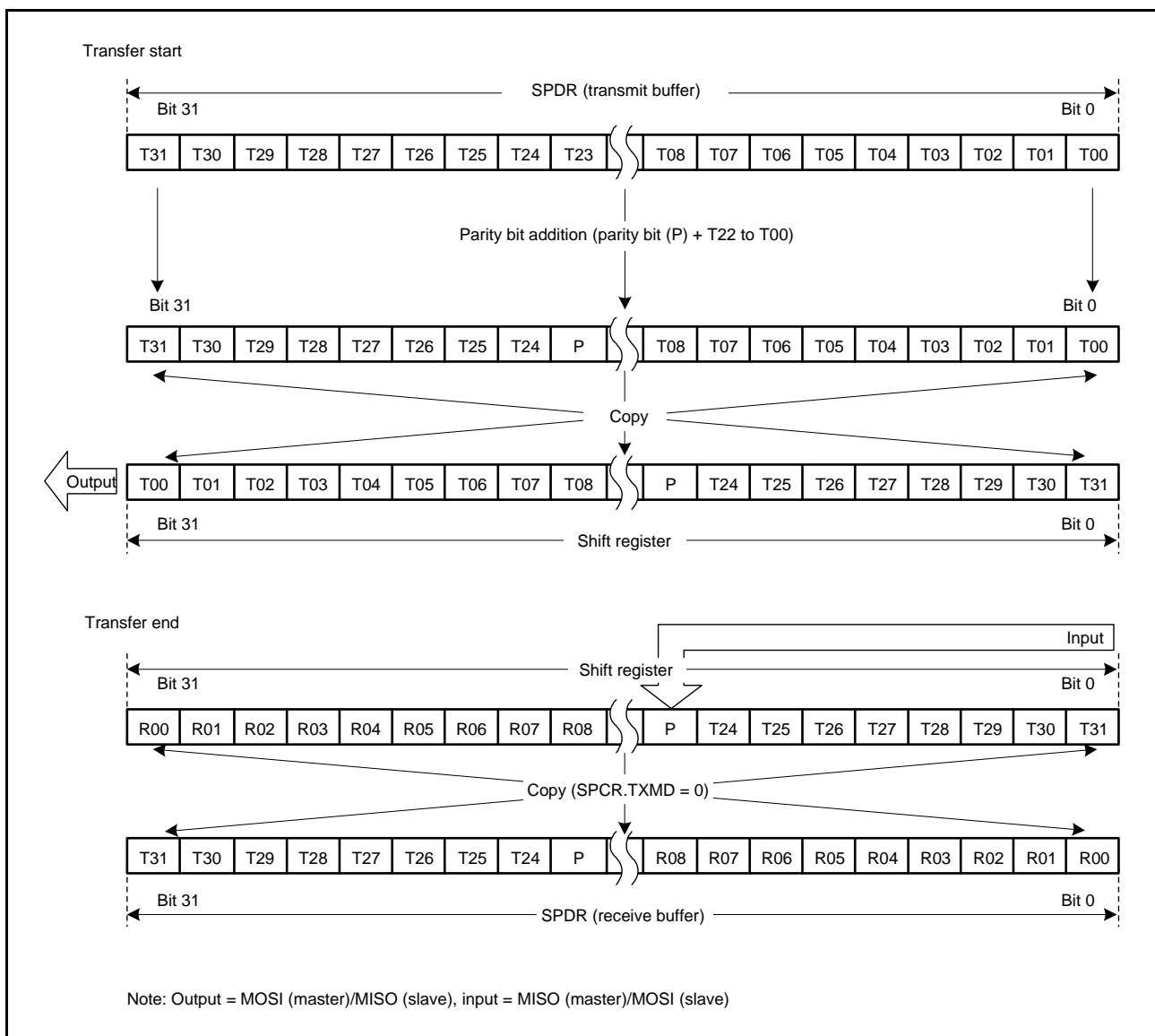


Figure 33.16 LSB First Transfer (2) (24-Bit Data, Parity Function Enabled)

33.3.6 Communications Operating Mode

Full-duplex synchronous serial communications or transmit operations only can be selected by the communications operating mode select bit (TXMD) in the RSPI control register (SPCR). The SPDR access shown in Figure 33.17 and Figure 33.18 indicates the condition of access to the RSPI data register (SPDR), where I denotes an idle cycle and W a write cycle.

33.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)

Figure 33.17 shows an example of operation when the communications operating mode select bit (TXMD) in the RSPI control register (SPCR) is set to 0. In the example of Figure 33.17, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMD.CPHA bit is 1, and the SPCMD.CPOL bit is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

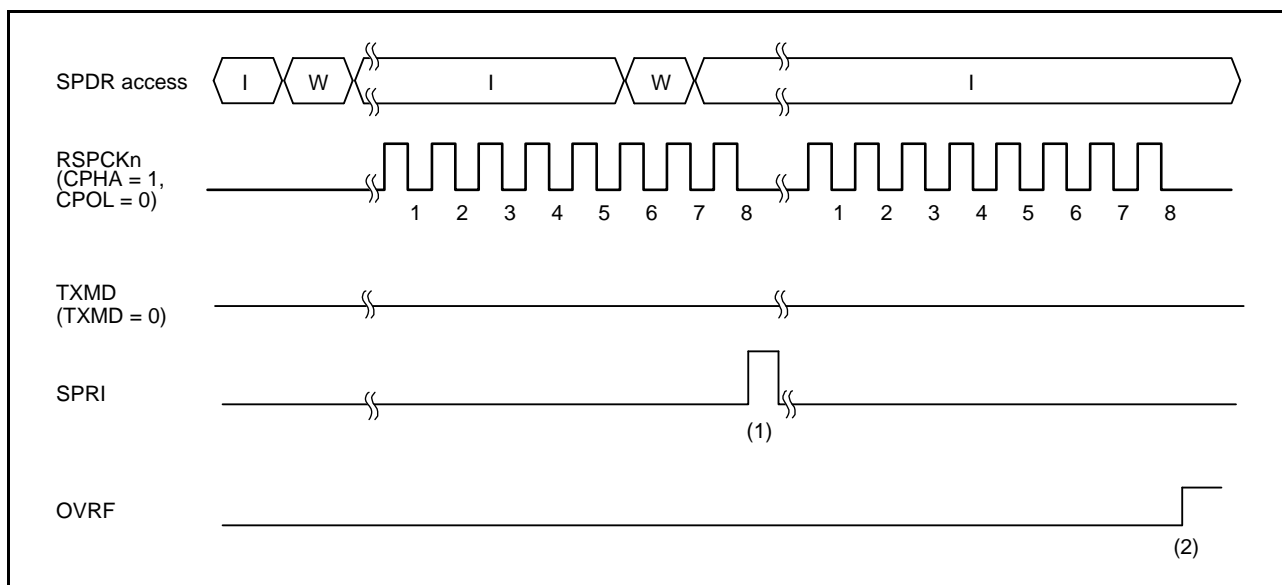


Figure 33.17 Operation Example of SPCR.TXMD = 0

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 OVRF flag to 1 and discards the received data in the shift register.

When performing full-duplex synchronous serial communications (SPCR.TXMD = 0), the RSPI transmits transmit data and receives received data. Therefore, the OVRF flag is set to 1 at the timings of (1) and (2).

33.3.6.2 Transmit Operations Only (SPCR.TXMD = 1)

Figure 33.18 shows an example of operation when the communications operating mode select bit (TXMD) in the RSPI control register (SPCR) is set to 1. In the example of Figure 33.18, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMD.CPHA bit is 1, and the SPCMD.CPOL bit is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

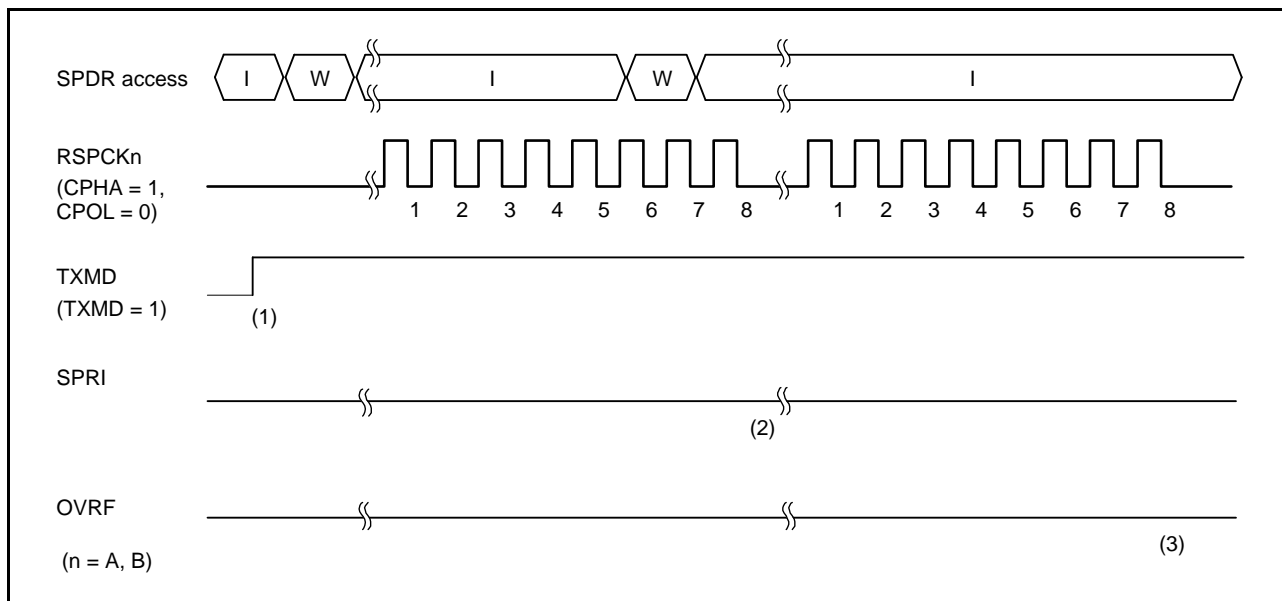


Figure 33.18 Operation Example of SPCR.TXMD = 1

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 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 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 OVRF flag remains cleared to 0 at the timings of (1) to (3).

33.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 33.19 shows an example of operation of the RSPI transmit buffer empty interrupt (SPTI) and the RSPI receive buffer full interrupt (SPRI). The SPDR access shown in Figure 33.19 indicates the condition of access to the RSPI data register (SPDR), where I denotes an idle cycle, W a write cycle, and R a read cycle. In the example of Figure 33.19, 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 SPCMD.CPHA bit is 1, and the SPCMD.CPOL bit is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

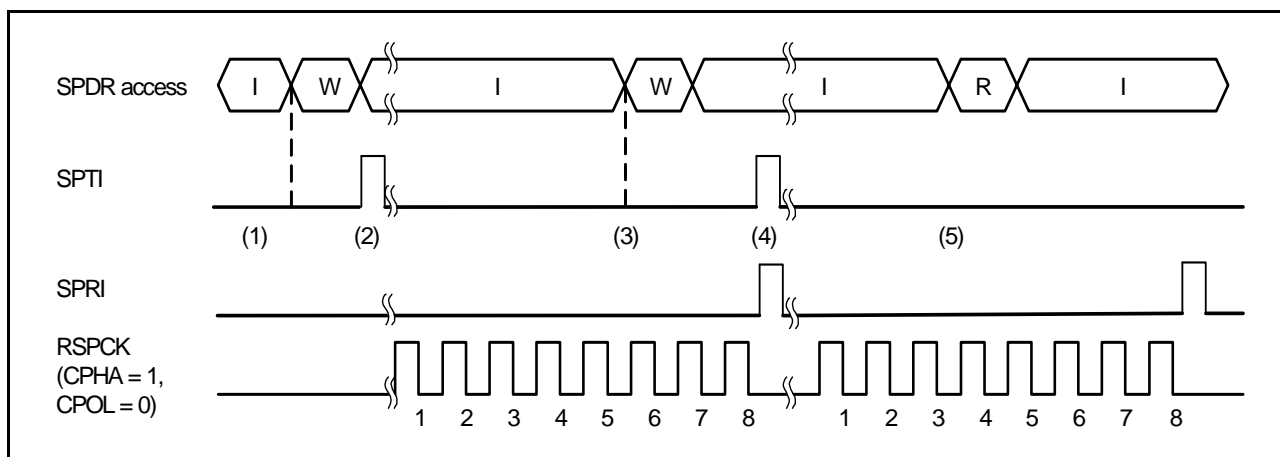


Figure 33.19 Operation Example of SPTI and SPRI Interrupts

The operation of the interrupts at timings shown in steps (1) to (5) in the figure is described below.

1. When transmit data is written to SPDR when the transmit buffer of SPDR is empty (data for the next transfer is not set), the RSPI writes data to the transmit buffer.
2. If the shift register is empty, the RSPI copies the data in the transmit buffer to the shift register and generates a transmit buffer empty interrupt request (SPTI). How a serial transfer is started depends on the mode of the RSPI. For details, see section 33.3.10, SPI Operation, and section 33.3.11, 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). Because 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 RSPI sends the data in the receive buffer to the bus inside the chip.

If SPDR is written to when the transmit buffer holds data that has not yet been transmitted, the RSPI does not update the data in the transmit buffer. When writing to SPDR, make sure to use a transmit buffer empty interrupt request. To use an RSPI transmit interrupt, set the SPTIE bit in SPCR to 1.

If the RSPI is disabled (the SPE bit in SPCR being 0), set the SPTEF bit to 0.

When serial transfer ends with the receive buffer being full, the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (see section 33.3.8, Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an RSPI receive interrupt, set the SPRIE bit in SPCR to 1.

The status of the transmit/receive buffer can be checked by either using a transmit/receive interrupt or by reading the corresponding IR.IRn flag of the ICU.

33.3.8 Error Detection

In the normal RSPI serial transfer, the data written to the transmit buffer of SPDR is serially transmitted, and the serially received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit/receive buffer or the status of the RSPI at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, the RSPI detects the event as an overrun error, parity error, or mode fault error. Table 33.9 shows the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 33.9 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function

	Occurrence Condition	RSPI Operation	Error Detection
A	SPDR is written when the transmit buffer is full.	The contents of the transmit buffer are kept. Missing write data.	None
B	Serial transfer is started in slave mode when transmit data is still not loaded on the shift register.	Data received in previous serial transfer is serially transmitted.	None
C	SPDR is read when the receive buffer is empty.	Previously received serial data is output.	None
D	Serial transfer terminates when the receive buffer is full.	The contents of the receive buffer are kept. Missing serial receive data.	Overrun error
E	An incorrect parity bit is received when performing full-duplex synchronous serial communications with the parity function enabled.	The parity error flag is asserted.	Parity error
F	The SSL0 input signal is asserted when the serial transfer is idle in multi-master mode.	Driving of the RSPCK, MOSI, SSL1 to SSL3 output signals is stopped. RSPI is disabled.	Mode fault error
G	The SSL0 input signal is asserted during serial transfer in multi-master mode.	Serial transfer is suspended. Missing transmit/receive data. Driving of the RSPCK, MOSI, SSL1 to SSL3 output signals is stopped. RSPI is disabled.	Mode fault error
H	The SSL0 input signal is negated during serial transfer in slave mode.	Serial transfer is suspended. Missing transmit/receive data. Driving of the MISO output signal is stopped. RSPI is disabled.	Mode fault error

On operation A shown in Table 33.9, the RSPI does not detect an error. To prevent data omission during the writing to SPDR, write operations to SPDR should be executed using a transmit interrupt request.

Likewise, the RSPI does not detect an error on operation B. In a serial transfer that was started before the shift register was updated, the RSPI sends the data that was received in the previous serial transfer, and does not treat the operation indicated in B as an error. Note that the received data from the previous serial transfer is retained in the receive buffer of SPDR, thus it can be correctly read (if SPDR is not read before the end of the serial transfer, an overrun error may occur).

Similarly, the RSPI does not detect an error on operation C. To prevent extraneous data from being read, SPDR read operation should be executed using a receive interrupt request.

An overrun error shown in D is described in section 33.3.8.1, **Overrun Error**. A parity error shown in E is described in section 33.3.8.2, **Parity Error**. A mode fault error shown in F to H is described in section 33.3.8.3, **Mode Fault Error**.

For the transmit and receive interrupts, refer to section 33.3.7, **Transmit Buffer Empty/Receive Buffer Full Interrupts**.

33.3.8.1 Overrun Error

If a serial transfer ends when the receive buffer of SPDR is full, the RSPI detects an overrun error, and sets the OVRF bit in SPSR to 1. When the OVRF bit 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 bit in SPSR to 0, write 0 to the OVRF bit after the CPU has read SPSR with the OVRF bit set to 1.

Figure 33.20 shows an example of operation of the SPSR.OVRF flag. The SPSR and SPDR accesses shown in Figure 33.20 indicates the condition of accesses to SPSR and SPDR, respectively, where I denotes an idle cycle, W a write cycle, and R a read cycle. In the example of Figure 33.20, the RSPI performs an 8-bit serial transfer in which the SPCMD.CPHA bit is 1 and the SPCMD.CPOL bit is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

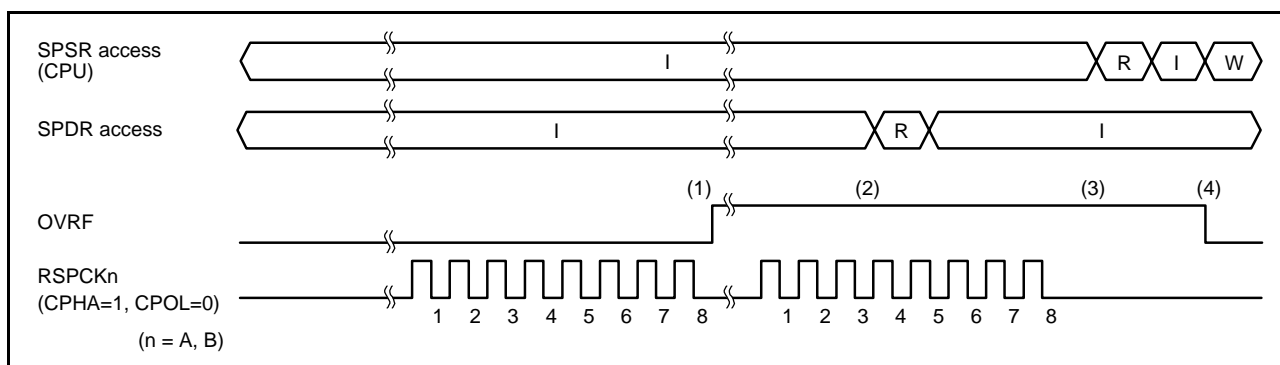


Figure 33.20 Operation Example of OVRF Flag

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 bit 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 SPCMD to the SPSR.SPECM[2:0] bits.
2. When SPDR is read, the RSPI outputs the data in the receive buffer to an internal bus. The receive buffer becoming empty does not clear the OVRF bit.
3. If the serial transfer ends with the OVRF bit being 1 (an overrun error), 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. When in master mode, the RSPI does not update the SPSR.SPECM[2:0] bits. When in an overrun error state and the RSPI does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPI determines that the shift register is empty; in this manner, data transfer from the transmit buffer to the shift register is enabled.
4. If the CPU writes the value 0 to the OVRF bit after reading SPSR when the OVRF bit is 1, the RSPI clears the OVRF bit.

The occurrence of an overrun can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When using an RSPI error interrupt, set the SPEIE bit in the RSPI control register (SPCR) to 1. When executing a serial transfer without using an RSPI error interrupt, 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 SPCMD at the occurrence of the error can be checked by reading the SPSR.SPECM[2:0] bits.

If an overrun error occurs and the OVRF bit is set to 1, normal reception operations cannot be performed until the OVRF bit is cleared. The OVRF bit is cleared to 0 under the following condition:

[Clearing condition]

Writing of "0" to the flag (OVRF) after the CPU has read register SPSR while the flag's value was "1".

33.3.8.2 Parity Error

If full-duplex synchronous serial communications is performed with the TXMD bit in the RSPI control register (SPCR) cleared to 0 and the SPPE bit in RSPI control register 2 (SPCR2) 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 PERF bit in the RSPI status register (SPSR) to 1. Since the RSPI does not copy the data in the shift register to the receive buffer when the OVRF bit is set to 1, parity error detection is not performed for the received data. To set the PERF bit in SPSR to 0, write 0 to the PERF bit after the CPU has read SPSR with the PERF bit set to 1.

Figure 33.21 shows an example of operation of the OVRF and PERF bits in SPSR. The SPSR access shown in Figure 33.21 indicates the condition of access to SPSR, where I denotes an idle cycle, W a write cycle, and R a read cycle. In the example of Figure 33.21, full-duplex synchronous serial communications is performed while the TXMD bit in the RSPI control register (SPCR) is 0 and the SPPE bit in RSPI control register 2 (SPCR2) is 1. The RSPI performs an 8-bit serial transfer in which the SPCMD.CPHA bit is 1 and the SPCMD.CPOL bit is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

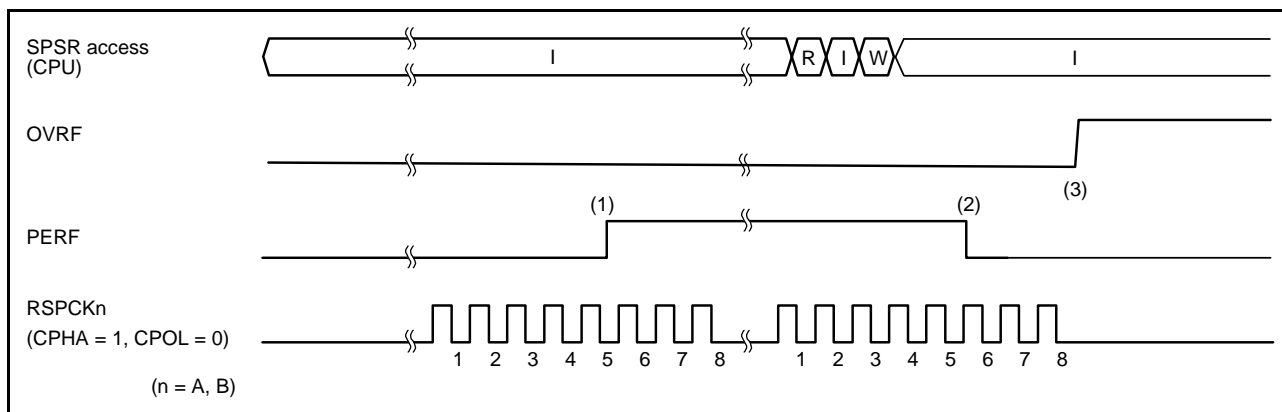


Figure 33.21 Operation Example of PERF Flag

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 bit to 1 if a parity error is detected. In master mode, the RSPI copies the pointer value to SPCMD to the SPSSR.SPECM[2:0] bits.
2. If the CPU writes the value 0 to the PERF bit after reading SPSR when the PERF bit is 1, the RSPI clears the PERF flag to 0.
3. When the RSPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The RSPI does not perform parity error detection at this timing.

The occurrence of a parity error can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When using an RSPI error interrupt, set the SPEIE bit in the RSPI control register (SPCR) to 1. When executing a serial transfer without using an RSPI error interrupt, measures should be taken to ensure the early detection of parity errors, such as reading SPSR immediately after SPDR is read. When the RSPI is used in master mode, the pointer value to SPCMD at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

The PERF bit is cleared to 0 under the following condition:

[Clearing condition]

Writing of "0" to the flag (PERF) after the CPU has read register SPSR while the flag's value was "1".

33.3.8.3 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 SSL0 input signal of the RSPI in multi-master mode, the RSPI detects a mode fault error irrespective of the status of the serial transfer, and sets the MODF bit in the RSPI status register (SPSR) to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to SPCMDm to the SPSSR.SPECM[2:0] bits. The active level of the SSL0 signal is determined by the SSL0P bit in the RSPI slave select polarity register (SSLP).

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit in the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSL0 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 SPE bit in SPCR to 0 (see section 33.3.9, 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. When using an RSPI error interrupt, set the SPEIE bit in the RSPI control register (SPCR) to 1. To detect a mode fault error without using an RSPI error interrupt, it is necessary to poll SPSR. When using the RSPI in master mode, the pointer value to SPCMD at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

When the MODF bit is 1, the RSPI ignores the writing of the value 1 to the SPE bit by the CPU. To enable the RSPI function after the detection of a mode fault error, the MODF bit must be set to 0.

The MODF bit is cleared to 0 under the following condition:

[Clearing condition]

Writing of "0" to the flag (MODF) after the CPU has read register SPSR while the flag's value was "1".

33.3.9 Initializing RSPI

If the CPU writes the value 0 to the SPE bit in the RSPI control register (SPCR) or the RSPI clears the SPE bit to 0 because of the detection of a mode fault error, the RSPI disables the RSPI function, and initializes some of the module functions. When a system reset is generated, the RSPI initializes all of the module functions. The following describes initialization by the clearing of the SPE bit and initialization by a system reset.

33.3.9.1 Initialization by Clearing the SPE Bit

When the SPE bit in SPCR is cleared, the RSPI performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (high-impedance) 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 CPU resets the value 1 to the SPE bit.

The OVRF and MODF flags in SPSR are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPI is initialized, data from the receive buffer can be read in order to check the status of error occurrence during an RSPI transfer.

The transmit buffer is initialized to an empty state. Therefore, if the SPTIE bit in SPCR is set to 1 after RSPI initialization, an RSPI transmit interrupt is generated. When the RSPI is initialized by the CPU, in order to disable any RSPI transmit interrupt, the value 0 should be written to the SPTIE bit simultaneously with the writing of the value 0 to the SPE bit. To disable any RSPI transmit interrupt after a mode fault error is detected, use an error handling routine to write the value 0 to the SPTIE bit.

33.3.9.2 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 33.3.9.1, Initialization by Clearing the SPE Bit.

33.3.10 SPI Operation

33.3.10.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (see section 33.3.8, 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 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). If the shift register is empty due to the writing to SPDR, the RSPI copies the data in the transmit buffer to the shift register and starts a 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 from the CPU.

For details on the RSPI transfer format, see section 33.3.4, Transfer Format. The polarity of the SSL output pins depends on the settings in the RSPI slave select polarity register (SSLP).

(2) Terminating a Serial Transfer

Irrespective of the CPHA bit in the RSPI command register (SPCMD), the RSPI terminates the serial transfer after transmitting an RSPCK edge corresponding to the final 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 SPCMD.SPB[3:0] bit setting. The polarity of the SSL_n output pin (n = A, B) depends on the settings in the RSPI slave select polarity register (SSLP).

For details on the RSPI transfer format, see section 33.3.4, Transfer Format.

(3) Sequence Control

The transfer format that is employed in master mode is determined by SPSCR, SPCMD, SPBR, SPCKD, SSLND, and SPND.

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 RSPI command registers SPCMD0 to SPCMD7: SSL_n pin output signal value (n = A, B), 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 SPCMD0 to SPCMD7. The RSPI contains a pointer to the SPCMD that makes up the sequence. The CPU can check the value of this pointer by reading the SPSSR.SPCP[2:0] bits. When the SPE bit in the RSPI control register (SPCR) is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0, 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.

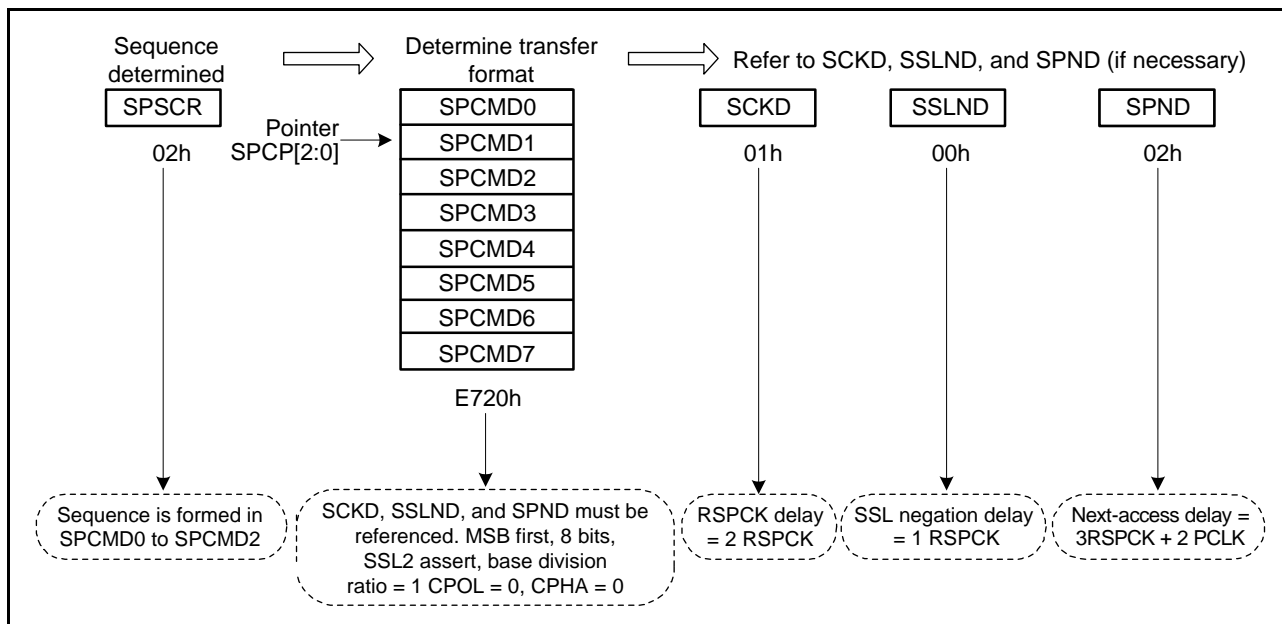


Figure 33.22 Determination Procedure of Serial Transfer Mode in Master Mode (SPI Operation)

(4) Burst Transfer

If the SSLKP bit in the RSPI command register (SPCMD) that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSL signal level during the serial transfer until the beginning of the SSL signal assertion for the next serial transfer. If the SSL signal level for the next serial transfer is the same as the SSL signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLn (n = A, B) signal assertion status (burst transfer).

Figure 33.23 shows an example of an SSL signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 settings. The text below explains the RSPI operations (1) to (7) as shown in Figure 33.23. It should be noted that the polarity of the SSL output signal depends on the settings in the RSPI slave select polarity register (SSLP).

1. Based on SPCMD0, the RSPI asserts the SSL signal and inserts RSPCK delays.
2. The RSPI executes serial transfers according to SPCMD0.
3. The RSPI inserts SSL negation delays.
4. Because the SSLKP bit in SPCMD0 is 1, the RSPI keeps the SSL 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 SSLn signal and inserts RSPCK delays.
6. The RSPI executes serial transfers according to SPCMD1.
7. Because the SSLKP bit in SPCMD1 is 0, the RSPI negates the SSL signal. In addition, a next-access delay is inserted according to SPCMD1.

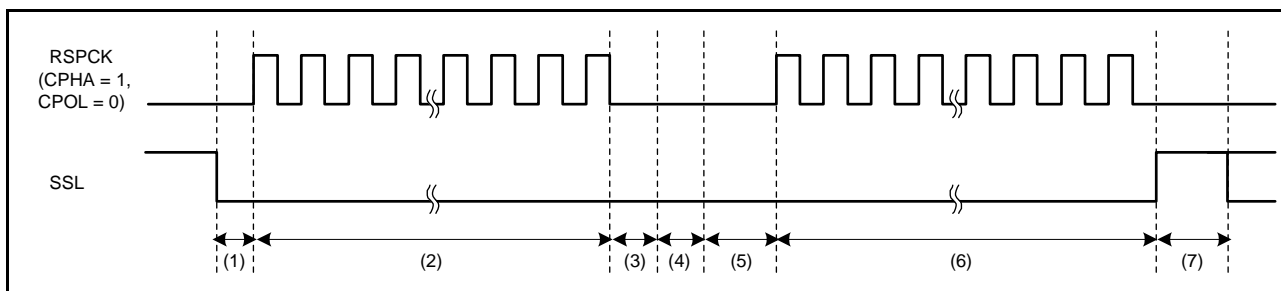


Figure 33.23 Example of Burst Transfer Operation using SSLKP Bit

If the SSLn signal output settings in the SPCMD in which 1 is assigned to the SSLKP bit are different from the SSL signal output settings in the SPCMD to be used in the next transfer, the RSPI switches the SSL signal status to SSL signal assertion ((5) in Figure 33.23) corresponding to the command for the next transfer. Note that if such an SSL signal switching occurs, the slaves that drive the MISO signal compete, and collision of signal levels may occur.

The RSPI in master mode references the SSL signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA (m = 0 to 7) bit is 0, the RSPI can accurately start serial transfers by asserting the SSLn signal for the next transfer. For this reason, burst transfers in master mode can be executed irrespective of CPHA bit settings (see section 33.3.10.2, Slave Mode Operation). (n = A, B)

(5) RSPCK Delay (t1)

The RSPCK delay value of the RSPI in master mode depends on the SPCMD.SCKDEN bit setting and the SPCKD setting. The RSPI determines the SPCMD to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SCKDEN bit in the selected SPCMD and SPCKD, as shown in Table 33.10. For a definition of RSPCK delay, see section 33.3.4, Transfer Format.

Table 33.10 Relationship among SCKDEN Bit, SPCKD, and RSPCK Delay Value

SPCMDm.SCKDEN Bit	SPCKD	RSPCK Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

[Legend] m = 0 to 7

(6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPI in master mode depends on the SPCMD.SLNDEN bit setting and the SSLND setting. The RSPI determines the SPCMD to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SLNDEN bit in the selected SPCMD and SSLND, as shown in Table 33.11. For a definition of SSL negation delay, see section 33.3.4, Transfer Format.

Table 33.11 Relationship among SLNDEN Bit, SSLND, and SSL Negation Delay Value

SPCMDm.SLNDEN Bit	SSLND	SSL Negation Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

[Legend] m = 0 to 7

(7) Next-Access Delay (t_3)

The next-access delay value of the RSPI in master mode depends on the SPCMD.SPNDEN bit setting and the SPND setting. The RSPI determines the SPCMD to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPNDEN bit in the selected SPCMD and SPND, as shown in Table 33.12. For a definition of next-access delay, see section 33.3.4, Transfer Format.

Table 33.12 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value

SPCMDm.SLNDEN Bit	SPND	Next-Access Delay Value
0	000 to 111	1 RSPCK + 2 PCLK
1	000	1 RSPCK + 2 PCLK
	001	2 RSPCK + 2 PCLK
	010	3 RSPCK + 2 PCLK
	011	4 RSPCK + 2 PCLK
	100	5 RSPCK + 2 PCLK
	101	6 RSPCK + 2 PCLK
	110	7 RSPCK + 2 PCLK
	111	8 RSPCK + 2 PCLK

[Legend] m = 0 to 7

(8) Initialization Flowchart

Figure 33.24 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, DMACA, and input/output ports, see the descriptions given in the individual blocks.

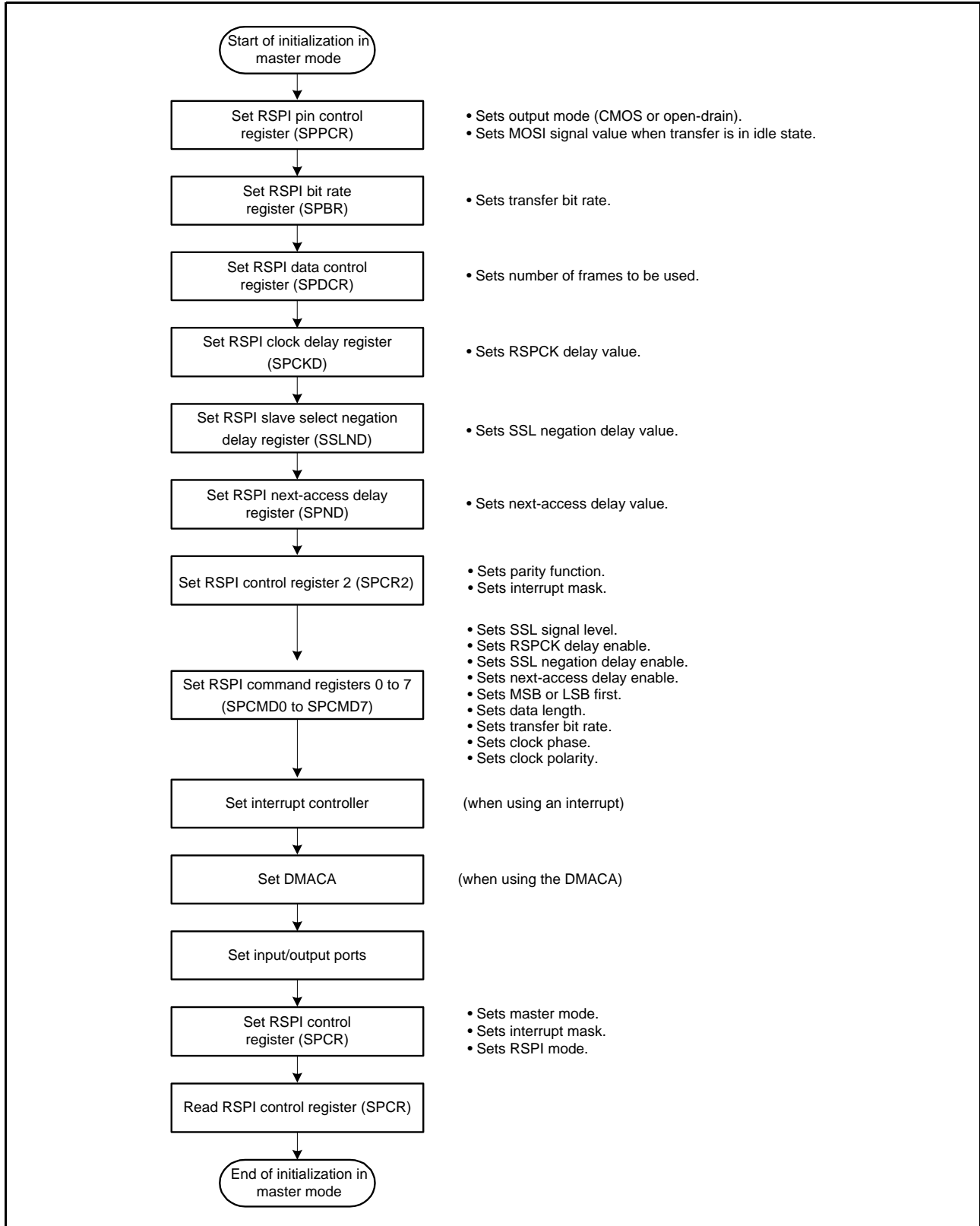


Figure 33.24 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Transfer Operation Flowchart

Figure 33.25 is a flowchart illustrating a transfer in SPI operation when the RSPI is used in master mode.

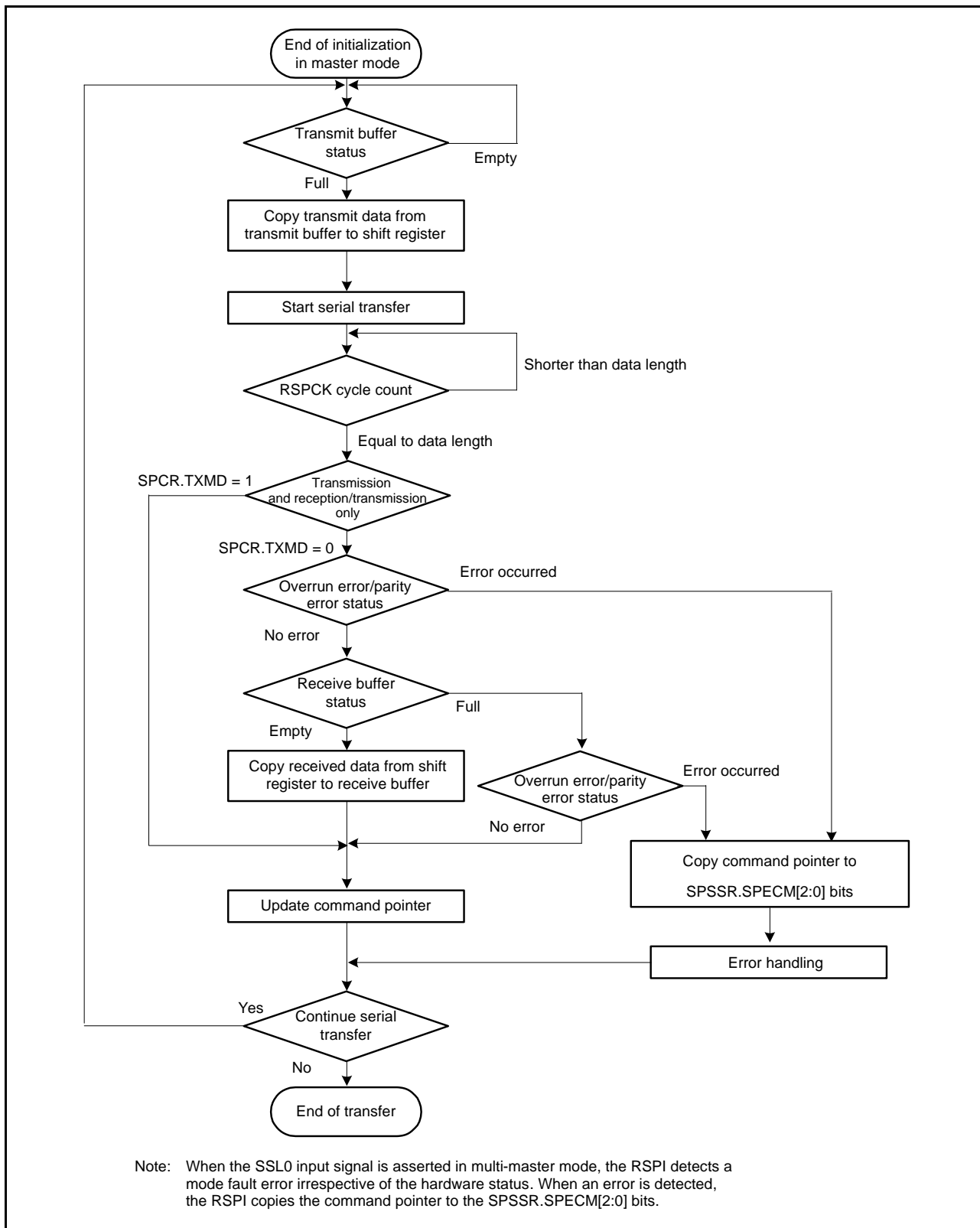


Figure 33.25 Transfer Operation Flowchart in Master Mode (SPI Operation)

33.3.10.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSL0 input signal assertion, the RSPI needs to start driving valid data to the MISO output signal. For this reason, when the CPHA bit is 0, the assertion of the SSL0 input signal triggers the start of a serial transfer.

If the CPHA bit is 1, when detecting the first RSPCK edge in an SSL0 signal asserted condition, the RSPI needs to start driving valid data to the MISO output signal. For this reason, when the CPHA bit is 1, the first RSPCK edge in an SSL0 signal asserted condition triggers the start of a serial transfer.

When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to "full", so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI leaves the status of the shift register unchanged, in the full state.

Irrespective of CPHA bit setting, the timing at which the RSPI starts driving of the MISO output signal is the SSL0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on the CPHA bit setting.

For details on the RSPI transfer format, see section 33.3.4, Transfer Format. The polarity of the SSL0 input signal depends on the setting of the SSL0P bit in the RSPI slave select polarity register (SSLP).

(2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCK edge corresponding to the final sampling timing. When free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR). Upon termination of a serial transfer the RSPI changes the status of the shift register to "empty". A mode fault error occurs if the RSPI detects an SSL0 input signal negation from the beginning of serial transfer to the end of serial transfer (see section 33.3.8, 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 SSL0 input signal depends on the SSLP.SSL0P bit setting.

For details on the RSPI transfer format, see section 33.3.4, 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 SSL0 input signal. In the type of configuration shown in Figure 33.5 as an example, if the RSPI is used in single-slave mode, the SSL0 signal is always fixed at the active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute transmit/receive operations by the RSPI in a configuration in which the SSL0 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 SSL0 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 SSL0 input signal. If the CPHA bit is 1, the period from the first RSPCK edge to the sampling timing for the reception of the final bit in an SSL0 signal active state corresponds to a serial transfer period. Even when the SSL0 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, second and subsequent serial transfers during burst transfer cannot be executed correctly.

(5) Initialization Flowchart

Figure 33.26 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, DMACA, and input/output ports, see the descriptions given in the individual blocks.

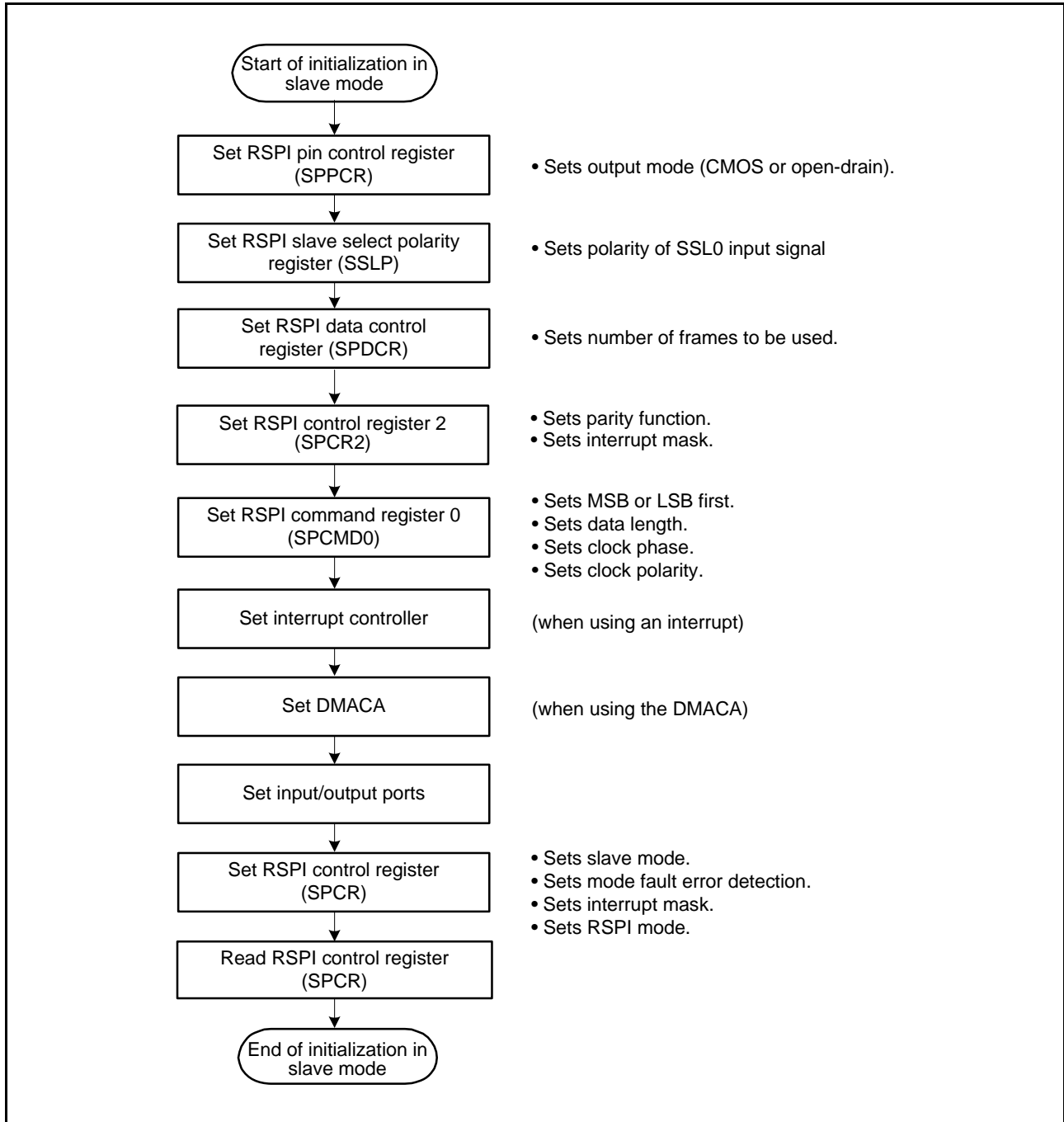


Figure 33.26 Example of Initialization Flowchart in Slave Mode (SPI Operation)

(6) Transfer Operation Flowchart (CPHA = 0)

Figure 33.27 is a flowchart illustrating a transfer in SPI operation when the RSPI is used in slave mode with the SPCMD0.CPHA bit set to 0.

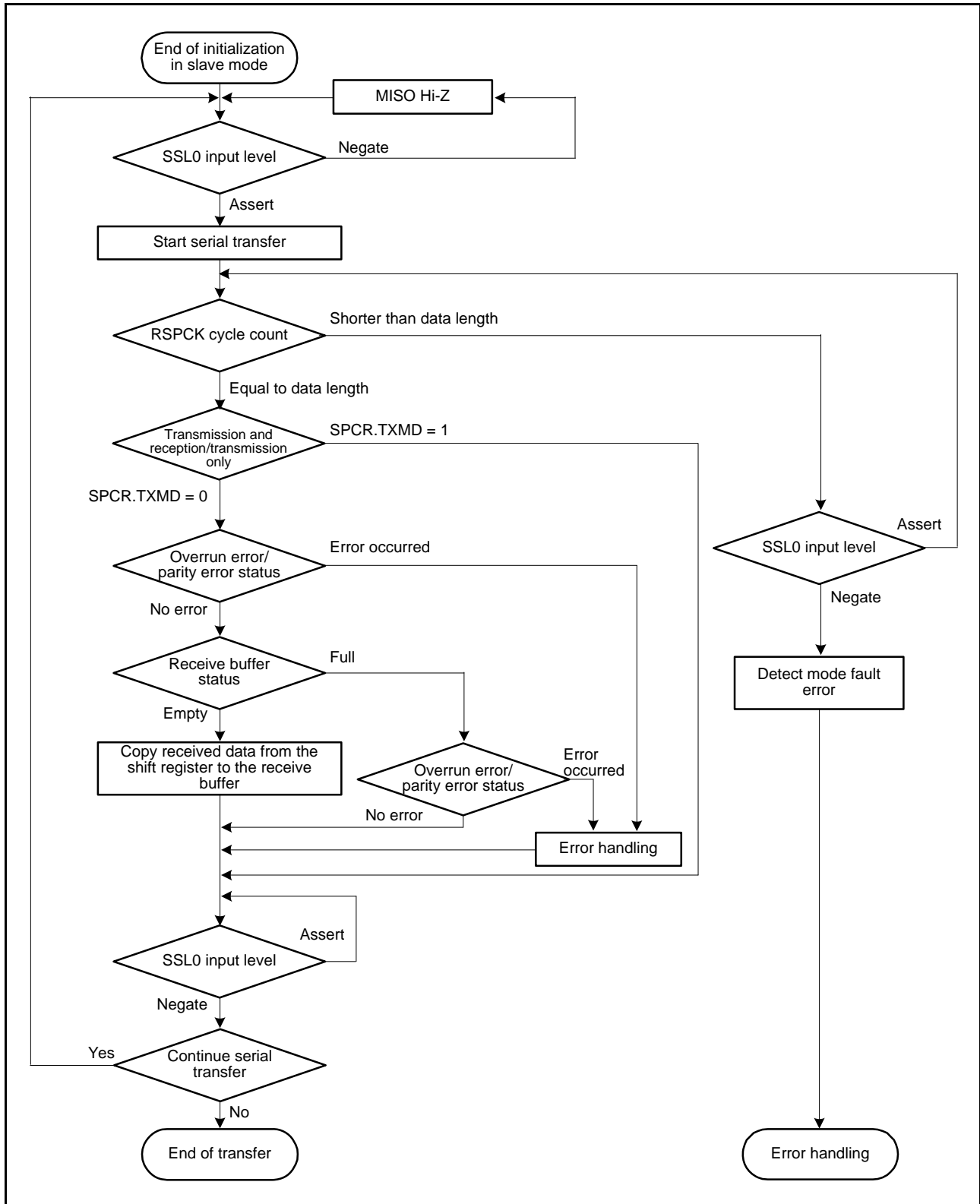


Figure 33.27 Transfer Operation Flowchart in Slave Mode (CPHA = 0) (SPI Operation)

(7) Transfer Operation Flowchart (CPHA = 1)

Figure 33.28 is a flowchart illustrating a transfer in SPI operation when the RSPI is used in slave mode with both the SPCMD0.CPHA bit and SPCR.MODFEN bit set to 1. The subsequent operations are not guaranteed when the serial transfer is started with the MODFEN bit set to 0 and the SSL0 input level is negated with a number of RSPCK cycles shorter than the data length.

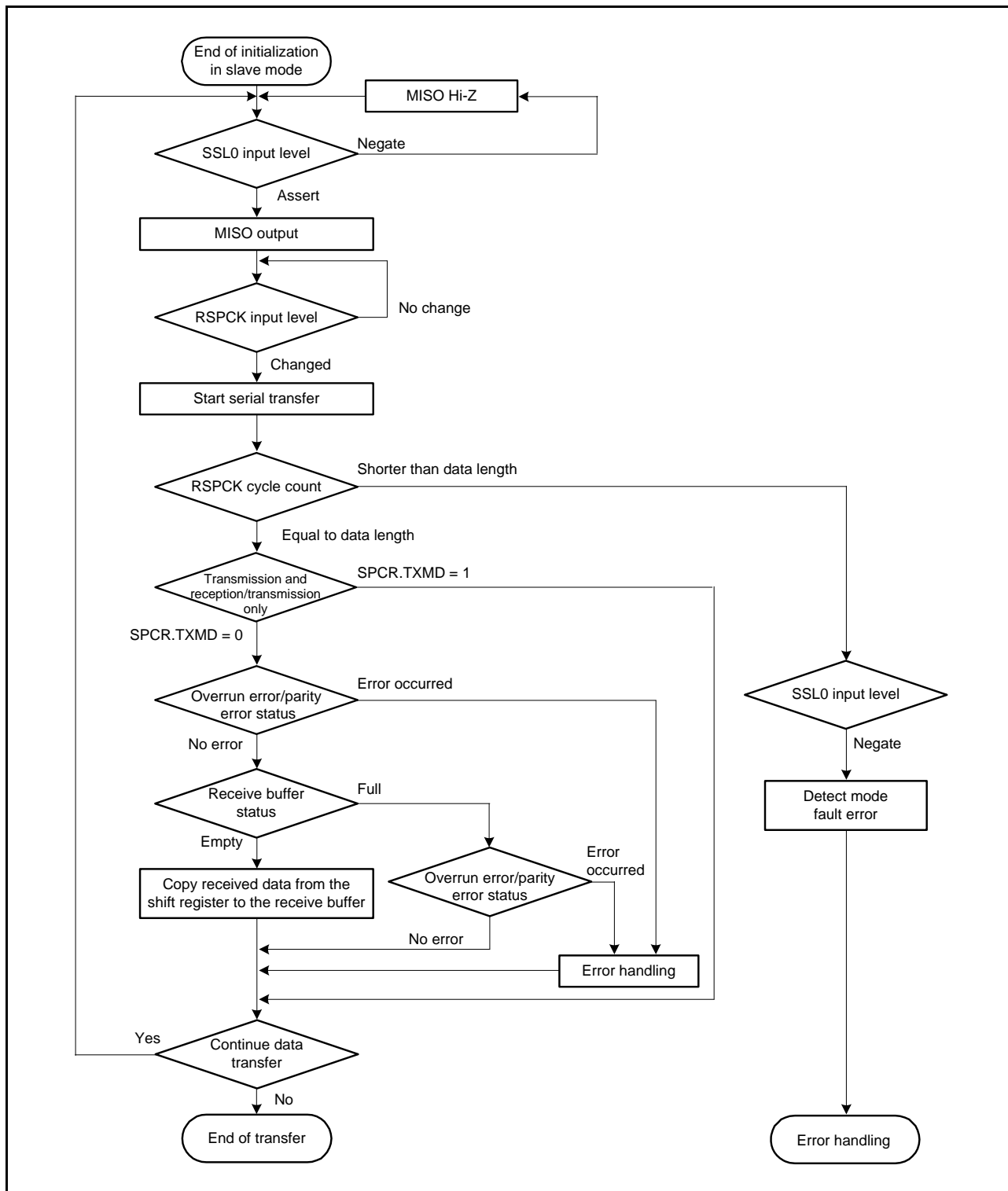


Figure 33.28 Transfer Operation Flowchart in Slave Mode (CPHA = 1) (SPI Operation)

33.3.11 Clock Synchronous Operation

Setting the SPMS bit in the RSPI control register (SPCR) to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLn pin is not used, and the three pins of RSPCKn, MOSIn, and MISO_n handle communications. The SSLn pin is available as I/O port pins. (n = A, B)

Although clock synchronous operation does not require use of the SSLn 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 SSLn pin is not used.

Furthermore, operation is not guaranteed if clock synchronous operation proceeds when the SPCMD.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

33.3.12 Master Mode Operation

(1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer 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). If the shift register is empty due to the writing of 0 either after the writing to SPDR, the RSPI copies the data in the transmit buffer to the shift register and starts a 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 from the CPU.

For details on the RSPI transfer format, see section 33.3.4, Transfer Format.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCK 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 SPCMD.SPB[3:0] bit setting.

For details on the RSPI transfer format, see section 33.3.4, Transfer Format.

(3) Sequence Control

The transfer format employed in master mode is determined by SPSCR, SPCMD, SPBR, SPCKD, SSLND, and SPND. Although the SSL 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 RSPI command registers SPCMD0 to SPCMD7: SSL 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 SPCMD0 to SPCMD7. The RSPI contains a pointer to the SPCMD that makes up the sequence. The CPU can check the value of this pointer by reading the SPSSR.SPSP[2:0] bits. When the SPE bit in the RSPI control register (SPCR) is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0, 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.

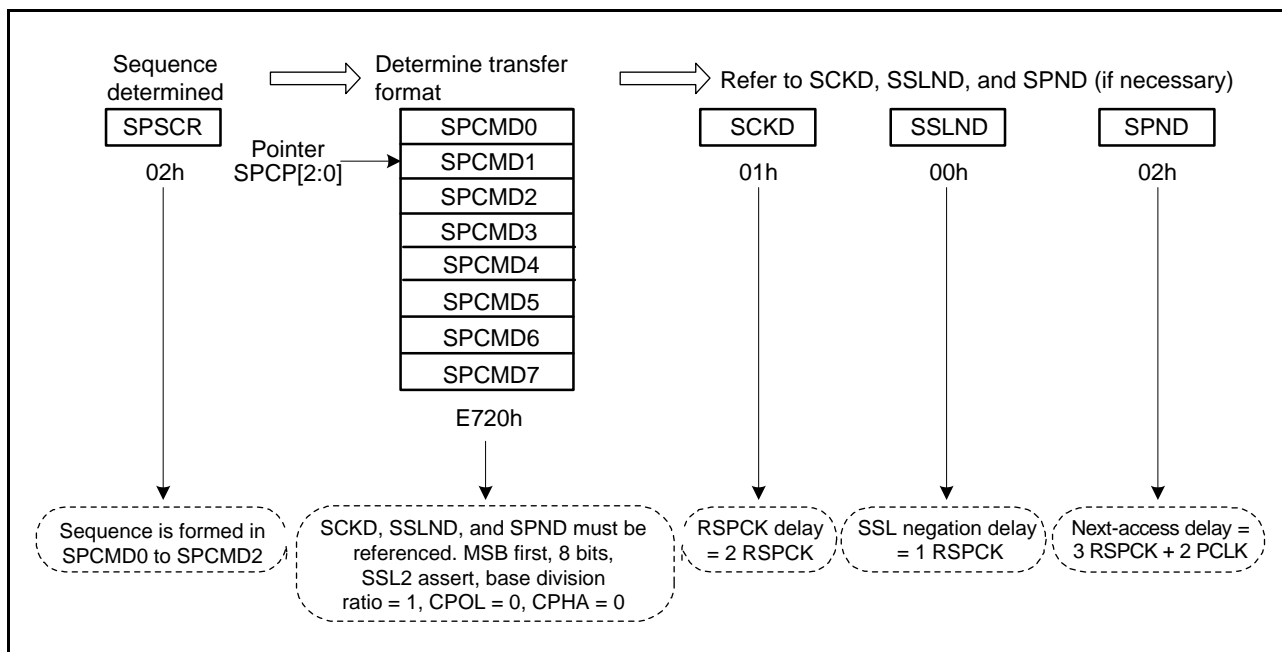


Figure 33.29 Determination Procedure of Serial Transfer Mode in Master Mode (Clock Synchronous Operation)

(4) Initialization Flowchart

Figure 33.30 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, DMACA, and input/output ports, see the descriptions given in the individual blocks.

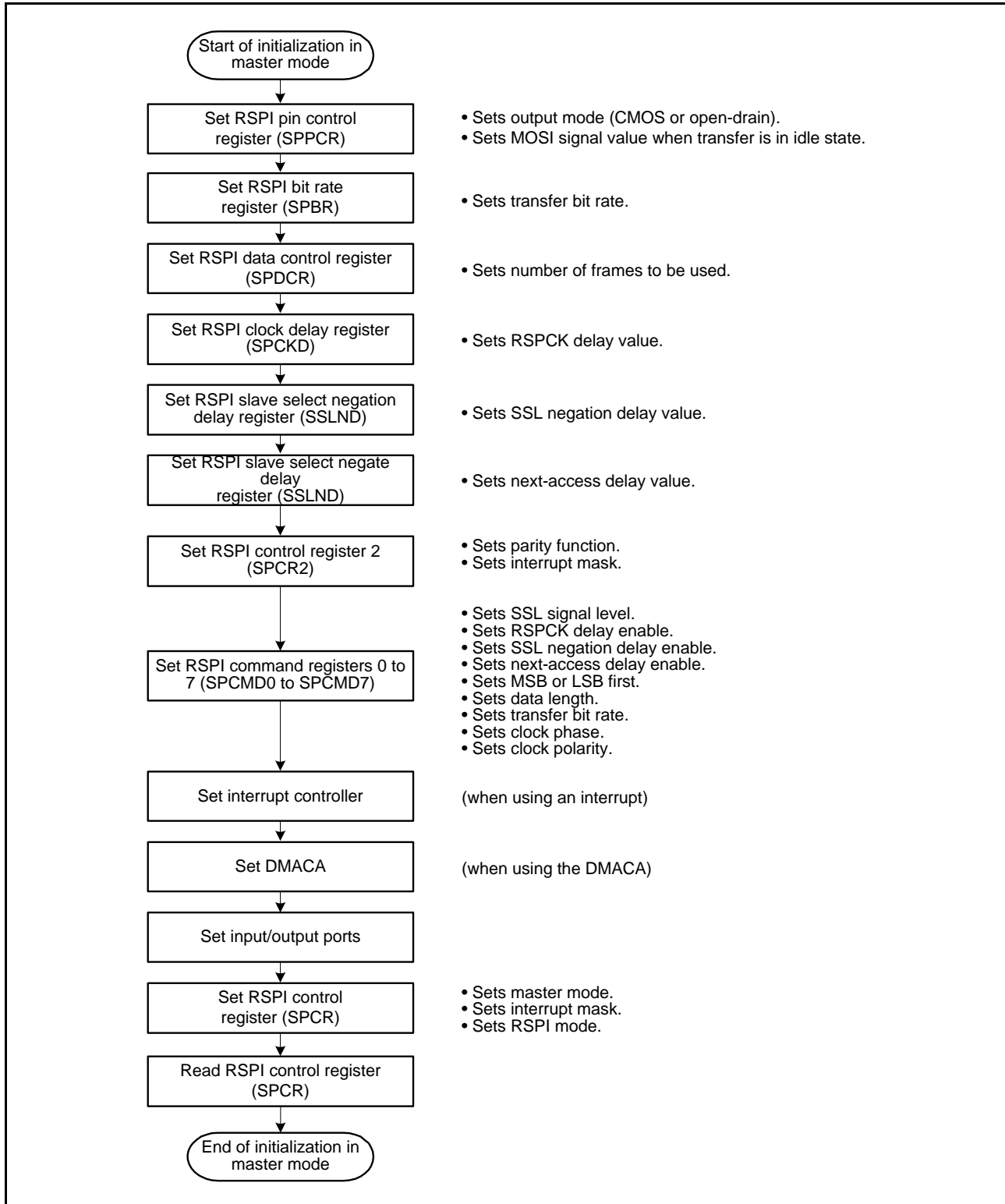


Figure 33.30 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(5) Transfer Operation Flowchart

Figure 33.31 is a flowchart illustrating a transfer in clock synchronous operation when the RSPI is used in master mode.

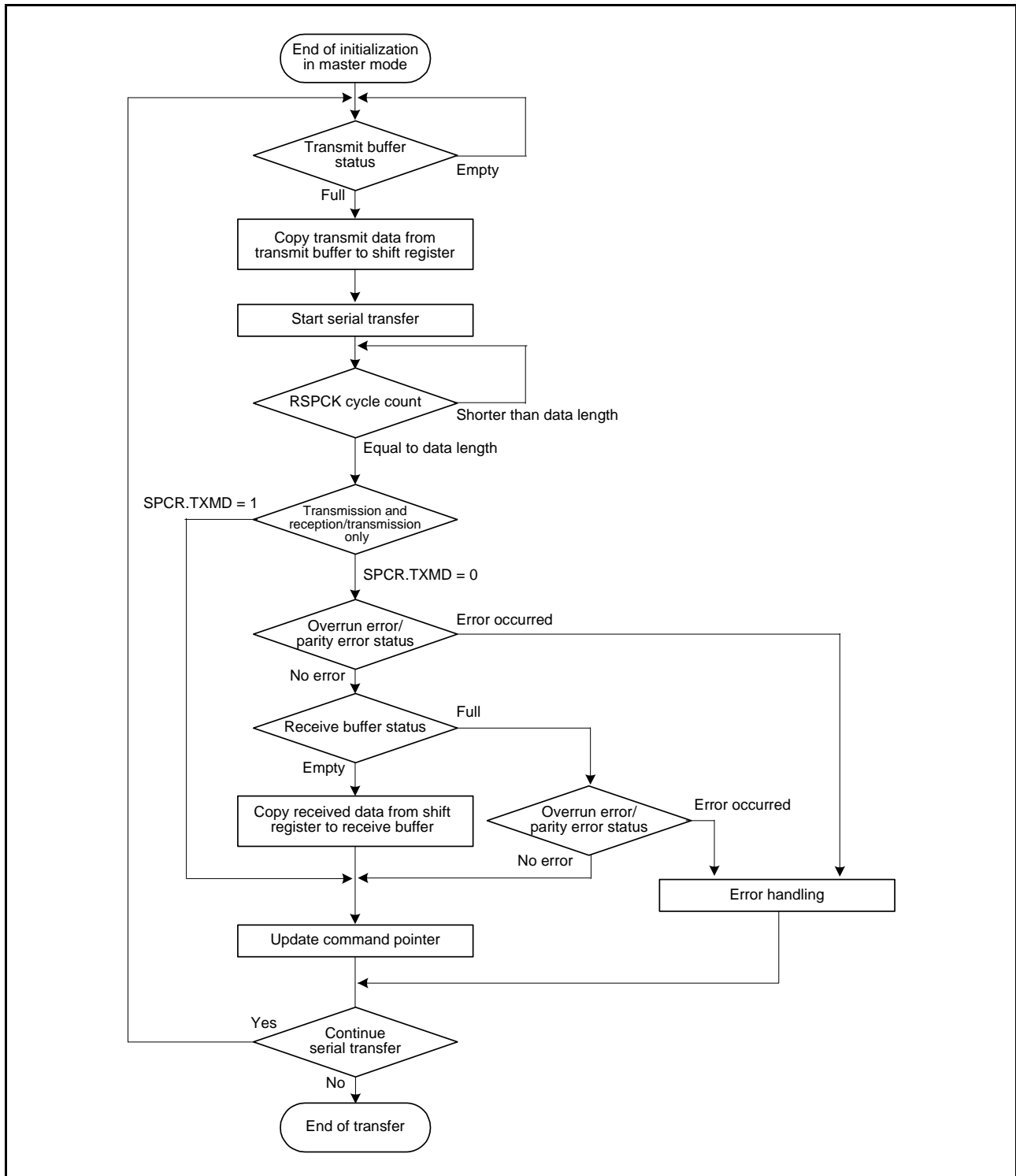


Figure 33.31 Transfer Operation Flowchart in Master Mode (Clock Synchronous Operation)

33.3.13 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCK edge triggers the start of a serial transfer in the RSPI.

When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to "full", so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI keeps the status of the shift register unchanged, in the full state.

When the SPMS bit is 1, the RSPI always drives the MISO output signal.

For details on the RSPI transfer format, see section 33.3.4, 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 RSPCK edge corresponding to the final sampling timing. When free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR). Upon termination of a serial transfer the RSPI changes the status of the shift register to "empty". The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting.

For details on the RSPI transfer format, see section 33.3.4, Transfer Format.

(3) Initialization Flowchart

Figure 33.32 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, DMACA, and input/output ports, see the descriptions given in the individual blocks.

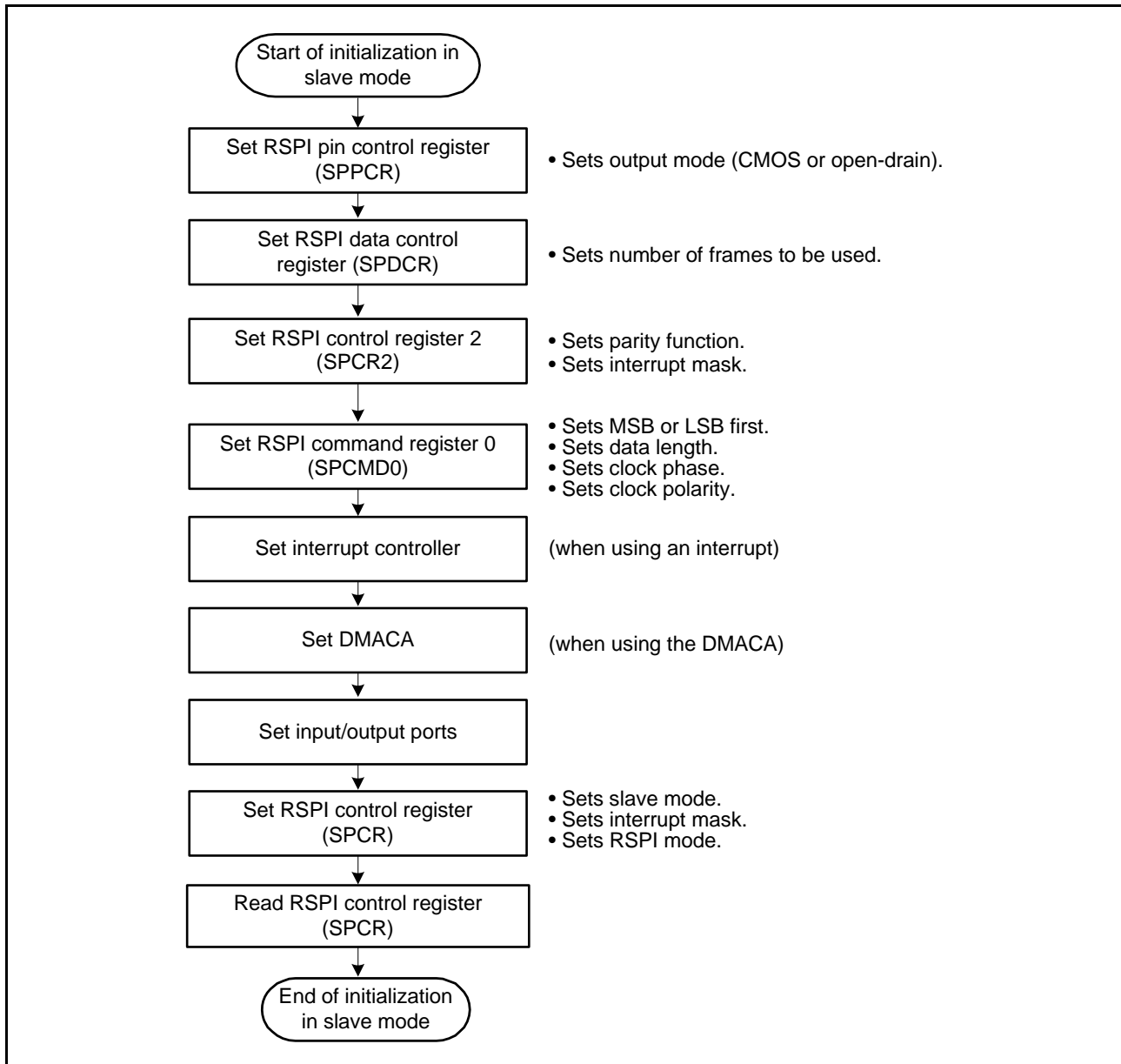


Figure 33.32 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(4) Transfer Operation Flowcharts

Figure 33.33 is a flowchart illustrating a transfer when the RSPi is in clock synchronous operation.

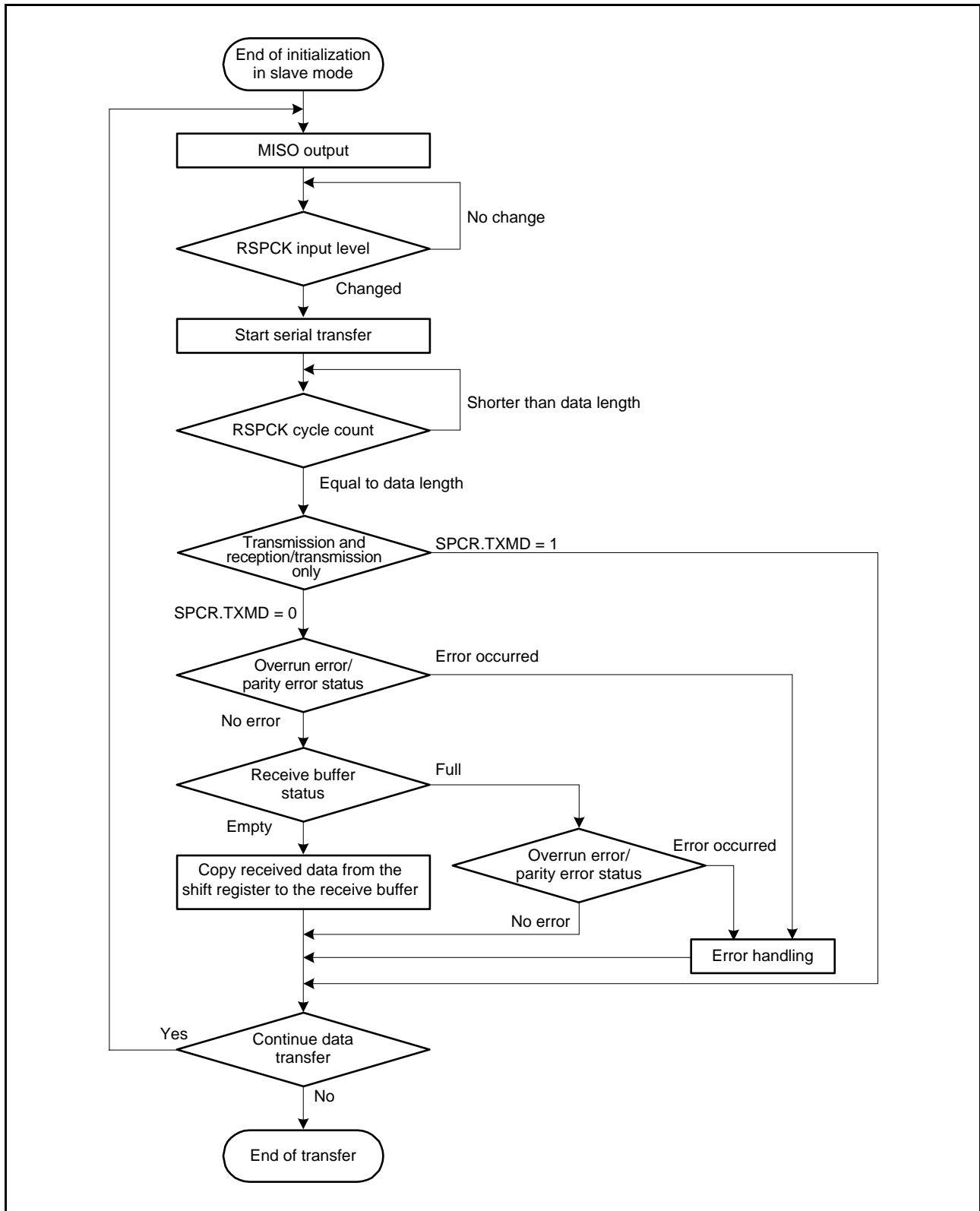


Figure 33.33 Transfer Operation Flowchart in Slave Mode (CPHA = 1) (Clock Synchronous Operation)

33.3.14 Error Handling

Figure 33.34 to Figure 33.36 show error handling for the RSPI. The following error handling is used to return from the error state after an error has occurred in master mode or slave mode.

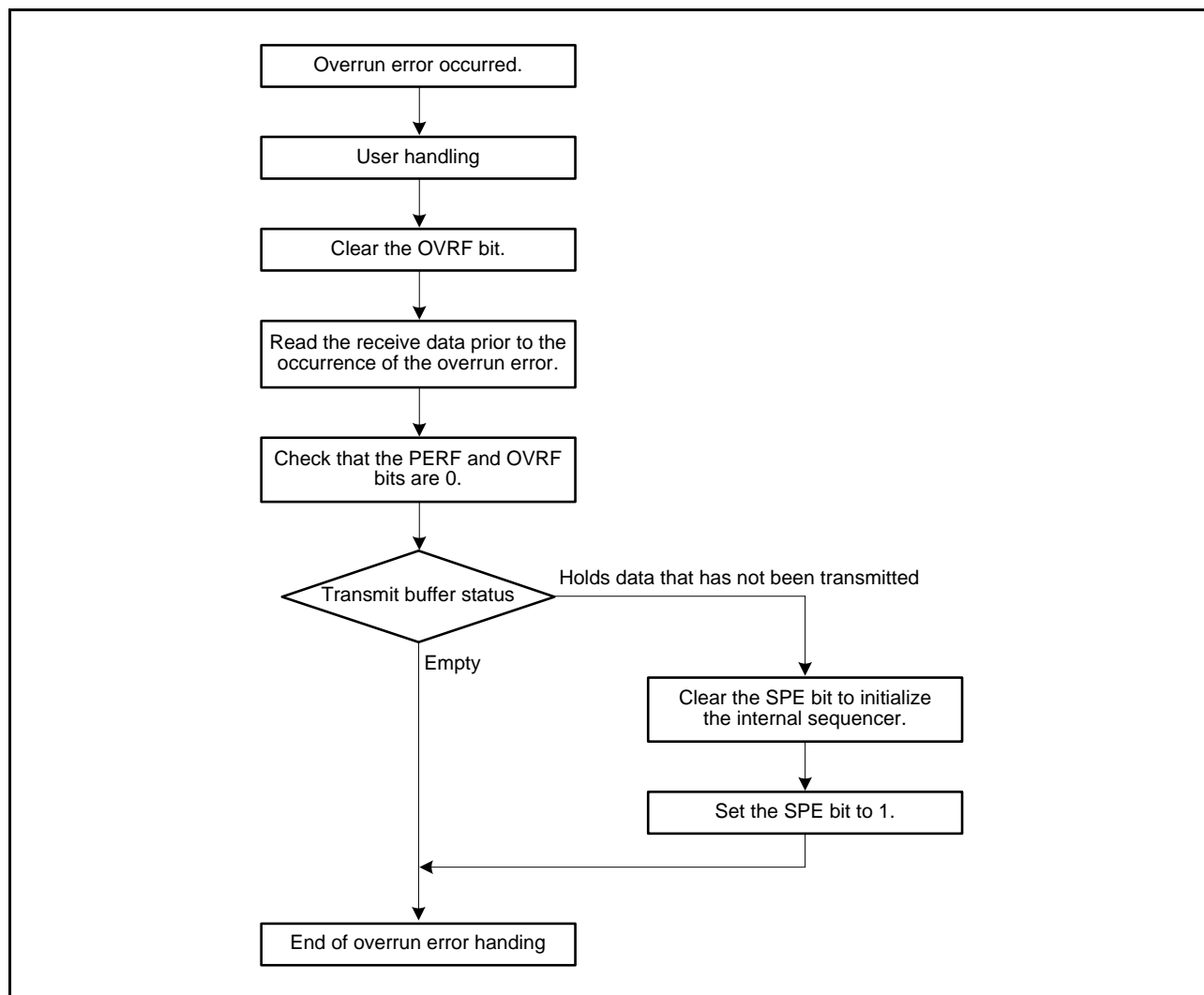


Figure 33.34 Error Handling (Overrun Error)

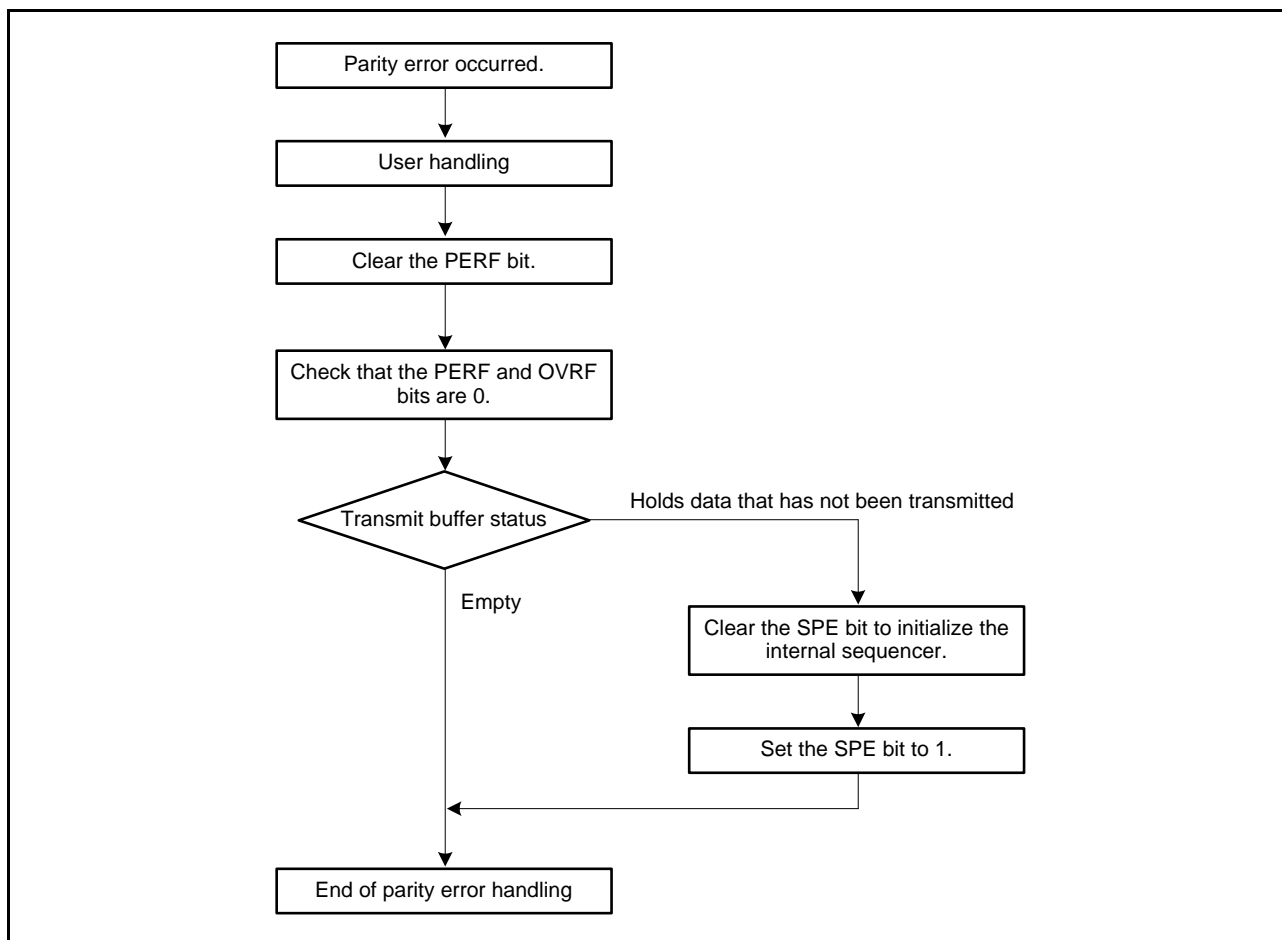


Figure 33.35 Error Handling (Parity Error)

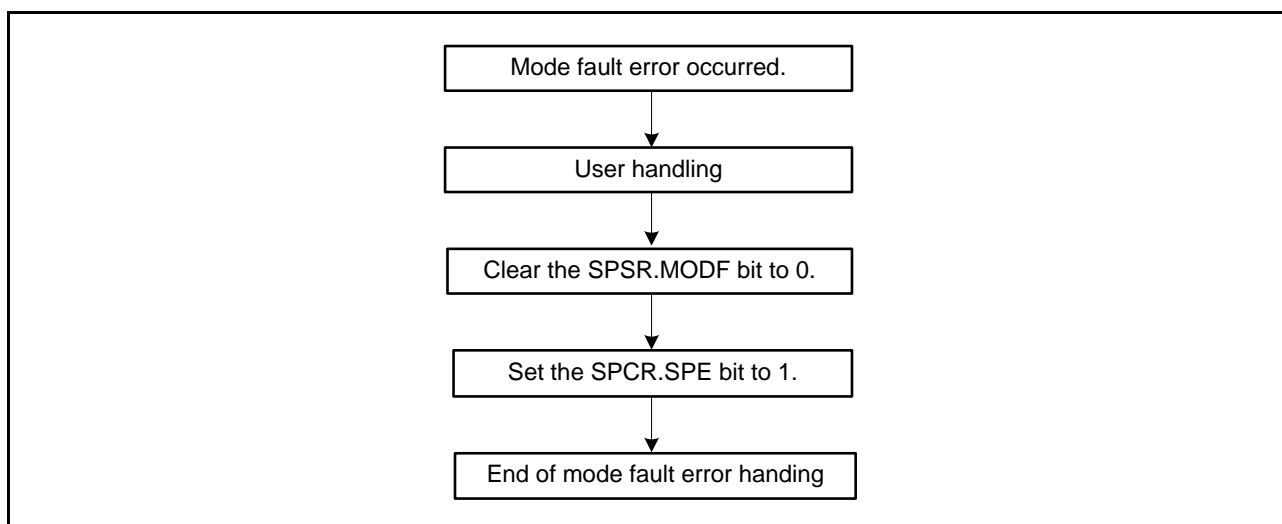


Figure 33.36 Error Handling (Mode Fault Error)

33.3.15 Loopback Mode

When the CPU writes 1 to the SPPCR.SPLP2 bit or SPPCR.SPLP bit in loopback mode, the RSPI shuts off the path between the MISO pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path of the shift register. The RSPI does not shut off the path between the MOSI pin and the shift register if the SPCR.MSTR bit is 1, and between the MISO pin and the shift register if the SPCR.MSTR bit is 0.

When a serial transfer is executed in loopback mode, the transmit data for the RSPI becomes the received data for the RSPI. (n = A, B)

Table 33.13 shows the relationship among the SPLP2 and SPLP bits in SPPCR and the received data. Figure 33.37 shows the configuration of the shift register input/output paths for the case where the RSPI in master mode is set in loopback mode (SPPCR.SPLP2 = 0, SPPCR.SPLP = 1).

Table 33.13 SPLP2 and SPLP Bit Settings and Received Data

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSI pin or MISO pin (n = A, B)
0	1	Reversed transmit data
1	0	Transmit data
1	1	Transmit data

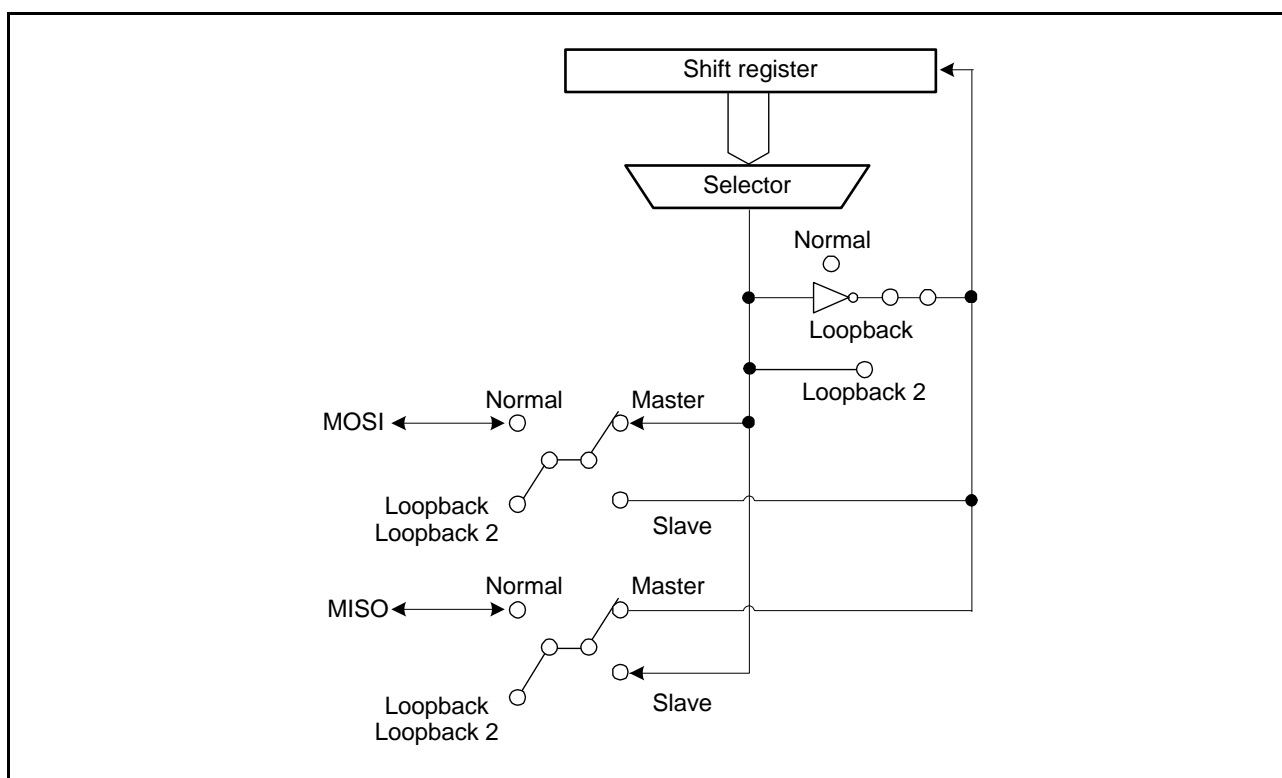


Figure 33.37 Configuration of Shift Register Input/Output Paths in Loopback Mode (Master Mode)

33.3.16 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 33.38.

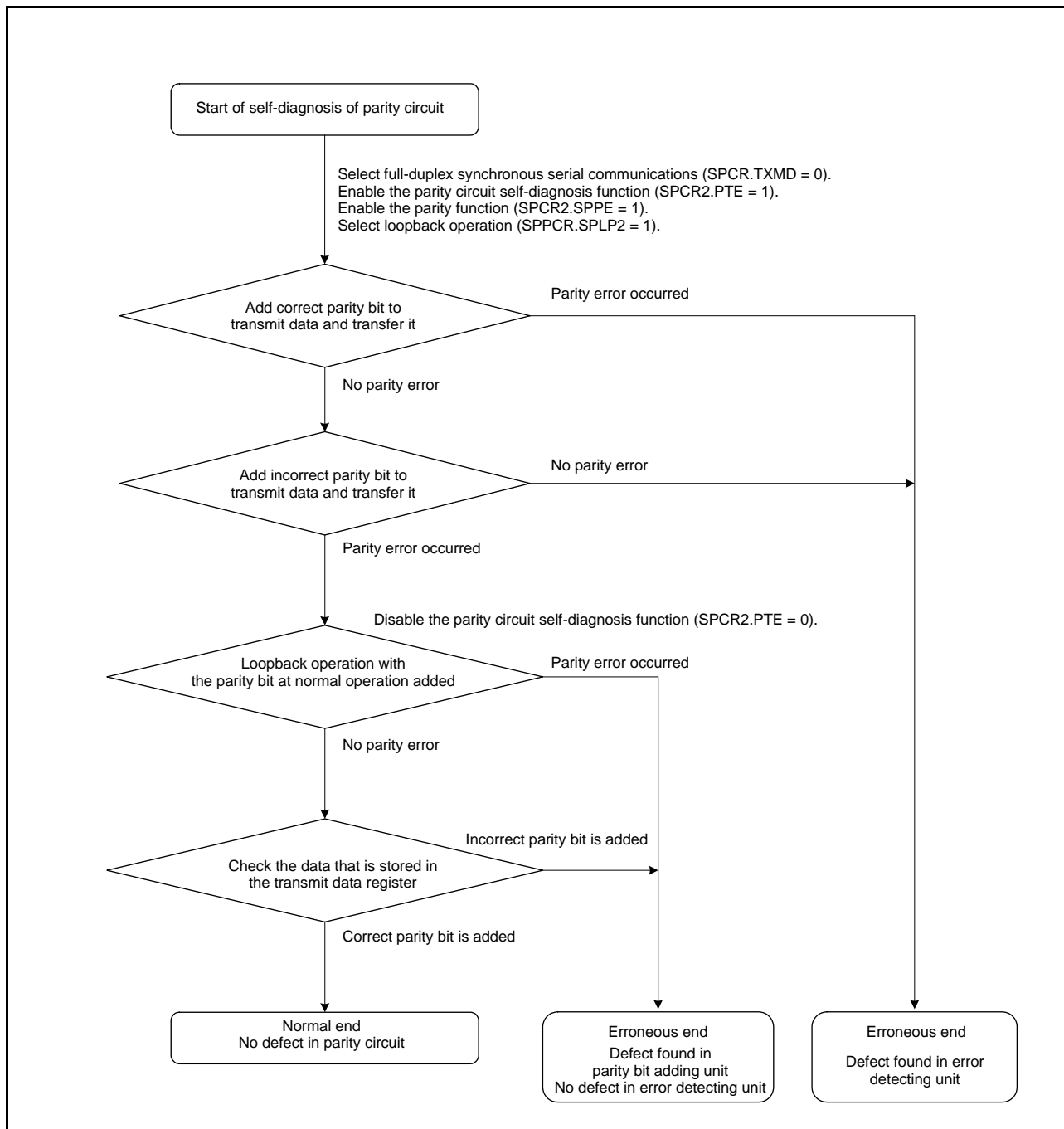


Figure 33.38 Flowchart for Self-Diagnosis of Parity Circuit

33.3.17 Interrupt Sources

The RSPI has interrupt sources of receive buffer full, transmit buffer empty, mode fault, and RSPI idle. In addition, the DTC or DMACA can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Table 33.14 shows the RSPI interrupt sources. When any of the interrupt conditions in Table 33.14 is met, an interrupt is generated. Clear the interrupt sources by data transfer.

When using the DTC or DMACA to perform data transmission/reception, the DTC or DMACA must be set up first to be in a status in which transfer is enabled before making the RSPI settings. For the method for setting the DTC or DMACA, refer to section 14, DMA Controller (DMACA), or section 16, Data Transfer Controller (DTCa).

Table 33.14 Interrupt Sources of RSPI

Interrupt Source	Abbreviation	Interrupt Condition	DMACA/DTC Activation
Receive buffer full	SPRI	(sprie = 1) • (Receive buffer full)	○
Transmit buffer empty	SPTI	(sptie = 1) • (Transmit buffer empty)	○
Mode fault Overrun Parity error	SPEI	(speie = 1) • {(modf = 1) (ovrf = 1) (perf = 1)}	—
RSPI idle	SPII	(spiie = 1) • (idlnf = 0)	—

33.4 Usage Note

33.4.1 Transmit Operation when Parity Function is Enabled in Master Mode

To perform transmit operation with the parity bit added when the parity function is enabled in master mode, the following settings of each command register should be set to the same value.

Transfer bit length setting of each command register

MSB-first/LSB-first setting of each command register

34. 12-Bit A/D Converter (S12AD)

34.1 Overview

The RX62N/RX621 Group includes a 12-bit successive approximation A/D converter. Up to 8-channel analog inputs can be selected.

The A/D converter has two operating modes: single-cycle scan mode in which the analog input of up to eight channels is converted for only once in ascending channel order and continuous scan mode in which the analog inputs of up to eight arbitrarily selected channels are continuously converted in ascending channel order.

Table 34.1 lists the specifications of the A/D converter and Table 34.2 indicates the functions of the A/D converter. Figure 34.1 shows a block diagram of the A/D converter.

Note that either the 12-bit A/D converter or the 10-bit A/D converter should be exclusively selected by the MSTPA23, MSTPA22, and MSTPA17 bits in module stop control register A (MSTPCRA).

Table 34.1 Specifications of A/D Converter

Item	Specifications
Number of units	One unit (S12AD0)
Input channels	8 channels
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	1.0 μ s per 1 channel (when operating with peripheral module clock PCLK = 50 MHz)
A/D conversion clock	4 types: PCLK, PCLK/2, PCLK/4, PCLK/8
Data registers	Eight data registers The A/D conversion result is held in a 12-bit A/D data register. In A/D-converted value addition mode, 14 bits of data are stored in an A/D data register.
Operating modes	<ul style="list-style-type: none"> Single-cycle scan mode: A/D conversion is to be performed for only once on the analog inputs of up to eight arbitrarily selected channels. Continuous scan mode: A/D conversion is to be performed sequentially on the analog inputs of up to eight arbitrarily selected channels.
Conditions of A/D conversion start	<ul style="list-style-type: none"> Software trigger Conversion start trigger by the multifunction timer pulse unit (MTU) or 8-bit timer (TMR). External trigger A/D conversion can be triggered from the ADTRG0# pin.
Function	<ul style="list-style-type: none"> Sample-and-hold function A/D-converted value addition mode
Interrupt source	<ul style="list-style-type: none"> A/D conversion end interrupt (S12ADI0) request can be generated on completion of A/D conversion. An S12ADI0 interrupt can activate DMA controller (DMACA) or data transfer controller (DTC).
Power-down function	Module stop state can be specified.*

Note: * When the module stop state is canceled, A/D conversion can be started after 10 ms has elapsed.

Table 34.2 Functions of A/D Converter

Item			Function/Internal Trigger Source		
Analog input channel			AN0 to AN7		
A/D conversion start conditions	Software	Software trigger	Enabled		
	External trigger	ADTRG0#	Enabled		
	Internal trigger (MTU2, TMR)*3	TRG0AN_0	MTU0.TGRA and MTU0.TCNT	Input capture/compare match	
			TRG0BN_0		
		TRGAN_0	MTU0.TGRA and MTU0.TCNT	Input capture/compare match	
			MTU1.TGRA and MTU1.TCNT		
			MTU2.TGRA and MTU2.TCNT		
			MTU3.TGRA and MTU3.TCNT		
			MTU4.TGRA and MTU4.TCNT		
			MTU4.TCNT		
		TRGAN_1	MTU6.TGRA and MTU6.TCNT	Input capture/compare match	
			MTU7.TGRA and MTU7.TCNT		
			MTU8.TGRA and MTU8.TCNT		
			MTU9.TGRA and MTU9.TCNT		
			MTU10.TGRA and MTU10.TCNT		
		MTU10.TCNT	TCNT underflow (trough) in complementary PWM mode		
TRG0EN_0	MTU0.TGRE and MTU0.TCNT	compare match			
TRG0FN_0	MTU0.TGRF and MTU0.TCNT	compare match			
TRG4ABN_0	MTU4.TADCORA and MTU4.TCNT or MTU4.TADCORB and MTU4.TCNT	Compare match with the A/D converter start request delaying function			
TRG4ABN_1	MTU10.TADCORA and MTU10.TCNT or MTU10.TADCORB and MTU10.TCNT				
TMTRG0AN_0	TMR0.TCORA and TMR0.TCNT	compare match			
TMTRG0AN_1	TMR2.TCORA and TMR2.TCNT	compare match			
Interrupt			S12ADIO interrupt (S12ADIO)		
Module stop function setting*1*2			MSTPCRA.MSTPA17 bit		

Note 1. When the module stop state is canceled, A/D conversion can be started after 10 ms has elapsed.

Note 2. For details, see section 9, Low Power Consumption.

Note 3. _0 and _1 suffixed to internal trigger names indicate a unit number. For internal trigger output setting, see section 17.4.3, A/D Converter Activation and section 21.6.2, A/D Converter Activation.

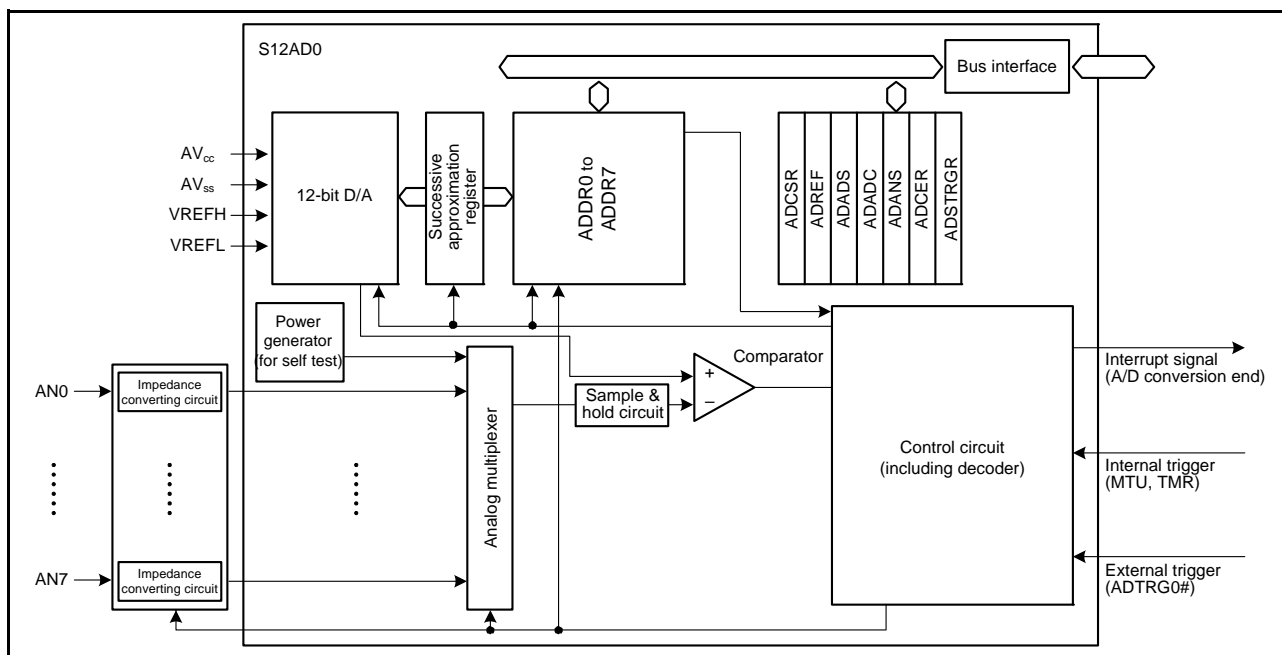


Figure 34.1 Block Diagram of A/D Converter

Table 34.3 indicates the input pins of the A/D converter.

Table 34.3 Input Pins of A/D Converter

Pin Name	Input	Function
AVcc	Input	Analog block power supply pin
AVss	Input	Analog block ground pin
VREFH	Input	Reference power supply pin
VREFL	Input	Reference power supply ground pin
AN0 to AN7	Input	Analog input pins
ADTRG0#	Input	External trigger input pin for starting A/D conversion

34.2 Register Descriptions

Table 34.4 lists the registers of the A/D converter.

Table 34.4 Registers of A/D Converter

Module Symbol	Register Name	Register Symbol	Value after Reset	Address	Access Size
S12AD0	A/D data register 0	ADDR0	0000h	0008 9020h	16
	A/D data register 1	ADDR1	0000h	0008 9022h	16
	A/D data register 2	ADDR2	0000h	0008 9024h	16
	A/D data register 3	ADDR3	0000h	0008 9026h	16
	A/D data register 4	ADDR4	0000h	0008 9028h	16
	A/D data register 5	ADDR5	0000h	0008 902Ah	16
	A/D data register 6	ADDR6	0000h	0008 902Ch	16
	A/D data register 7	ADDR7	0000h	0008 902Eh	16
	A/D control register	ADCSR	00h	0008 9000h	8
	A/D channel select register	ADANS	0000h	0008 9004h	16
	A/D-converted value addition mode select register	ADADS	0000h	0008 9008h	16
	A/D-converted value addition count select register	ADADC	00h	0008 900Ch	8
	A/D control extended register	ADCER	0000h	0008 900Eh	16
	A/D start trigger select register	ADSTRGR	00h	0008 9010h	8

34.2.1 A/D Data Registers n (ADDRn) (n = 0 to 7)

A/D data registers 0 to 7 (ADDR0 to ADDR7) are 16-bit read-only registers which store the A/D conversion results of channels AN0 to AN7.

ADDRn uses different formats under the following conditions.

- Setting of A/D data register format select bit (right-alignment or left-alignment)
- Setting of A/D-converted value addition mode select bits (not selected or selected)

The following minimum and maximum values apply to channels on which A/D-converted value addition mode is selected. When A/D-converted value addition mode is selected, the setting of the A/D data register format select bit (ADCER.ADRFMT) becomes invalid.

First conversion: $0000h \leq ADDRn (n = 0 \text{ to } 7) \leq 3FFCh$

(ADDRn (n = 0 to 7): Bits 15 and 14 = 00b, bits 13 to 2 = AD11 to AD0, bits 1 and 0 = 00b)

Second conversion: $0000h \leq ADDRn (n = 0 \text{ to } 7) \leq 7FF8h$

(ADDRn (n = 0 to 7): Bit 15 = 0b, bits 14 to 2 = AD12 to AD0, bits 1 and 0 = 00b)

Third conversion: $0000h \leq ADDRn (n = 0 \text{ to } 7) \leq BFF4h$

(ADDRn (n = 0 to 7): Bits 15 to 2 = AD13 to AD0, bits 1 and 0 = 00b)

Fourth conversion: $0000h \leq ADDRn (n = 0 \text{ to } 7) \leq FFF0h$

(ADDRn (n = 0 to 7): Bits 15 to 2 = AD13 to AD0, bits 1 and 0 = 00b)

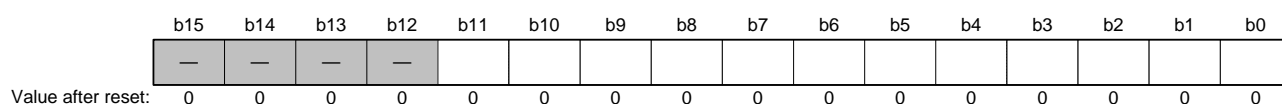
(1) Registers: ADDR0 to ADDR7

The ADRFMT bit in ADCER can be used to set either right- or left-alignment. Here, the AD11 to AD0 bits in ADDRn indicate a 12-bit A/D converted value. Other bits are reserved. These reserved bits are always read as 0. The write value should always be 0.

In A/D-converted value addition mode, setting the ADRFMT bit in ADCER is invalid and left-alignment is always set. Here, the AD13 to AD0 bits in ADDRn indicate the sum of all the converted values in A/D-converted value addition mode.

- When right-alignment is selected

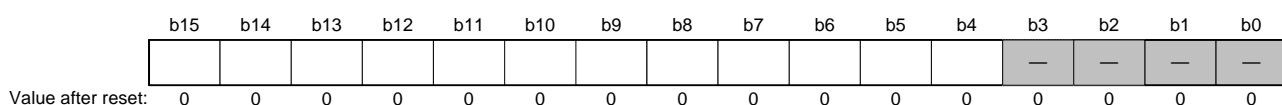
Addresses: 0008 9020h to 0008 902Eh



Bit	Symbol	Bit Name	Description	R/W
b11 to b0	—	—	12-bit A/D-converted value	R
b15 to b12	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

- When left-alignment is selected

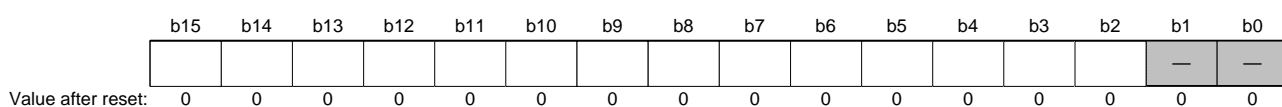
Addresses: 0008 9020h to 0008 902Eh



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b15 to b4	—	—	12-bit A/D-converted value	R

- When A/D-converted value addition mode is selected

Addresses: 0008 9020h to 0008 902Eh



Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b15 to b2	—	—	14-bit A/D-converted value addition result	R

When A/D-converted value addition mode is selected, the setting of the A/D data register format select bit (ADRFMT) in ADCER becomes invalid.

34.2.2 A/D Control Register (ADCSR)

Address: 0008 9000h

	b7	b6	b5	b4	b3	b2	b1	b0
	ADST	ADCS	—	ADIE	CKS[1:0]	TRGE	EXTRG	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EXTRG	Trigger Select	0: Scan conversion is started by a timer source selected by the A/D start trigger select register (ADSTRGR). 1: Scan conversion is started by an external trigger (ADTRG0#).	R/W
b1	TRGE	Trigger Enable	0: Disables scan conversion to be started by an external trigger (ADTRG0#) or a trigger of MTU or TMR. 1: Enables scan conversion to be started by an external trigger (ADTRG0#) or a trigger of MTU or TMR.	R/W
b2, b3	CKS[1:0]	Clock Select	b3 b2 00: PCLK/8 01: PCLK/4 10: PCLK/2 11: PCLK	R/W
b4	ADIE	A/D Scan Conversion End Interrupt Enable	0: Disables S12ADI0 interrupt generation upon scan conversion completion. 1: Enables S12ADI0 interrupt generation upon scan conversion completion.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b6	ADCS	Scan Conversion Mode Select	0: Single-cycle scan mode 1: Continuous scan mode	R/W
b7	ADST	A/D Conversion Start	0: Stops a scan conversion process. 1: Starts a scan conversion process.	R/W

Note : Starting a scan conversion using an external trigger
If 1 is written to both the TRGE and EXTRG bits in ADCSR when a high signal is input to the external trigger pin (ADTRG0#), and then if the ADTRG0# signal is driven low, the falling edge of ADTRG0# is detected and the scan conversion process started. In this case, the pulse width of the low-input must be at least 1.5 PCLK clock cycles.

ADCSR selects the clock and performs the settings for scan conversion start/stop, scan mode and scan conversion trigger. The CKS[1:0] bits and ADCS bit should be set while the ADST bit is 0.

ADIE Bit (A/D Scan Conversion End Interrupt Enable)

The ADIE bit enables or disables the A/D scan conversion end interrupt (S12ADI0).

When scan conversion of the specified channels is completed while the ADIE bit is set to 1, an A/D scan conversion end interrupt (S12ADI0) is generated.

ADCS Bit (Scan Conversion Mode Select)

The ADCS bit selects the scan mode. In single-cycle scan mode, scan conversion of the analog inputs from a maximum of eight channels selected with the ADANS register is performed only once in the ascending order of the channel number, and when one cycle of scan conversion is completed for all the selected channels, scan conversion is stopped.

In continuous scan mode, when the ADST bit in ADCSR is set to 1, scan conversion of the analog inputs from a maximum of eight channels selected with the ADANS register is performed in the ascending order of the channel number, and when one cycle of scan conversion is completed for all the selected channels, scan conversion is repeated beginning from the first channel. Scan conversion is repeated for all the selected channels until the ADST bit is set to 0.

ADST Bit (A/D Conversion Start)

This bit starts or stops a scan conversion process. Before the ADST bit is set to 1, set the A/D conversion clock or operation mode.

[Setting conditions]

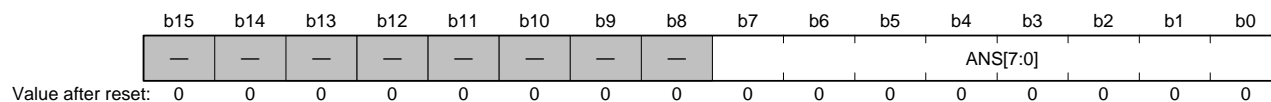
- When 1 is written by software
- When the ADCSR.TRGE bit is set to 1, and the MTU or TMR trigger selected by the ADSTRGR.ADSTRS[3:0] bits is detected
- When the ADCSR.TRGE and ADCSR.EXTRG bits are set to 1, the ADSTRGR.ADSTRS[3:0] bits are set to "0000b", and an external trigger is detected

[Clearing conditions]

- When 0 is written by software
- When the A/D conversion of all channels selected in single-cycle scan mode is completed

34.2.3 A/D Channel Select Register (ADANS)

Address: 0008 9004h

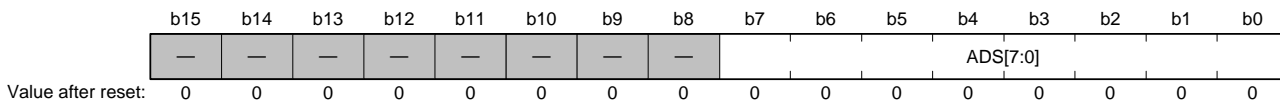


Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ANS[7:0]	A/D Conversion Channel Select	0: AN7 to AN0 are not subjected to scan conversion. 1: AN7 to AN0 are subjected to scan conversion.	R/W
b15 to b8	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

ADANS is used to select the channels ANn (n = 0 to 7) that are subjected to scan conversion. Note that the ADCSR.ADST bit must be cleared to 0 while the ADANS register values are changed.

34.2.4 A/D-Converted Value Addition Mode Select Register (ADADS)

Address: 0008 9008h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ADS[7:0]	A/D-Converted Value Addition Channel Select	0: A/D-converted value addition mode is not selected. 1: A/D-converted value addition mode is selected.	R/W
b15 to b8	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

ADADS selects the channels ANn (n = 0 to 7) on which A/D conversion is performed successively 2 to 4 times and then converted values are added (integrated). Note that the ADCSR.ADST bit must be cleared to 0 while the ADADS register values are changed.

ADS[7:0] Bits (A/D-Converted Value Addition Channel Select)

When the ADSn bit is set to 1, the A/D converter performs conversion on ANn successively 2 to 4 times and returns the added (integrated) conversion results to the A/D data register n (ADDRn). If the ADSn bit is cleared to 0, the A/D converter performs a normal 1-time conversion of ANn and returns the conversion result to the A/D data register n (ADDRn).

Figure 34.2 shows a scan conversion sequence in which both the ADS[2] and ADS[6] bits are set to 1, based on the assumption that the addition count is set to 4, and that channels AN0 to AN7 are selected. The conversion process begins with AN0. The AN2 conversion is performed successively 4 times, and the addition (integration) value is returned to the A/D data register n (ADDRn), after which the AN3 conversion process is started.

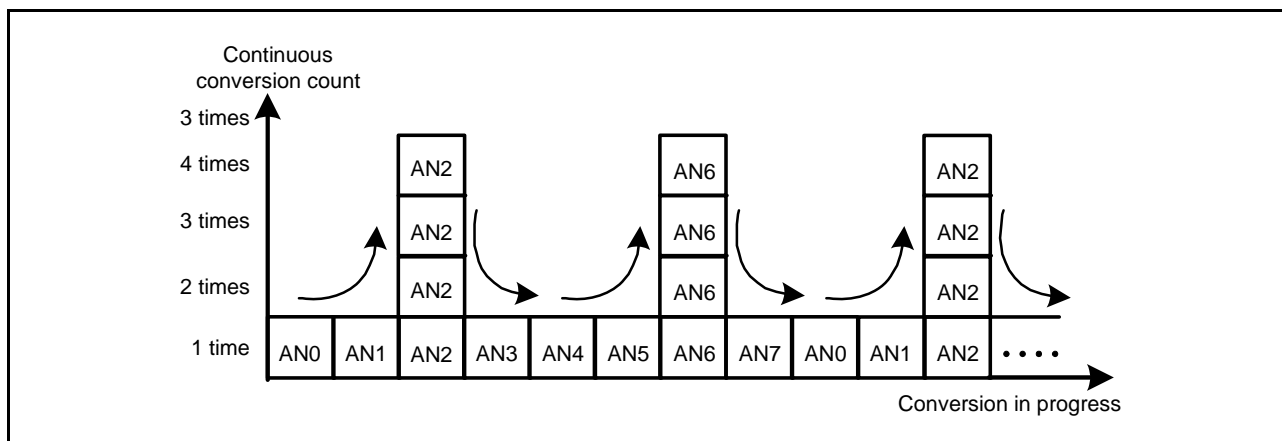
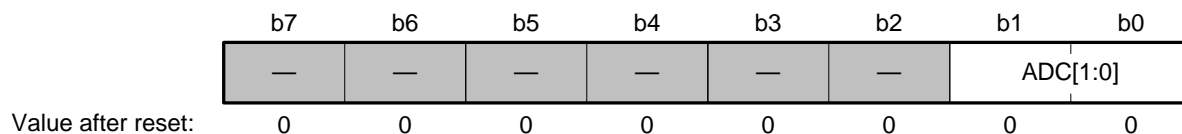


Figure 34.2 Scan Conversion Sequence with ADS[2] = 1 and ADS[6] = 1

For the channel for which the addition mode is not selected, the A/D data register n (ADDRn) format is determined by the ADCER.ADRFMT setting (right-alignment or left-alignment).

34.2.5 A/D-Converted Value Addition Count Select Register (ADADC)

Address: 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 always read as 0. The write value should always be 0.	R/W

ADADC sets the addition count for channels for which A/D-converted value addition mode is selected.

Set the ADADC register while the ADCSR.ADST bit is 0.

34.2.6 A/D Control Extended Register (ADCER)

Address: 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 always read as 0. The write value should always be 0.	R/W
b5	ACE	Automatic Clearing Enable	0: Disables automatic clearing of the A/D data register n (ADDRn) after it has been read. 1: Enables automatic clearing of the A/D data register n (ADDRn) after it has been read.	R/W
b14 to b6	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b15	ADRFMT	A/D Data Register Format Select	0: Right-alignment is selected for the A/D data register n (ADDRn) format. 1: Left-alignment is selected for the A/D data register n (ADDRn) format.	R/W

ADCER sets the A/D data register n (ADDRn) format and carries out automatic clearing of registers.

ACE Bit (Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing of the A/D data register n (ADDRn) after it has been read by the CPU, DTC, or DMACA. This function enables update failures of the A/D data register n (ADDRn) to be detected.

ADRFMT Bit (A/D Data Register Format Select)

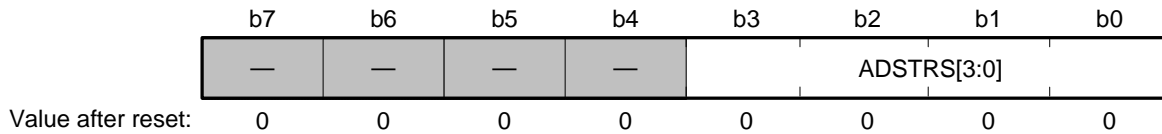
The ADRFMT bit specifies left-alignment or right-alignment for the data to be stored in the A/D data register n (ADDRn).

The format of the A/D data register n (ADDRn) associated with a channel on which A/D-converted value addition mode is selected is fixed to left-alignment, irrespective of the ADRFMT bit value.

For details on the format of the A/D data registers n (ADDRn), see section 34.2.1, A/D Data Registers n (ADDRn) (n = 0 to 7).

34.2.7 A/D Start Trigger Select Register (ADSTRGR)

Address: 0008 9010h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ADSTRS[3:0]	A/D Start Trigger Select	The A/D conversion startup source from the on-chip peripheral modules is selected by the combination of bits 3 to 0.	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

ADSTRGR selects the A/D conversion startup trigger.

- When using the A/D conversion startup source of the MTU and TMR, set the TRGE bit in ADCSR to 1 and set the EXTRG bit in ADCSR to 0.
- When using the A/D conversion startup trigger (ADTRG0#) of an external input, set the TRGE bit in ADCSR to 1 and set the EXTRG bit in ADCSR to 1.
- Software trigger (ADST bit in ADCSR) is always enabled regardless of the set values of the ADCSR.TRGE bit, the ADCSR.EXTRG bit, or the ADSTRGR register.

Table 34.5 List of A/D Conversion Startup Sources

Module	A/D Conversion Startup Sources	Triggers	ADSTRS[3]	ADSTRS[2]	ADSTRS[1]	ADSTRS[0]
ADC	ADST	Software trigger	—	—	—	—
External input	ADTRG0#	A/D conversion startup trigger pin	0	0	0	0
MTU	TRG1N	TRG0AN_0	0	0	0	1
	TRG2N	TRG0BN_0	0	0	1	0
	TRG3N	TRGAN_0	0	0	1	1
	TRG4N	TRGAN_1	0	1	0	0
	TRG5N	TRG0EN_0	0	1	0	1
	TRG6N	TRG0FN_0	0	1	1	0
	TRG7N	TRG4ABN_0	0	1	1	1
	TRG8N	TRG4ABN_1	1	0	0	0
TMR	TRG9N	TMTRG0AN_0	1	0	0	1
	TRG10N	TMTRG0AN_1	1	0	1	0

Note: When ADTRG0# is used as the A/D conversion start source, the Bn and Bn in DDR and ICR of PORTm corresponding to the pin should be set to 0 (input port) and 1 (the input buffer of the corresponding pin enabled), respectively. For details, refer to section 17, I/O Ports.

34.3 Operation

34.3.1 Scan Conversion

A scan conversion is performed in two operating modes: single-cycle scan mode and continuous scan mode. In the single-cycle scan mode, one or more specified channels are scanned once. In the continuous scan mode, one or more specified channels are scanned repeatedly until the ADST bit in ADCSR is set to 0 by software.

In either mode, A/D conversion is performed for ANn channels selected by ADANS, starting from the channel with the lowest number n.

34.3.2 Single-Cycle Scan Mode

In single-cycle scan mode, A/D conversion is to be performed for only once on the analog input of the specified single channel as below.

1. A/D conversion for the selected channel is started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, MTU, TMR, or an external trigger input. A/D conversion is performed for ANSn channels selected by ADANS, starting from the channel with the lowest number n.
2. When A/D conversion for the channel is completed, the A/D conversion result is stored into the corresponding A/D data register n (ADDRn) of the channel.
3. When A/D conversion of all selected channels is completed, if the ADIE bit in ADCSR is set to 1 (A/D interrupt enable by completing A/D conversion), an S12ADI0 interrupt request is generated.
4. The ADST bit remains at 1 during A/D conversion, and is automatically cleared to 0 when A/D conversion of all selected channels ends. Then the A/D converter enters a wait state.

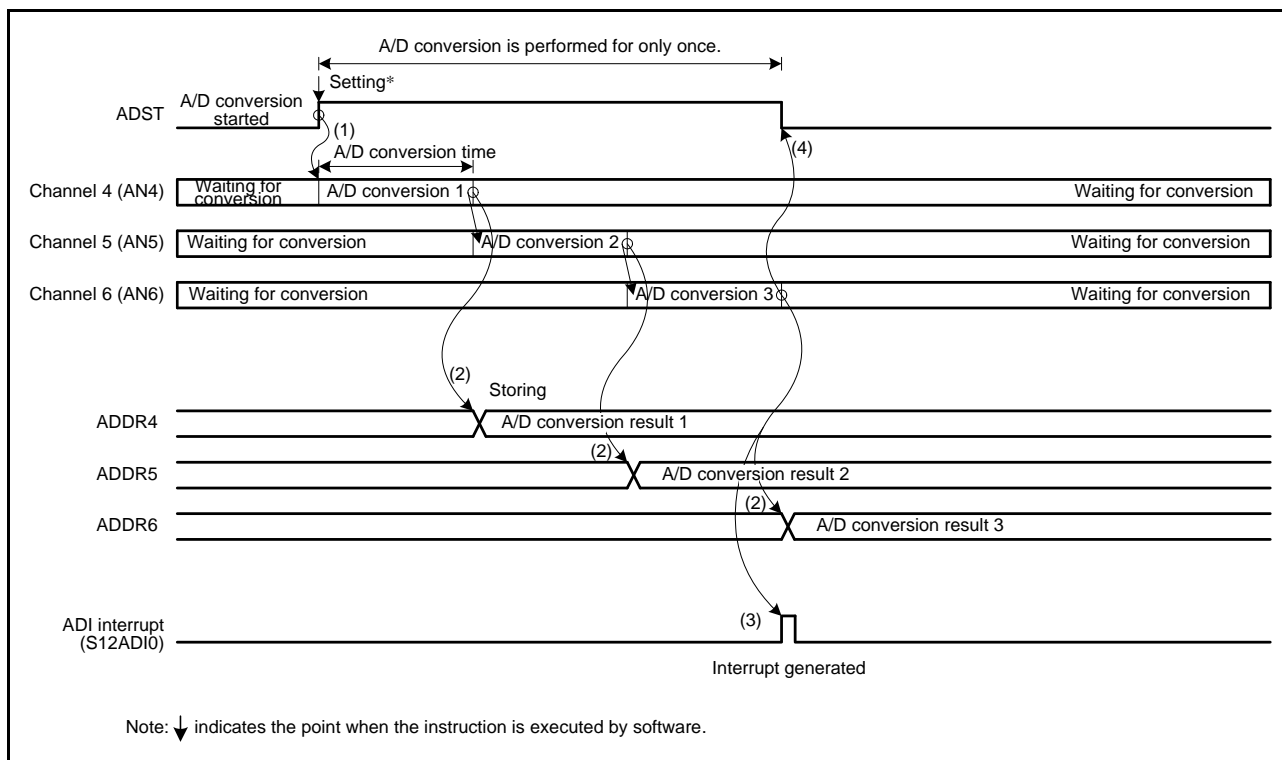


Figure 34.3 example of Operation of Single-Cycle Scan Mode

34.3.3 Continuous Scan Mode

In continuous scan mode, A/D conversion is to be performed sequentially on the analog inputs of the specified channels as below.

1. When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, MTU, TMR, or an external trigger input, A/D conversion is performed for ANn channels selected by ADANS, starting from the channel with the lowest number n.
2. When A/D conversion for the channel is completed, the A/D conversion result is stored into the corresponding A/D data register n (ADDRn).
3. When A/D conversion of all selected channels is completed, if the ADIE bit in ADCSR is set to 1 (A/D interrupt enable by completing A/D conversion), an S12ADI0 interrupt request is generated. A/D converter starts A/D conversion for ANSn channels selected by ADANS, starting from the channel with the lowest number n.
4. The ADST bit is not cleared automatically, and steps 2 to 3 are repeated as long as the ADST bit remains set to 1 (A/D conversion start). When the ADST bit is cleared to 0 (A/D conversion stop), A/D conversion stops and the A/D converter enters a wait state.
5. If the ADST bit is later set to 1, A/D conversion starts again for ANSn channels selected by ADANS, starting from the channel with the lowest number n.

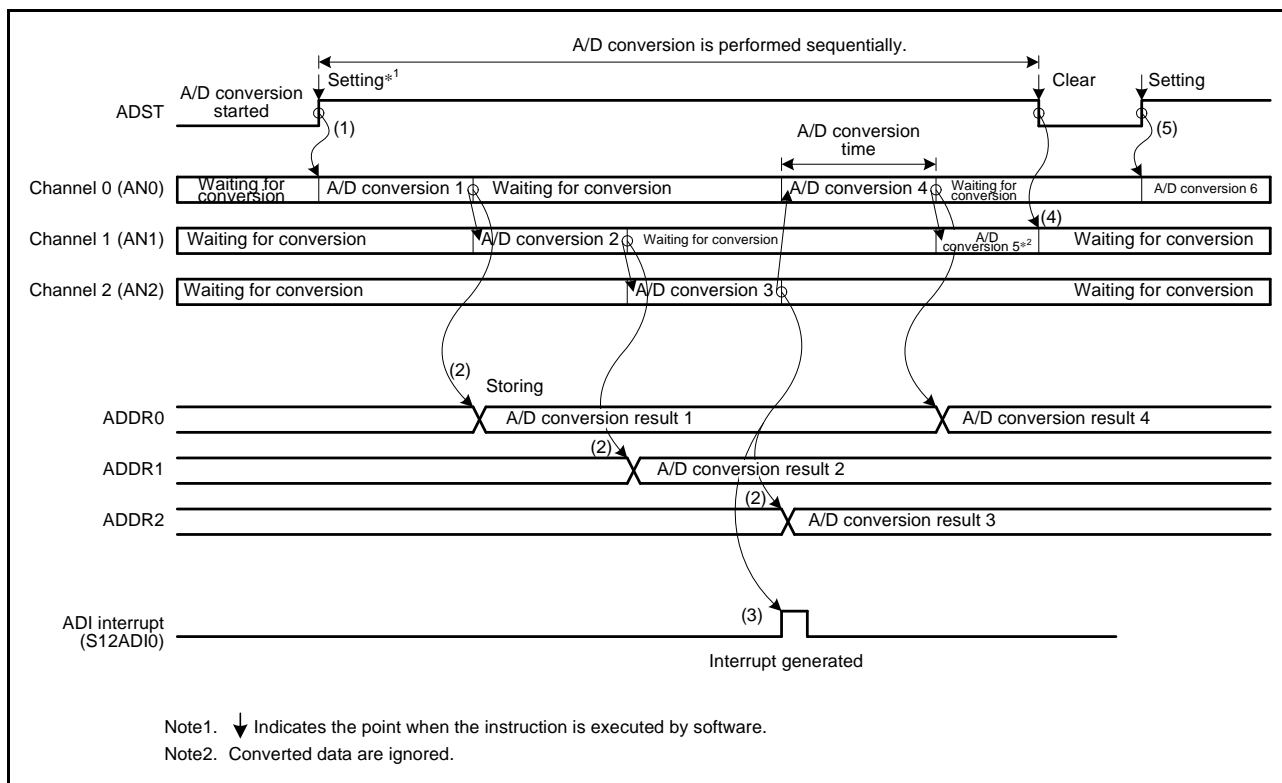


Figure 34.4 Example of Operation of Continuous Scan Mode

34.3.4 Analog Input Sampling and Scan Conversion Time

In scan conversion, activation can be selected from software activation, activation by trigger of MTU and TMR, and activation by ADTRG0# (external trigger). After start-of-scan-conversion delay time (t_D) has passed, the A/D converter samples the analog input and then begins the A/D conversion process.

Scan conversion time (t_{SCAN}) includes start-of-scan-conversion delay time (t_D), A/D conversion processing time (t_{CONV}), and end-of-scan-conversion delay time (t_{ED}). The scan conversion time is shown in Table 34.6.

The scan conversion time (t_{SCAN}) in single-cycle scan mode for which the number of conversions is n can be determined as follows:

$$t_{SCAN} = t_D + (t_{CONV} \times n) + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single-cycle scan minus t_{ED} .

The scan conversion time for the second and subsequent cycles in continuous scan mode is a fixed period equal to $(t_{CONV} \times n)$.

Table 34.6 Scan Conversion Time

Item	Symbol	ADTRG# (External Trigger)	MTU and TMR (Internal Trigger)	Software (Software Trigger)	Unit
Start-of-scan-conversion delay time*2	t_D	4PCLK + 3ADCLK*1	2PCLK + 3ADCLK	2PCLK + 3ADCLK	Cycle
A/D conversion processing time	t_{CONV}	50 ADCLK	50 ADCLK	50 ADCLK	
End-of-scan-conversion delay time*3	t_{ED}	1PCLK + 2ADCLK	1PCLK + 2ADCLK	1PCLK + 2ADCLK	

[Legend]

PCLK: Module clock

ADCLK: A/D conversion clock

Note 1. For the external trigger input timing, see section 41.3.5, Timing of On-Chip Peripheral Modules Timing.

Note 2. The maximum time from software write or trigger input to the start of A/D conversion.

Note 3. The time from the end of A/D conversion to the generation of an A/D conversion end interrupt.

34.3.5 Usage Example of A/D Data Register n (ADDRn) Automatic Clearing Function

Setting the ACE bit in ADCER to 1 automatically clears the A/D data registers n (ADDRn) to 0000h when the ADDRn registers are read by the CPU, DTC, or DMACA.

This function is used to detect update failures of the ADDRn registers.

Examples in which the function to automatically clear the ADDRn registers are enabled and disabled are shown below.

In a case where the ACE bit in ADCER is 0 (automatic clearing is disabled), if the A/D conversion result (0222h) is not written to the ADDRn registers for some reason, the old data (0111h) will become the ADDRn value. Furthermore, if this ADDRn value is written to a general register using an A/D scan conversion end interrupt, the old data (0111h) can be saved in the general register. When checking data that has not been updated, it is necessary to frequently save the old data in the RAM or a general register.

In a case where the ACE bit in ADCER is 1 (automatic clearing is enabled), when ADDRn = 0111h is read by the CPU, DTC, or DMACA, ADDRn is automatically cleared to 0000h. After that, if the A/D conversion result of 0222h cannot be transferred to ADDRn for some reason, the cleared data (0000h) remains as the ADDRn value. If this ADDRn value is written to a general register using an A/D scan conversion end interrupt at this point, 0000h will be saved in the general register. Occurrence of an ADDRn update failure can be determined by simply checking whether the read data value is 0000h.

34.3.6 A/D-Converted Value Addition Function

The same channel is A/D converted continuously two to four times and the sum of the converted values is stored in the A/D data register n (ADDRn). 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.

34.3.7 Starting Scan Conversion with External Trigger

The A/D converter can be activated by the input of an external trigger. To start up the A/D converter by an external trigger, the pin function should be set up using PFCR. After setting the A/D start trigger select register (ADSTRGR) to 00h and applying a high signal to the ADTRG0# pin, both the TRGE and EXTRG bits in ADCSR should be set to 1. Figure 34.5 shows a timing diagram of the external trigger input.

For details on pin function settings, see section 17, I/O Ports.

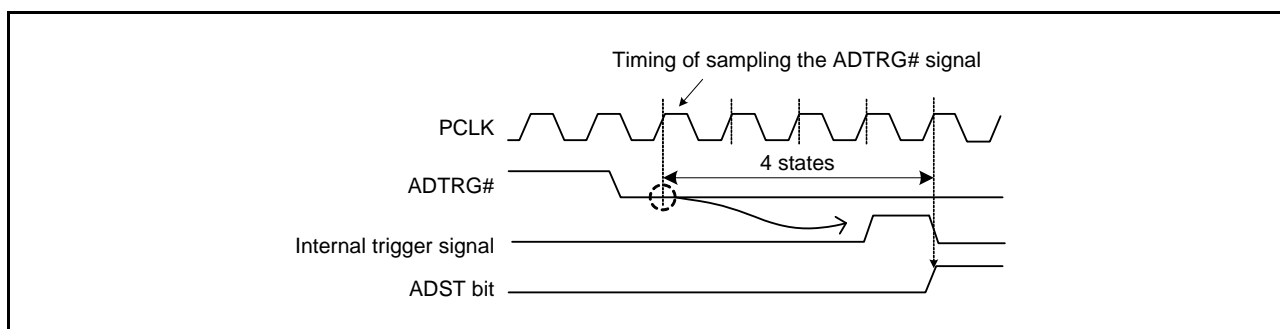


Figure 34.5 External Trigger Input Timing

34.3.8 Starting Scan Conversion with Trigger from Peripheral Modules

A scan conversion can be activated by a timer trigger of the MTU or TMR. To start up a scan conversion by a timer trigger, the TRGE bit in ADCSR should be set to 1, the EXTRG bit in ADCSR should be cleared to 0, and the relevant source should be selected by ADSTRGR.ADSTRS[3:0] bits.

34.3.8.1 A/D Converter Activation with TRG0AN_0 and TRG0BN_0 of MTU

The A/D converter can be activated by generating a trigger signal TRG0AN_0 when an input capture or compare match occurs in TGRA of the MTU (unit 0) in channel 0.

The A/D converter can also be activated by generating a trigger signal TRG0BN_0 when an input capture or compare match occurs in TGRB of the MTU (unit 0) in channel 0.

Figure 34.6 shows a connection of MTU (unit 0) TRG0AN_0 and TRG0BN_0 outputs and the A/D converter.

To start up the A/D converter when an input capture or compare match occurs in TGRA of the MTU (unit 0) in channel 0, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, and ADSTRGR.ADSTRS[3:0] bits should be set to 0001b (to select TRG1N as the source and TRG0AN_0 as the corresponding trigger).

To start up the A/D converter when an input capture or compare match occurs in TGRB of the MTU (unit 0) in channel 0, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, and ADSTRGR.ADSTRS[3:0] bits should be set to 0010b (to select TRG2N as the source and TRG0BN_0 as the corresponding trigger).

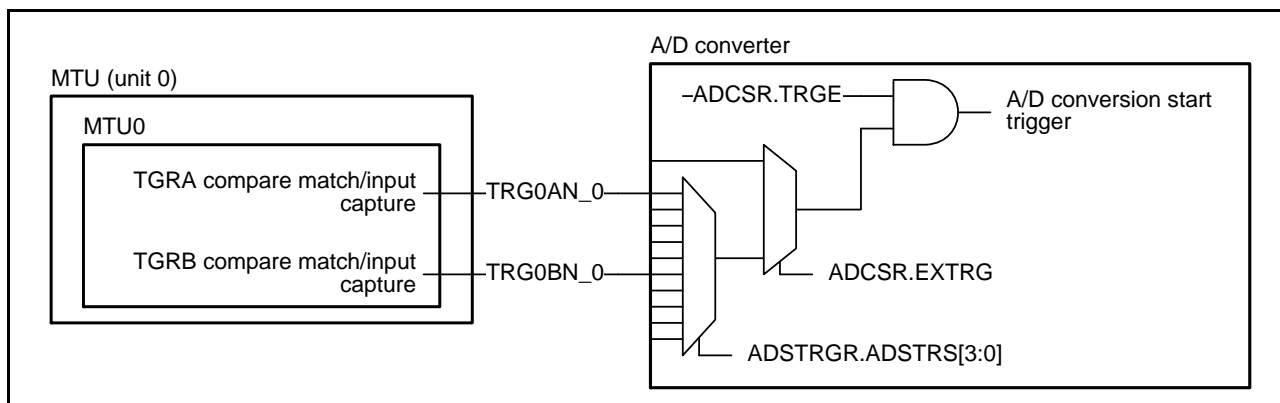


Figure 34.6 Connection of MTU (unit 0) TRG0AN_0 and TRG0BN_0 Outputs and A/D Converter

34.3.8.2 A/D Converter Activation with TRGAN_0 and TRGAN_1 of MTU

The A/D converter can be activated by generating a trigger signal TRGAN_0 when an input capture or compare match occurs in TGRA of the MTU (unit 0) in channels 0 to 4 and a TCNT underflow (trough) occurs in channel 4 in complementary PWM mode. Likewise, the A/D converter can be activated by generating a trigger signal TRGAN_1 when an input capture or compare match occurs in TGRA of the MTU (unit 1) in channels 6 to 10 and a TCNT underflow (trough) occurs in channel 10 in complementary PWM mode.

Figure 34.7 shows a connection of MTU (unit 0, unit 1) TRGAN_0 and TRGAN_1 outputs and the A/D converter.

To start up the A/D converter when an input capture or compare match occurs in TGRA of the MTU (unit 0) in channels 0 and 2, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, ADSTRGR.ADSTRS[3:0] bits should be set to 0011b (to select TRG3N as the source and TRGAN_0 as the corresponding trigger), and the MTU0.TIER.TTGE bit and the MTU2.TIER.TTGE bit should be set to 1.

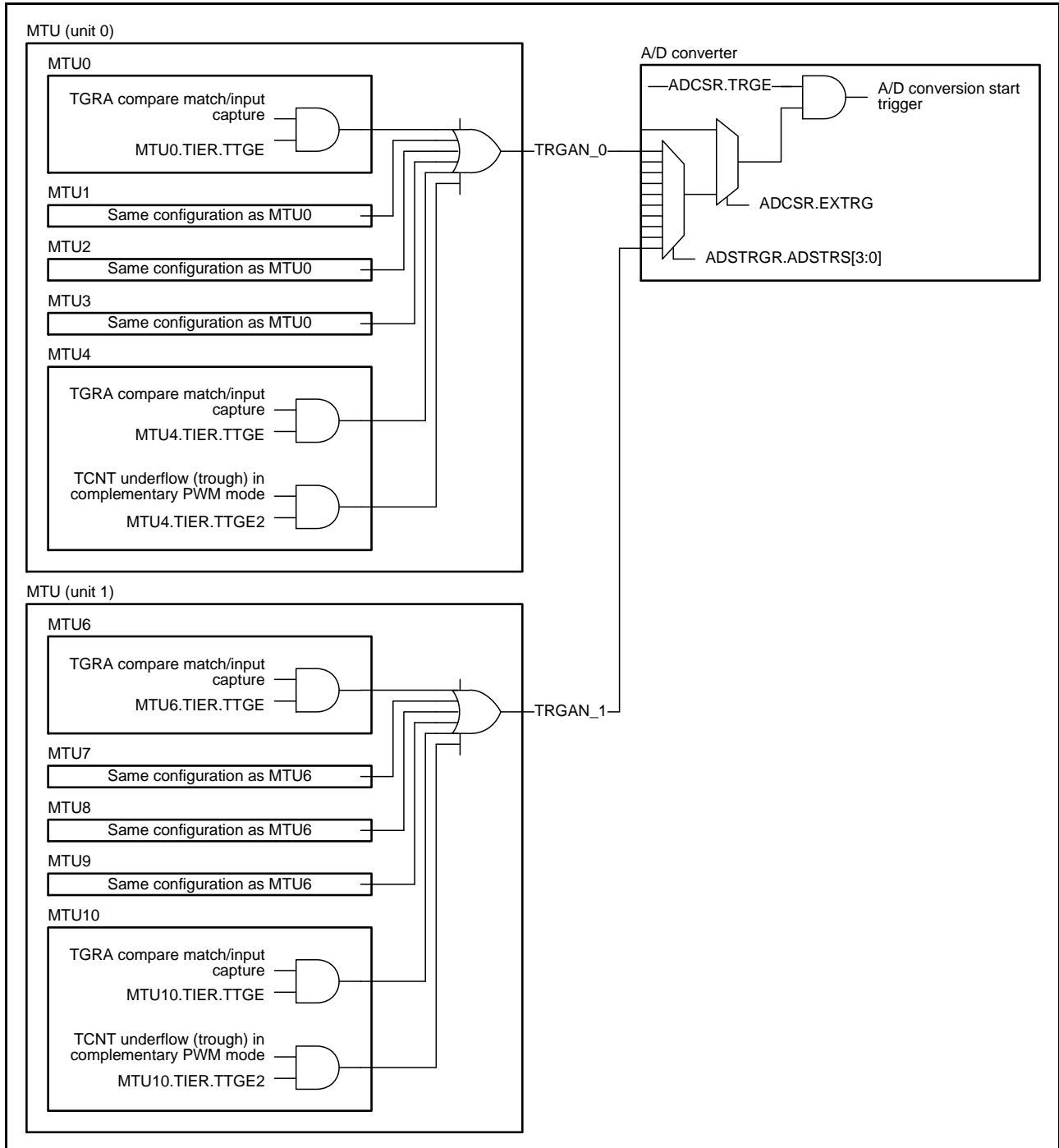


Figure 34.7 Connection of MTU (unit 0, unit 1) TRGAN_0 and TRGAN_1 Outputs and A/D Converter

34.3.8.3 A/D Converter Activation with TRG0EN_0 and TRG0FN_0 of MTU

The A/D converter can be activated by generating a trigger signal TRG0EN_0 when a compare match occurs in TGRE of the MTU (unit 0) in channel 0. The A/D converter can also be activated by generating a trigger signal TRG0FN_0 when a compare match occurs in TGRF of the MTU (unit 0) in channel 0.

Figure 34.8 shows a connection of MTU (unit 0) TRG0EN_0 and TRG0FN_0 outputs and the A/D converter.

To start up the A/D converter when a compare match occurs in TGRE of the MTU (unit 0) in channel 0, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, and ADSTRGR.ADSTRS[3:0] bits should be set to 0101b (to select TRG5N as the source and TRG0EN_0 as the corresponding trigger).

To start up the A/D converter when a compare match occurs in TGRF of the MTU (unit 0) in channel 0, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, and ADSTRGR.ADSTRS[3:0] bits should be set to 0110b (to select TRG6N as the source and TRG0FN_0 as the corresponding trigger).

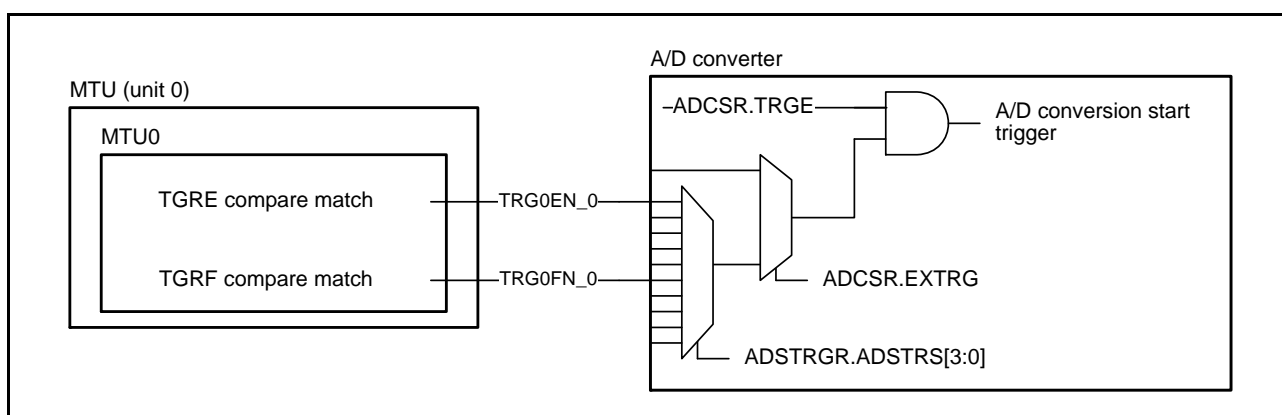


Figure 34.8 Connection of MTU (unit 0) TRG0EN_0 and TRG0FN_0 Outputs and A/D Converter

34.3.8.4 A/D Converter Activation with TRG4ABN_0 and TRG4ABN_1 of MTU

The A/D converter can be activated by generating a trigger signal TRG4ABN_0 when a compare match with the A/D converter start request delaying function of the MTU (unit 0) in channel 4 occurs. Likewise, the A/D converter can be activated by generating a trigger signal TRG4ABN_1 when a compare match with the A/D converter start request delaying function of the MTU (unit 1) in channel 10 occurs.

Figure 34.9 shows a connection of MTU (unit 0, unit 1) TRG4ABN_0 and TRG4ABN_1 outputs and the A/D converter.

To start up the A/D converter by the A/D converter start request delaying function of the MTU (unit 0) in channel 4 when the TCNT count matches the TADCORA value, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, ADSTRGR.ADSTRS[3:0] bits should be set to 0111b (to select TRG7N as the source and TRG4ABN_0 as the corresponding trigger), the cycle should be set to MTU4.TADCOBRA/B and MTU4.TADCORA/B, and the MTU4.TADCR.UT4AE bit should be set to 1.

For details on the A/D converter start request delaying function, see section 17.3.9, A/D Converter Start Request Delaying Function.

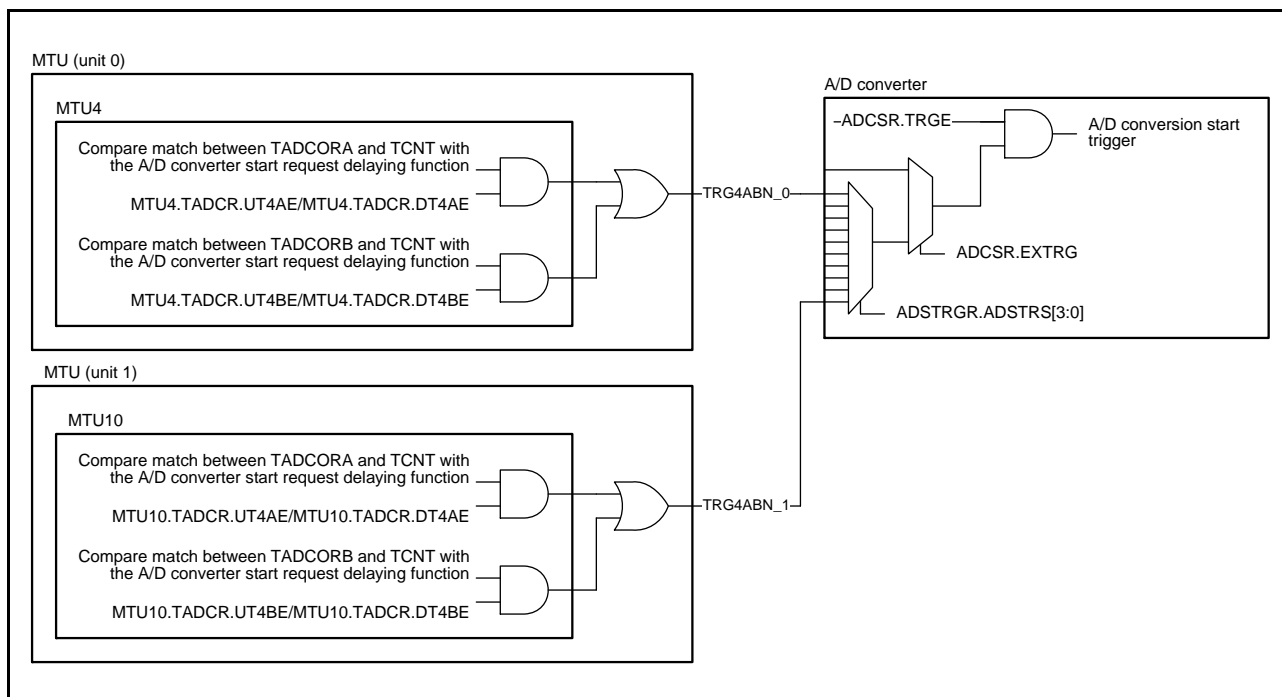


Figure 34.9 Connection of MTU (unit 0, unit 1) TRG4ABN_0 and TRG4ABN_1 Outputs and A/D Converter

34.3.8.5 A/D Converter Activation with TMTRG0AN_0 and TMTRG0AN_1 of TMR

The A/D converter can be activated when a compare match (compare match A) occurs in TCORA of the TMR (unit 0) in channel 0. The A/D converter can also be activated when a compare match (compare match A) occurs in TCORA of the TMR (unit 1) in channel 2.

Figure 34.10 shows a connection of TMR (unit 0, unit 1) TMTRG0AN_0 and TMTRG0AN_1 outputs and the A/D converter.

To start up the A/D converter when a compare match (compare match A) occurs in TCORA of the TMR (unit 0) in channel 0, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, ADSTRGR.ADSTRS[3:0] bits should be set to 1001b (to select TRG9N as the source and TMTRG0AN_0 as the corresponding trigger), and the TMR0.TCSR.ATDE bit should be set to 1.

To start up the A/D converter when a compare match (compare match A) occurs in TCORA of the TMR (unit 1) in channel 2, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, ADSTRGR.ADSTRS[3:0] bits should be set to 1010b (to select TRG10N as the source and TMTRG0AN_1 as the corresponding trigger), and the TMR2.TCSR.ATDE bit should be set to 1.

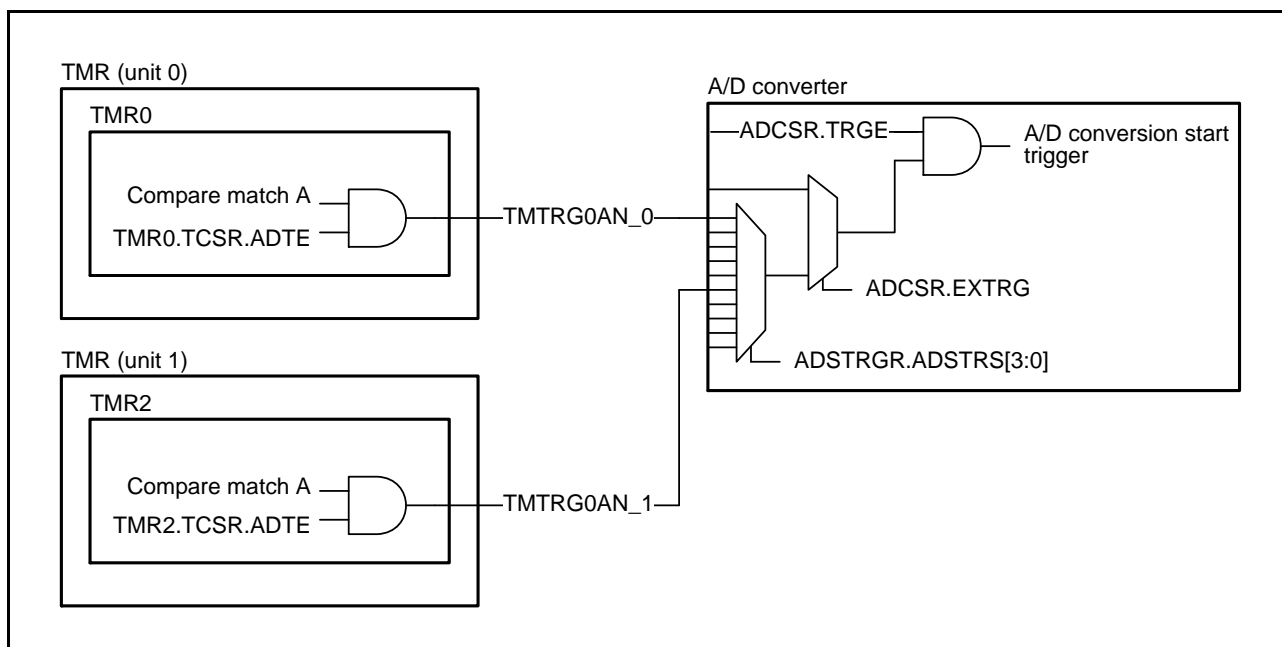


Figure 34.10 Connection of TMR (unit 0, unit 1) TMTRG0AN_0 and TMTRG0AN_1 Outputs and A/D Converter

34.4 Interrupt Sources and DMA Transfer Request

34.4.1 Interrupt Request on Completion of Each Scan Conversion

The A/D converter can send a scan conversion end interrupt request (S12ADI0 interrupt) to the CPU.

By setting the ADIE bit in ADCSR to 1, an S12ADI0 interrupt is enabled; by clearing the ADIE bit to 0, an S12ADI0 interrupt is disabled.

In addition, the DTC or DMACA can be started up when an S12ADI0 interrupt is generated. Using an S12ADI0 interrupt to allow the DMACA to read the converted data enables continuous conversion without burden on software.

For details on DTC settings, see section 16, Data Transfer Controller (DTCa), and for details on DMACA settings, see section 14, DMA Controller (DMACA).

34.5 Usage Notes

34.5.1 Selecting 12-Bit A/D Converter or 10-Bit A/D Converter

The 12-bit A/D converter or 10-bit A/D converter should be selected by the MSTPA23, MSTPA22, and MSTPA17 bits in module stop control register A (MSTPCRA).

Setting the MSTPCRA.MSTPA17 bit to 0 selects the 12-bit A/D converter.

Setting the MSTPCRA.MSTPA23 and MSTPCRA.MSTPA22 bits to 0 selects the 10-bit A/D converter.

If the MSTPCRA.MSTPA23, MSTPCRA.MSTPA22, and MSTPCRA.MSTPA17 bits are all set to 0 at the same time, the MSTPCRA.MSTPA17 bit setting takes priority and the MSTPCRA.MSTPA23 and MSTPA22 bit settings are ignored; that is, the 10-bit A/D converter is not selected in this case.

For details, see section 9.2.2, Module Stop Control Register A (MSTPCRA), in section 9, Low Power Consumption.

34.5.2 Module Stop Function Setting

Operation of the A/D converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the A/D converter to be halted. After the module stop state is canceled, wait for 10 ms to start A/D conversion. For details, see section 9, Low Power Consumption.

34.5.3 Notes on Restarting A/D Conversion

Stopping analog input to the A/D converter by clearing the ADST bit in ADCSR to 0, requires two cycles of the ADCLK. Starting analog input to the A/D converter by setting the ADST bit in ADCSR to 1, requires three cycles of the ADCLK.

34.5.4 Notes on Disabling A/D Conversion

To disable A/D conversion when an external trigger or timer has been selected as the condition for starting A/D conversion, set the ADCSR.TRGE bit to 0 to select the software trigger as the trigger to start A/D conversion, and then set the ADST bit in ADCSR to 0 (to stop A/D conversion).

34.5.5 Notes on Entering Low Power Consumption States

When the RX62N/RX621 Group enters the module stop state or software standby mode, make sure to stop the A/D conversion. To do so, set the ADST bit in ADCSR to 0, and allow time for disabling of the analog input to the A/D converter.

Follow the procedure given below to ensure that this time is secured.

- (1) Set the ADCSR.TRGE bit to 0 (software trigger).
- (2) Clear the ADCSR.ADST bit to 0.
- (3) Set the ADCR.CKS[1:0] bits to 11b (PCLK).
- (4) After confirming that the A/D converter has been disabled (at least 6 PCLK cycles are required to stop the conversion), place the LSI in the module stop state or software standby mode.

34.5.6 Notes on Canceling Software Standby Mode

When the software standby mode is canceled, wait until the crystal oscillation settling time or PLL-circuit settling time has elapsed. Then, wait for another 10 ms to start the A/D conversion. For details, see section 9, Low Power Consumption.

34.5.7 Notes on Using A/D Converter and D/A Converter Simultaneously

Since the A/D converter and the D/A converter use the same power supply, using them simultaneously may affect A/D conversion accuracy.

If the following registers are set during A/D conversion as below, it may make the accuracy worse by approximately 2 LSB.

- Data is written to the D/A data register m (DADRm) of the D/A converter during A/D conversion.
- Data is written to the D/A control register (DACR) during A/D conversion and when the DADRm register value of the D/A converter is not 00h.

If the above settings affect the conversion accuracy, take any of the following countermeasures.

(1) Average the A/D conversion results using a program.

Example of averaging: Perform A/D conversion four consecutive times for the same pin, and then calculate the average between two values other than the maximum and minimum values of the A/D conversion results.

(2) Discard the A/D conversion result during conversion.

(3) Write data to the DACR register of the D/A converter after setting the DADRm register value of the D/A converter to 00h.

35. 10-Bit A/D Converter (ADa)

35.1 Overview

The RX62N/RX621 Group includes two successive approximation type 10-bit A/D converters (units 0 and 1). Each unit allows up to four analog input channels to be selected.

The A/D converter has two modes of operation: single mode which converts the analog input of a specified channel once, and scan mode which continuously converts the analog inputs of up to four channels.

Table 35.1 lists the specifications of the A/D converter and the Table 35.2 compares the two units Figure 35.1 and Figure 35.2 show block diagrams of the A/D converter units 1 and 2, respectively.

Note that either the 12-bit A/D converter or the 10-bit A/D converter should be exclusively selected by the MSTPA23, MSTPA22, and MSTPA17 bits in module stop control register A (MSTPCRA).

Table 35.1 Specifications of A/D Converter

Item	Specifications
Number of units	Two units
Input channels	Each unit: 4 channels (total 8 channels)
A/D conversion method	Successive approximation method
Resolution	10 bits
Conversion time	1.0 μ s per 1 channel (when operating peripheral module clock PCLK = 50 MHz)
A/D conversion clock	4 types: PCLK, PCLK/2, PCLK/4, PCLK/8
Operating modes	<ul style="list-style-type: none"> • Single mode: A/D conversion is performed once on the analog input of a specified channel. • Scan mode <ul style="list-style-type: none"> Continuous scan mode: A/D conversion is performed sequentially on the analog inputs of up to four specified channels. One-cycle scan mode: A/D conversion is performed for a single cycle on the analog inputs of up to four specified channels.
Conditions of A/D conversion start	<ul style="list-style-type: none"> • Software trigger • Conversion start trigger by the multi-function timer pulse unit (MTU) or 8-bit timer (TMR). • External trigger <ul style="list-style-type: none"> A/D conversion for each unit can be triggered from the ADTRGn# pin.
Function	<ul style="list-style-type: none"> • Sample-and-hold function • Number of sampling state is adjustable. • Self-diagnostic functions
Interrupt source	<ul style="list-style-type: none"> • A/D conversion end interrupt (ADI) request can be generated by each unit. • An ADI interrupt can activate data transfer controller (DTC) or DMA controller (DMACA).
Power-down function	Module stop state can be set for each unit.

Table 35.2 Comparison Overview by Each Unit

Item		Function			Unit 0 (AD0)	Unit 1 (AD1)
		Internal Trigger Source				
Analog input channel					AN0 AN1 AN2 AN3	AN4 AN5 AN6 AN7
A/D conversion start conditions *1	Software trigger	Software trigger			√	√
	External trigger	ADTRG0#			√	—
		ADTRG1#			—	√
	Internal trigger (MTU, TMR)*3	TRG0AN_0	MTU0.TGRA and MTU0.TCNT	Input capture/compare match	√	—
			TRG0BN_0	MTU0.TGRB and MTU0.TCNT	Input capture/compare match	—
		TRGAN_0	MTU0.TGRA and MTU0.TCNT	Input capture/compare match	√	√
			MTU1.TGRA and MTU1.TCNT			
			MTU2.TGRA and MTU2.TCNT			
			MTU3.TGRA and MTU3.TCNT			
			MTU4.TGRA and MTU4.TCNT			
			MTU4.TCNT	TCNT underflow (trough) in complementary PWM mode		
TRGAN_1		MTU6.TGRA and MTU6.TCNT	Input capture/compare match	√	√	
		MTU7.TGRA and MTU7.TCNT		√	√	
	MTU8.TGRA and MTU8.TCNT		√	√		
	MTU9.TGRA and MTU9.TCNT		√	√		
	MTU10.TGRA and MTU10.TCNT		√	√		
	MTU10.TCNT	TCNT underflow (trough) in complementary PWM mode	√	√		
TRG4ABN_0	MTU4.TADCORA and MTU4.TCNT or MTU4.TADCORB and MTU4.TCNT	Compare match with the A/D converter start request delaying function	√	√		
TRG4ABN_1	MTU10.TADCORA and MTU10.TCNT or MTU10.TADCORB and MTU10.TCNT	Compare match with the A/D converter start request delaying function	√	√		
TMTRG0AN_0	TMR0.TCORA and TMR0.TCNT	Compare match	√	√		
Interrupt					ADI0	ADI1
Module stop function setting*2					MSTPCRA.MS TPA23 bit	MSTPCRA.M STPA22 bit

[Legend]

- √: Enabled
- : Disabled

Note 1. A/D conversion start conditions can be selected by each unit.

Note 2. For details, see section 9, Low Power Consumption.

Note 3. _0 and _1 suffixed to internal trigger names indicate a unit number. For internal trigger output setting, see section 17.4.3, A/D Converter Activation and section 21.6.2, A/D Converter Activation.

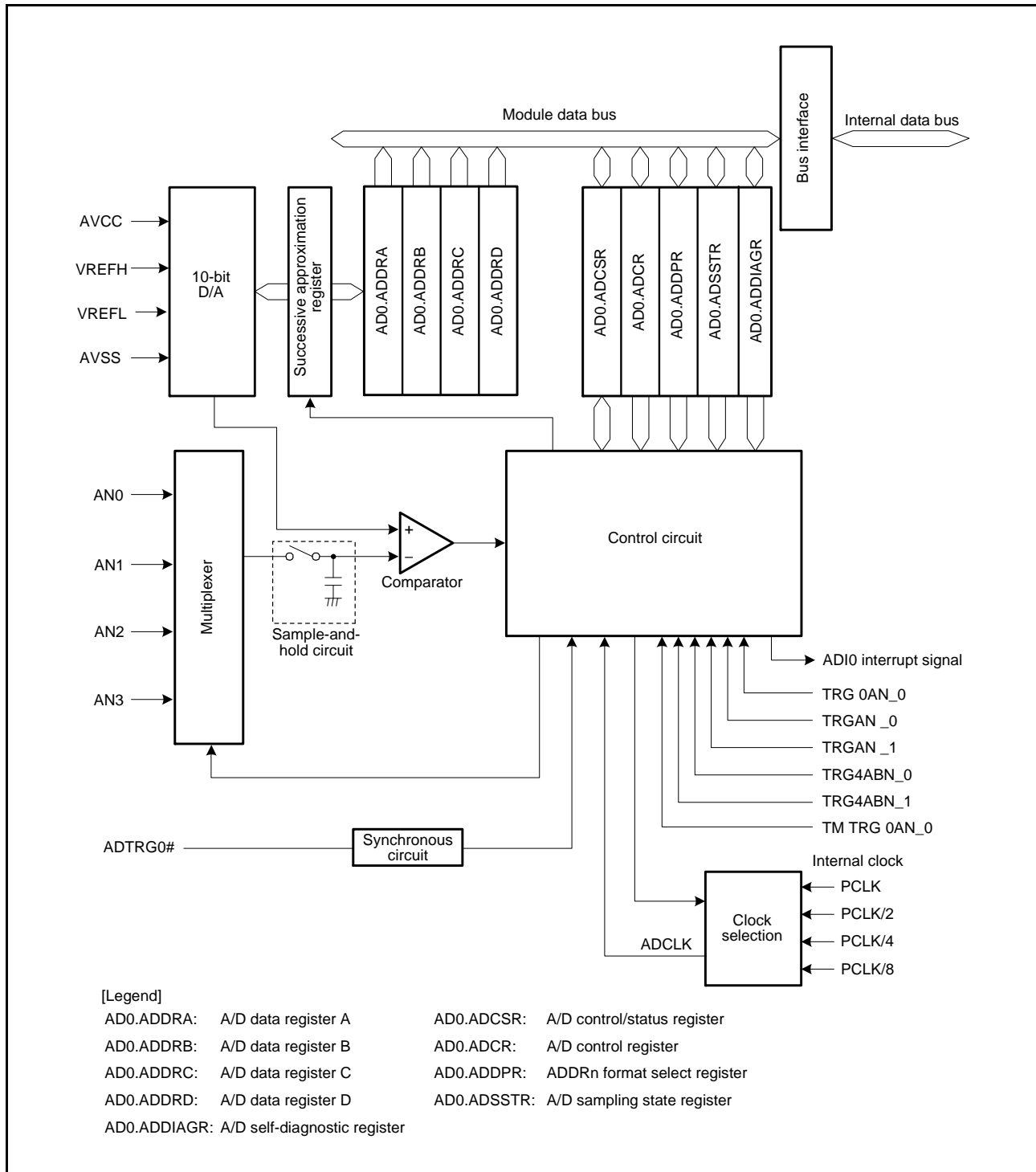


Figure 35.1 Block Diagram of A/D Converter Unit 0 (AD0)

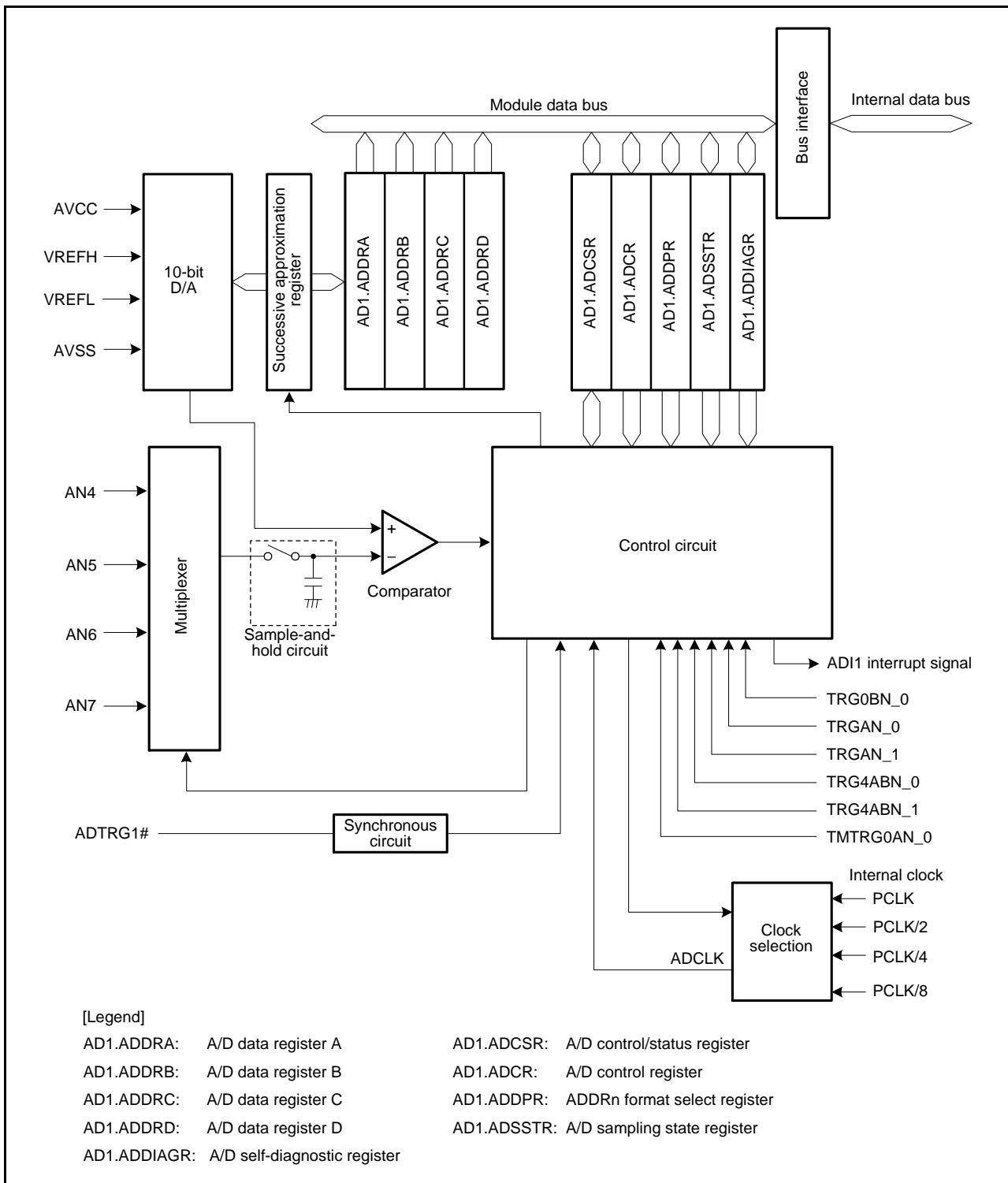


Figure 35.2 Block Diagram of A/D Converter Unit 1 (AD1)

Table 35.3 indicates the input pins of the A/D converter.

Table 35.3 Input Pins of A/D Converter

Unit	Module Symbol	Pin Name	Input	Function
0	AD0	AN0 to AN3	Input	Analog input pins
		ADTRG0#	Input	External trigger input pin for starting A/D conversion
1	AD1	AN4 to AN7	Input	Analog input pins
		ADTRG1#	Input	External trigger input pin for starting A/D conversion
Common		AVCC	Input	Analog block power supply pin
		AVSS	Input	Analog block ground pin
		VREFH	Input	A/D conversion reference power supply pin
		VREFL	Input	Ground pin for the A/D conversion reference power supply Connect this pin to the analog reference power supply (0 V).

35.2 Register Descriptions

Table 35.4 lists the registers of the A/D converter.

Table 35.4 Registers of A/D Converter

Unit	Module Symbol	Register Name	Register Symbol	Value after Reset	Address	Access Size
0	AD0	A/D data register A	ADDRA	0000h	0008 8040h	16
		A/D data register B	ADDRB	0000h	0008 8042h	16
		A/D data register C	ADDRC	0000h	0008 8044h	16
		A/D data register D	ADDRD	0000h	0008 8046h	16
		A/D control/status register	ADCSR	x0h	0008 8050h	8
		A/D control register	ADCR	00h	0008 8051h	8
		ADDRn format select register	ADDP	00h	0008 8052h	8
		A/D sampling state register	ADSSTR	19h	0008 8053h	8
		A/D self-diagnostic register	ADDIAGR	00h	0008 805Fh	8
1	AD1	A/D data register A	ADDRA	0000h	0008 8060h	16
		A/D data register B	ADDRB	0000h	0008 8062h	16
		A/D data register C	ADDRC	0000h	0008 8064h	16
		A/D data register D	ADDRD	0000h	0008 8066h	16
		A/D control/status register	ADCSR	x0h	0008 8070h	8
		A/D control register	ADCR	00h	0008 8071h	8
		ADDRn format select register	ADDP	00h	0008 8072h	8
		A/D sampling state register	ADSSTR	19h	0008 8073h	8
		A/D self-diagnostic register	ADDIAGR	00h	0008 807Fh	8

35.2.1 A/D Data Register n (ADDRn) (n = A to D)

Addresses: AD0.ADDRA 0008 8040h, AD0.ADDRB 0008 8042h, AD0.ADDRC 0008 8044h, AD0.ADDRD 0008 8046h
 AD1.ADDRA 0008 8060h, AD1.ADDRB 0008 8062h, AD1.ADDRC 0008 8064h, AD1.ADDRD 0008 8066h

ADDPR.DPSEL bit = 0 (Data padded at the LSB end)



ADDPR.DPSEL bit = 1 (Data padded at the MSB end)



ADDRn registers are 16-bit read-only registers, which store an A/D conversion result for each channel.

Table 35.5 lists the analog input channels and corresponding ADDRn registers.

10-bit data can be relocated by setting the DPSEL bit in ADDPR.

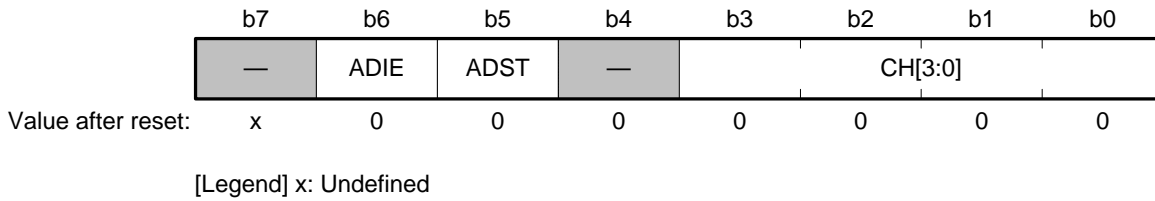
Bits "—" are always read as 0. The write value should always be 0.

Table 35.5 Analog Input Channels and Corresponding ADDRn Registers

Analog Input Channel	ADDRn Register
AN0	AD0.ADDRA
AN1	AD0.ADDRB
AN2	AD0.ADDRC
AN3	AD0.ADDRD
AN4	AD1.ADDRA
AN5	AD1.ADDRB
AN6	AD1.ADDRC
AN7	AD1.ADDRD

35.2.2 A/D Control/Status Register (ADCSR)

Addresses: AD0.ADCSR 0008 8050h, AD1.ADCSR 0008 8070h



Bit	Symbol	Bit Name	Description	R/W		
b3 to b0	CH[3:0]	Channel Select*	Unit	Single mode (ADCR.MODE[1:0] = 00b)	Scan mode (ADCR.MODE[1:0] = 10b or 11b)	R/W
			Unit 0	b3 b0 0 0 0 0: AN0 0 0 0 1: AN1 0 0 1 0: AN2 0 0 1 1: AN3 Settings other than above are prohibited.	b3 b0 0 0 0 0: AN0 0 0 0 1: AN0 and AN1 0 0 1 0: AN0 to AN2 0 0 1 1: AN0 to AN3 Settings other than above are prohibited.	
			Unit 1	b3 b0 0 0 0 0: AN4 0 0 0 1: AN5 0 0 1 0: AN6 0 0 1 1: AN7 Settings other than above are prohibited.	b3 b0 0 0 0 0: AN4 0 0 0 1: AN4 and AN5 0 0 1 0: AN4 to AN6 0 0 1 1: AN4 to AN7 Settings other than above are prohibited.	
b4	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W		
b5	ADST	A/D Start	0: Stops A/D conversion 1: Starts A/D conversion	R/W		
b6	ADIE	A/D Interrupt Enable	0: ADI interrupt is disabled by completing A/D conversion 1: ADI interrupt is enabled by completing A/D conversion	R/W		
b7	—	Reserved	The read value is undefined. The write value should always be 1.	R/W		

Note : * The PORTn.DDR.Bi bit for the analog input should be set to 0 (input port) and the PORTn.ICR.Bi bit should be set to 0 (disabling the input buffer and fixing the input signal to the high level). For details, see section 17, I/O Ports. (n = 4, i = 7 to 0)

ADCSR controls the A/D conversion operations.

CH[3:0] Bits (Channel Select)

These bits select analog input channels that allow A/D conversion.

- Single mode (ADCR.MODE[1:0] bits = 00b)
Select the single analog input channel that allows A/D.
- Scan mode (ADCR.MODE[1:0] bits = 10b or 11b)
Select analog input channels up to 4 that allow A/D conversion.

ADST Bit (A/D Start)

The ADST bit starts/stops A/D conversion.

Before setting the ADST bit to 1, complete the setting for A/D conversion clock and the operation mode.

[Setting conditions]

- When 1 is written by software
- Detection of the trigger selected by the ADCR.TRGS [2:0] bits

[Clearing conditions]

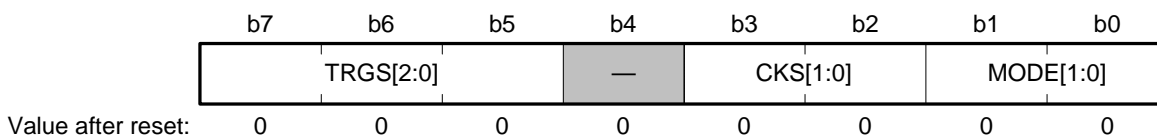
- When 0 is written by software
- The A/D conversion is completed in single mode
- The A/D conversion is completed on every selected channel in one-cycle scan mode.

ADIE Bit (A/D Interrupt Enable)

The ADIE bit enables/disables the A/D conversion end interrupt (ADI).

35.2.3 A/D Control Register (ADCR)

Addresses: AD0.ADCR 0008 8051h, AD1.ADCR 0008 8071h



Bit	Symbol	Bit Name	Description	R/W						
b1, b0	MODE[1:0]	Operation Mode Select	b1 b0 0 0: Single mode 0 1: Setting prohibited 1 0: Continuous scan mode 1 1: One-cycle scan mode	R/W						
b3, b2	CKS[1:0]	Clock Select	b3 b2 0 0: PCLK/8 0 1: PCLK/4 1 0: PCLK/2 1 1: PCLK	R/W						
b4	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W						
b7 to b5	TRGS[2:0]	Trigger Select	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Unit</th> <th>Trigger signal</th> </tr> </thead> <tbody> <tr> <td>Unit 0</td> <td> b7 b6 b5 0 0 0: Software trigger 0 0 1: Compare-match/input-capture A from MTU0 to MTU4 0 1 0: Compare-match A from TMR0 0 1 1: Trigger from ADTRG0#* 1 0 0: Compare-match/input-capture A from MTU0 1 0 1: Compare-match/input-capture A from MTU6 to MTU10 1 1 0: Compare-match from MTU4 1 1 1: Compare-match from MTU10 </td> </tr> <tr> <td>Unit 1</td> <td> b7 b6 b5 0 0 0: Software trigger 0 0 1: Compare-match/input-capture A from MTU0 to MTU4 0 1 0: Compare-match A from TMR0 0 1 1: Trigger from ADTRG1#* 1 0 0: Compare-match/input-capture B from MTU0 1 0 1: Compare-match/input-capture A from MTU6 to MTU10 1 1 0: Compare-match from MTU4 1 1 1: Compare-match from MTU10 </td> </tr> </tbody> </table>	Unit	Trigger signal	Unit 0	b7 b6 b5 0 0 0: Software trigger 0 0 1: Compare-match/input-capture A from MTU0 to MTU4 0 1 0: Compare-match A from TMR0 0 1 1: Trigger from ADTRG0#* 1 0 0: Compare-match/input-capture A from MTU0 1 0 1: Compare-match/input-capture A from MTU6 to MTU10 1 1 0: Compare-match from MTU4 1 1 1: Compare-match from MTU10	Unit 1	b7 b6 b5 0 0 0: Software trigger 0 0 1: Compare-match/input-capture A from MTU0 to MTU4 0 1 0: Compare-match A from TMR0 0 1 1: Trigger from ADTRG1#* 1 0 0: Compare-match/input-capture B from MTU0 1 0 1: Compare-match/input-capture A from MTU6 to MTU10 1 1 0: Compare-match from MTU4 1 1 1: Compare-match from MTU10	R/W
Unit	Trigger signal									
Unit 0	b7 b6 b5 0 0 0: Software trigger 0 0 1: Compare-match/input-capture A from MTU0 to MTU4 0 1 0: Compare-match A from TMR0 0 1 1: Trigger from ADTRG0#* 1 0 0: Compare-match/input-capture A from MTU0 1 0 1: Compare-match/input-capture A from MTU6 to MTU10 1 1 0: Compare-match from MTU4 1 1 1: Compare-match from MTU10									
Unit 1	b7 b6 b5 0 0 0: Software trigger 0 0 1: Compare-match/input-capture A from MTU0 to MTU4 0 1 0: Compare-match A from TMR0 0 1 1: Trigger from ADTRG1#* 1 0 0: Compare-match/input-capture B from MTU0 1 0 1: Compare-match/input-capture A from MTU6 to MTU10 1 1 0: Compare-match from MTU4 1 1 1: Compare-match from MTU10									

Note : * To start the A/D conversion by the ADTRGm# pin, the PORTn.DDR.Bi bit for the corresponding pin should be set to 0 (input port) and the PORTn.ICR.Bi bit should be set to 1 (input buffer for the corresponding pin is valid). For details, see section 17, I/O Ports. (m = 0, 1, n = 0, 1, i = 7, 3)

ADCR enables setting for an A/D conversion start trigger, an operating mode, and A/D conversion clock mode.
Set ADCR while the ADST bit in ADCSR is 0.

MODE[1:0] Bits (Operating Mode Select)

These bits select the A/D conversion operation mode.

CKS[1:0] Bits (Clock Select)

These bits set the frequency of the A/D conversion clock (ADCLK) and thus select the A/D conversion time.

Set the frequency of ADCLK to 4 MHz or higher.

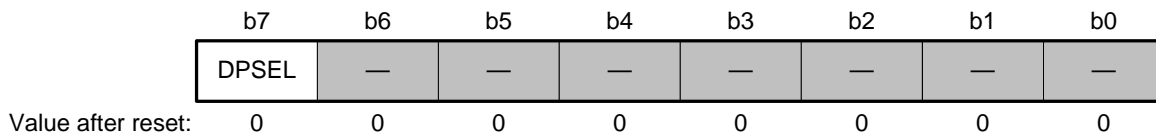
For details, see section 35.3.3, Input Sampling and A/D Conversion Time.

TRGS[2:0] Bits (Trigger Select)

These bits select the A/D conversion start trigger.

35.2.4 ADDRn Format Select Register (ADDPR)

Addresses: AD0.ADDPR 0008 8052h, AD1.ADDPR 0008 8072h



Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	DPSEL	ADDRn Format Select Register	0: A/D data is flushed at the LSB end. 1: A/D data is flushed at the MSB end.	R/W

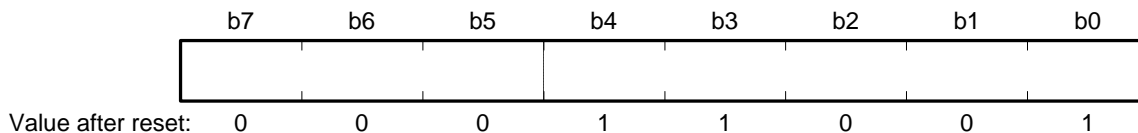
ADDPR selects the placement of data in the A/D data registers.

DPSEL Bit (ADDRn Format Select Register)

The DPSEL bit selects whether data in the A/D data register n (ADDRn) is flushed at the LSB or MSB end.

35.2.5 A/D Sampling State Register (ADSSTR)

Addresses: AD0.ADSSTR 0008 8053h, AD1.ADSSTR 0008 8073h



ADSSTR is an 8-bit readable/writable register that is used to set the sampling time for analog inputs.

Sampling time is adjustable when the signal source impedance of analog input is high and the sampling time is insufficient or the speed of peripheral module clock (PCLK) is low.

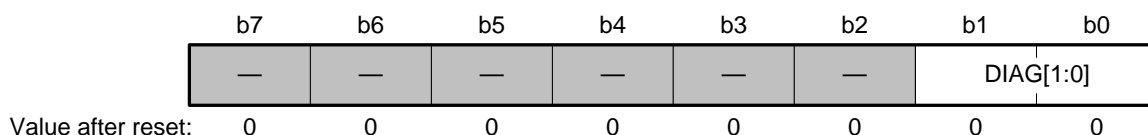
Set the value of 02h or larger.

Ensure to rewrite this register while the A/D conversion is stopped (the ADST bit in ADCSR = 0) in order to prevent incorrect operation.

For details, see section 35.3.3, Input Sampling and A/D Conversion Time.

35.2.6 A/D Self-Diagnostic Register (ADDIAGR)

Addresses: AD0.ADDIAGR 0008 805Fh, AD1.ADDIAGR 0008 807Fh



Bit	Symbol	Bit Name	Description	R/W
b1, b0	DIAG[1:0]	Self-Diagnostic	b1 b0 0 0: Self-diagnostic function is off. 0 1: A/D conversion of Vref x 0 voltage value is enabled. 1 0: A/D conversion of Vref x 1/2 voltage value is enabled. 1 1: A/D conversion of Vref x 1 voltage value is enabled.	R/W
b7 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

ADDIAGR specifies the self-diagnostic function.

Specify ADDIAGR while the ADCSR.ADST bit is 0.

DIAG[1:0] Bits (Self-Diagnostic)

The self-diagnostic function is used to detect faults in the A/D converter. A value for internally generated voltage, Vref x 0, Vref x 1/2, or Vref x 1, is selected.

To perform the self-diagnostic, select the voltage value by the ADDIAGR.DIAG[1:0] bits and start the A/D conversion in the procedure below.

- Single mode (the ADCR.MODE[1:0] bits = 00)
- Analog input AN0 is only enabled (the ADCSR.CH[3:0] bits = 0000)*
- A/D conversion is started by software. (the ADCR.TRGS[2:0] bits = 000)

After A/D conversion ends, the conversion result is stored in the A/D data register A. Then, whether the conversion result is within the normal range (normal) or not (error) can be found out from the values read from ADDRA by software. The execution time required for self-diagnostic is the same as that for A/D conversion of one channel.

Note: * As the input channel, set AN0 to perform the self-diagnostic of A/D converter (unit 0), and set AN4 to perform the self-diagnostic of A/D converter (unit 1). Though this setting is required to select the data register to store the conversion result, all-analog input (AN to AN7) has no effect.

35.3 Operation

The RX62N/RX621 Group includes two units of A/D converter and the each unit has a same feature.

Definitions of a single unit are given below.

The A/D converter has two operating modes: single mode and scan mode.

In single mode, A/D conversion is to be performed for only once on the analog input of the specified single channel.

In scan mode, A/D conversion is to be performed sequentially on the analog inputs of the specified channels up to four.

Two types of scan mode are provided, that is, continuous scan mode where A/D conversion is repeatedly performed and one-cycle scan mode where A/D conversion is performed for the specified channels for one cycle.

35.3.1 Single Mode

In single mode, A/D conversion is to be performed for only once on the analog input of the specified single channel as below.

1. A/D conversion for the selected channel is started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, MTU, TMR, or an external trigger input.
2. When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRn) of the channel.
3. When A/D conversion is completed, if the ADIE bit in ADCSR is set to 1 (A/D interrupt enable by completing A/D conversion), an ADI interrupt request is generated.
4. The ADST bit remains at 1 during A/D conversion, and is automatically cleared to 0 when A/D conversion ends. Then the A/D converter enters a wait state.
5. If the ADST bit is cleared to 0 during A/D conversion (A/D conversion stop), A/D conversion stops and the A/D converter enters a wait state.

Figure 35.3 shows an example of operation when AN1 is selected as an analog input.

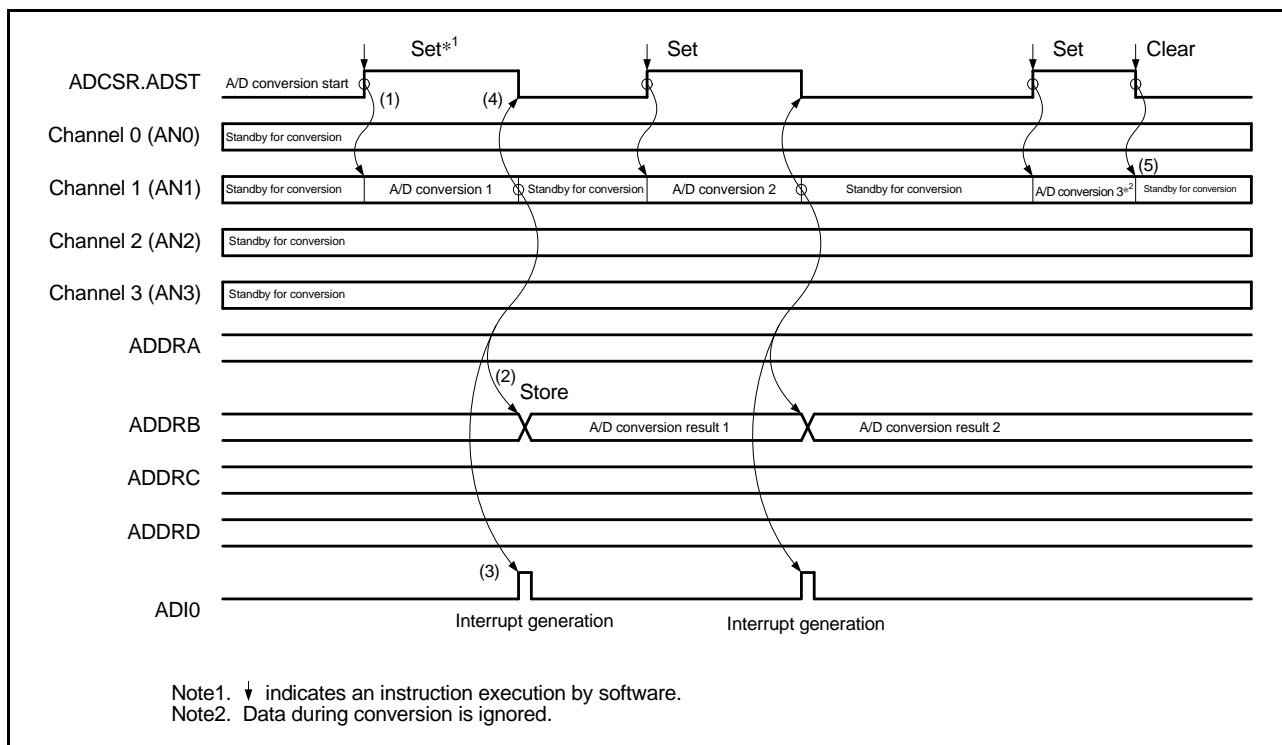


Figure 35.3 Example of A/D Converter Operation (Single Mode)

35.3.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the analog inputs of the specified channels up to four. Two types of scan mode are provided, that is, continuous scan mode where A/D conversion is repeatedly performed and one-cycle scan mode where A/D conversion is performed for the specified channels for one cycle.

35.3.2.1 Continuous Scan Mode

In continuous scan mode, A/D conversion is to be performed sequentially on the analog inputs of the specified channels as below.

1. When the ADST bit in ADCSR is set to 1 by software, MTU, TMR, or an external trigger input, A/D conversion starts on the first channel in the specified channel group.
2. When A/D conversion for each channel is completed, the A/D conversion result is stored into the corresponding A/D data register n (ADDRn).
3. When A/D conversion of all selected channels is completed, if the ADIE bit in ADCSR is set to 1 (A/D interrupt enable by completing A/D conversion), an ADI interrupt request is generated. A/D converter starts A/D conversion from the first channel.
4. The ADST bit is not cleared automatically, and steps 2 to 3 are repeated as long as the ADST bit remains set to 1 (A/D conversion start). When the ADST bit is cleared to 0 (A/D conversion stop), A/D conversion stops and the A/D converter enters a wait state.
5. If the ADST bit is later set to 1 (A/D conversion start), A/D conversion starts again from the first channel in the group.

Figure 35.4 shows an example of A/D conversion when three channels (AN0 to AN2) are selected for analog input.

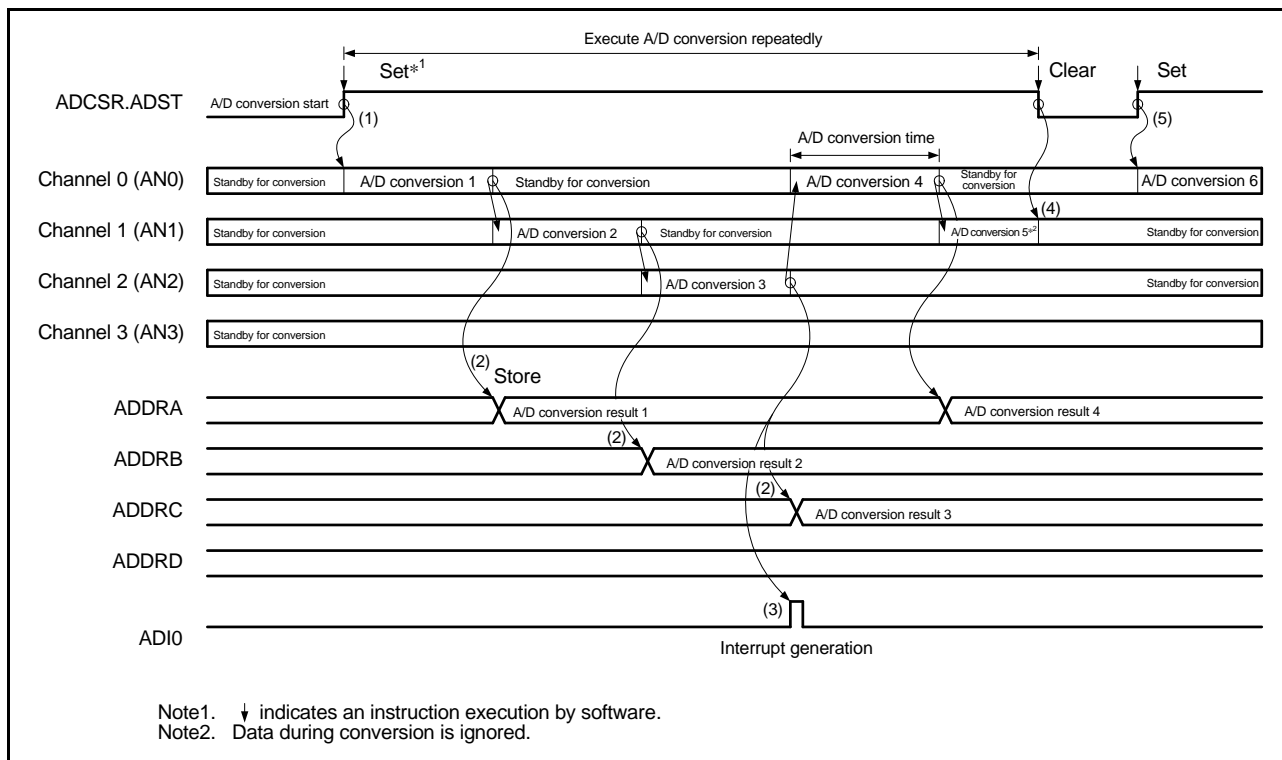


Figure 35.4 Example of A/D Converter Operation (Continuous Scan Mode)

35.3.2.2 One-Cycle Scan Mode

In one-cycle scan mode, A/D conversion is to be performed for one cycle on the analog inputs of the specified channels as below.

1. When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, MTU, TMR, or an external trigger input, A/D conversion starts on the first channel in the specified channel group.
2. When A/D conversion for each channel is completed, the A/D conversion result is stored into the corresponding A/D data registers (ADDRn).
3. When A/D conversion of all selected channels is completed, if the ADIE bit in ADCSR is set to 1 (A/D interrupt enable by completing A/D conversion), an ADI interrupt request is generated.
4. The ADST bit remains at 1 during A/D conversion, and is automatically cleared to 0 when A/D conversion of all selected channels end. The A/D converter enters a wait state.

Figure 35.5 shows an example of A/D conversion when three channels (AN4 to AN6) are selected for analog input.

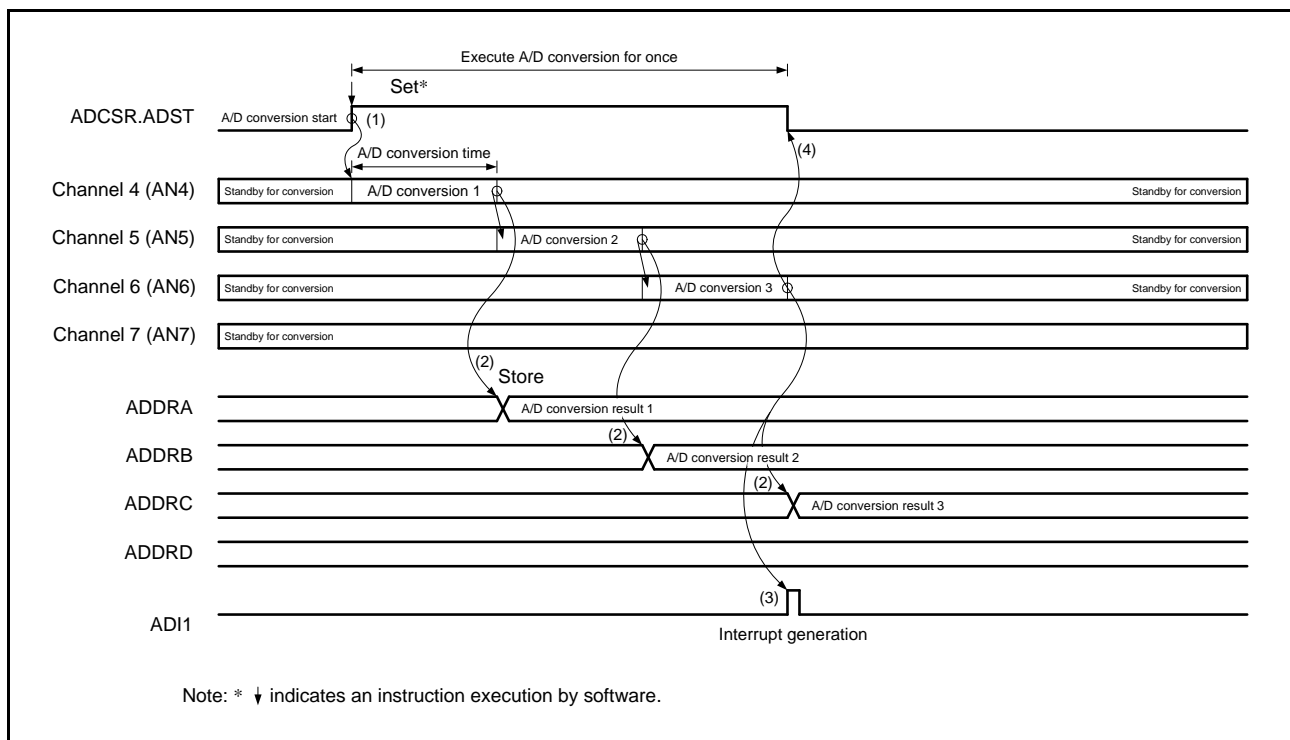


Figure 35.5 Example of A/D Converter Operation (One-Cycle Scan Mode)

35.3.3 Input Sampling and A/D Conversion Time

The A/D converter samples the analog input when the A/D conversion start delay time (t_D) passes after the conditions of A/D conversion start are generated by software, MTU, TMR, or an external trigger, then starts A/D conversion.

Figure 35.6 shows the A/D conversion timing.

The A/D conversion time (t_{CONV}) directly after the generation of the A/D conversion start condition includes the A/D conversion start delay time (t_D), the input sampling time (t_{SPL}), and the successive conversion time (t_{SAM}). The subsequent A/D conversion time (t_{CONV}) includes t_{SPL} and t_{SAM} .

The input sampling time (t_{SPL}) is the time charging of the input capacitance of the A/D converter's sample-and-hold circuit takes. If the impedance of the signal source is high and the sampling time is insufficient or the peripheral module clock (PCLK) is running at low speed, the sampling time can be adjusted by using the ADSSTR.

The successive conversion time (t_{SAM}) is fixed at 25 states of ADCLK.

Table 35.6 lists the sample of ADSSTR settings and Table 35.7 lists the A/D conversion time.

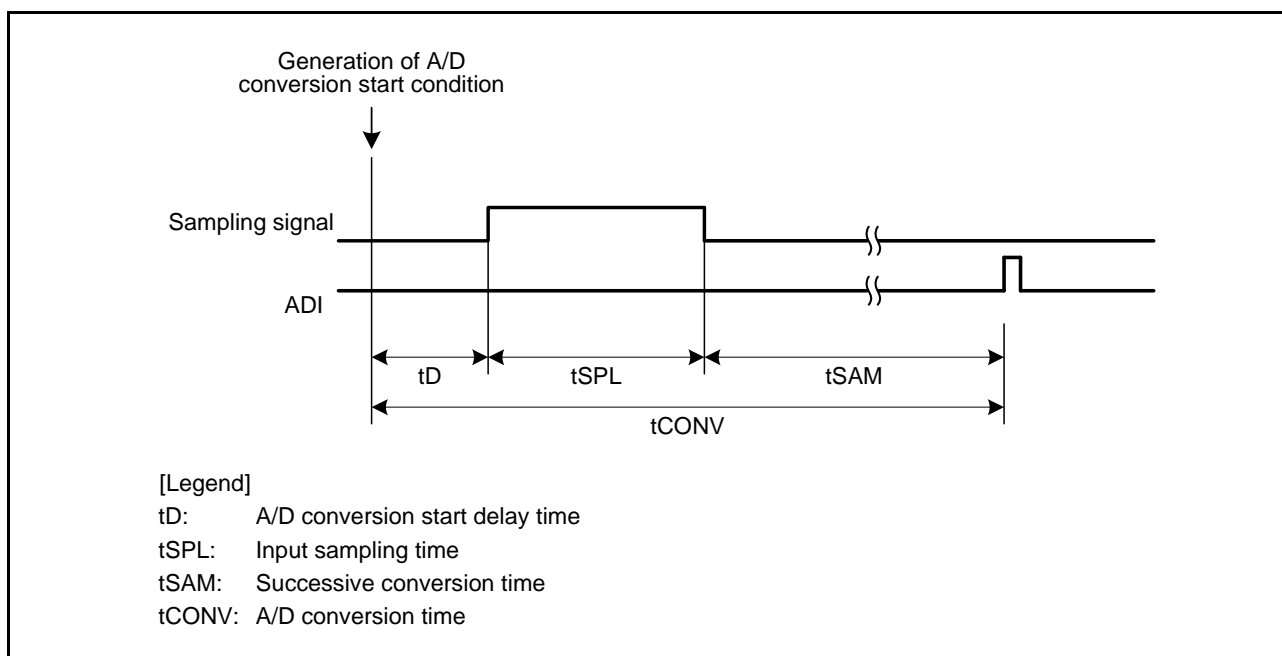


Figure 35.6 A/D Conversion Timing

Table 35.6 Example of ADSSTR Settings

Example of Setting	Setting Range	Sampling Time*
Standard (initial value)	19h	0.5 μ s (When PCLK = ADCLK = 50 MHz)
Analog input signal impedance is high and the sampling time may be insufficient	1Ah to FFh	Example: FFh 5.1 μ s (When PCLK = ADCLK = 50 MHz)
Input sampling time is less than the initial value when ADCLK is below 50 MHz	02h to 18h	Example: 14h 0.5 μ s (When PCLK = ADCLK = 40 MHz)

Note: * Set the sampling time $\geq 0.5 \mu$ s. Sampling time is shown as the formula below.

$$\text{Sampling time } (\mu\text{s}) = \frac{\text{Setting value of ADSSTR}}{\text{ADCLK (MHz)}}$$

Table 35.7 A/D Conversion Time

Item	Symbol	Formula	
		min	max
A/D conversion start delay time (1)	tD	$\frac{3}{\text{PCLK (MHz)}}$	$\frac{1}{\text{ADCLK (MHz)}} + \frac{4}{\text{PCLK (MHz)}}$
Input sampling time (2)	tSPL	$\frac{\text{Setting value of ADSSTR}}{\text{ADCLK (MHz)}}$	
Successive conversion time (3)	tSAM	$\frac{25}{\text{ADCLK (MHz)}}$	
A/D conversion time* ¹	tCONV	(1) + (2) + (3)	
A/D conversion time* ²	tCONV	(2) + (3)	

Notes: 1. A/D conversion time in single mode and scan mode (first round)

2. A/D conversion time in scan mode (after the second round)

The examples of the calculation of A/D conversion times are listed below.

When PCLK = ADCLK = 50 MHz, ADSSTR = 19h, and the conversion is the second round in scan mode,

$$\begin{aligned}
 \text{A/D conversion time (tCONV)} &= \text{ADSSTR/ADCLK} + 25/\text{ADCLK} \\
 &= 25/50 \text{ MHz} + 25/50 \text{ MHz} \\
 &= 0.5 \mu\text{s} + 0.5 \mu\text{s} \\
 &= 1.0 \mu\text{s}
 \end{aligned}$$

When PCLK = ADCLK = 40 MHz, ADSSTR = 14h, and conversion is the first (min.) round in scan mode,

$$\begin{aligned}
 \text{A/D conversion time (tCONV)} &= 3/\text{PCLK} + \text{ADSSTR/ADCLK} + 25/\text{ADCLK} \\
 &= 3/40 \text{ MHz} + 20/40 \text{ MHz} + 25/40 \text{ MHz} \\
 &= 0.075 \mu\text{s} + 0.5 \mu\text{s} + 0.625 \mu\text{s} \\
 &= 1.2 \mu\text{s}
 \end{aligned}$$

35.3.4 A/D Converter Activation by External Triggers

External trigger signals (ADTRG0# and ADTRG1#) are capable of starting A/D conversion by each of the units.

For unit 0, when the setting of the AD0.ADCR.TRGS[2:0] bits is 011b (specifying ADTRG0# as a trigger), a falling edge on the ADTRG0# pin leads to setting of the ADST (A/D conversion start) bit in AD0.ADCR to 1 and thus starts A/D conversion. Figure 35.7 shows the timing.

Take note that if the external trigger input is already at the low-level, selecting the external trigger may generate a falling edge in the internal signal and thus start A/D conversion.

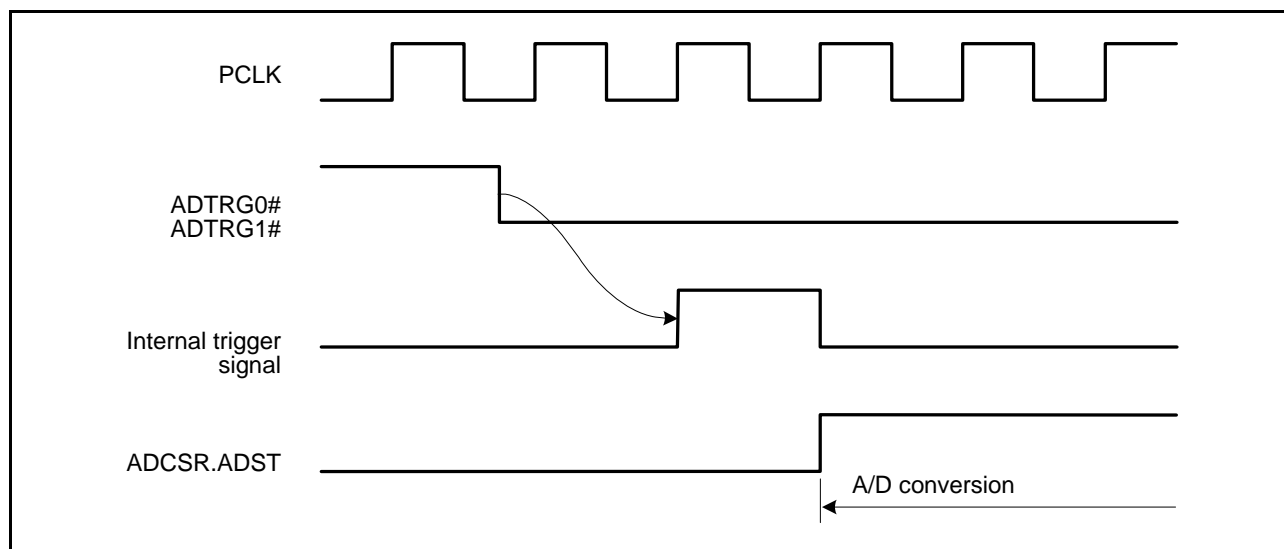


Figure 35.7 Timing of Activation by an External Trigger

35.3.5 A/D Converter Activation with TRG0AN_0 and TRG0BN_0 of MTU

The A/D converter unit 0 can be activated by generating a trigger signal TRG0AN_0 when an input capture or compare match occurs in TGRA of the MTU (unit 0) in channel 0.

The A/D converter unit 1 can also be activated by generating a trigger signal TRG0BN_0 when an input capture or compare match occurs in TGRB of the MTU (unit 0) in channel 0.

Figure 35.8 shows a connection of MTU (unit 0) TRG0AN_0 and TRG0BN_0 outputs and the A/D converter.

To start up the A/D converter when an input capture or compare match occurs in TGRA of the MTU (unit 0) in channel 0, AD0.ADCR.TRGS[2:0] bits should be set to 100b (to select the trigger signal TRG0AN_0).

To start up the A/D converter when an input capture or compare match occurs in TGRB of the MTU (unit 0) in channel 0, AD1.ADCR.TRGS[2:0] bits should be set to 100b (to select the trigger signal TRG0BN_0).

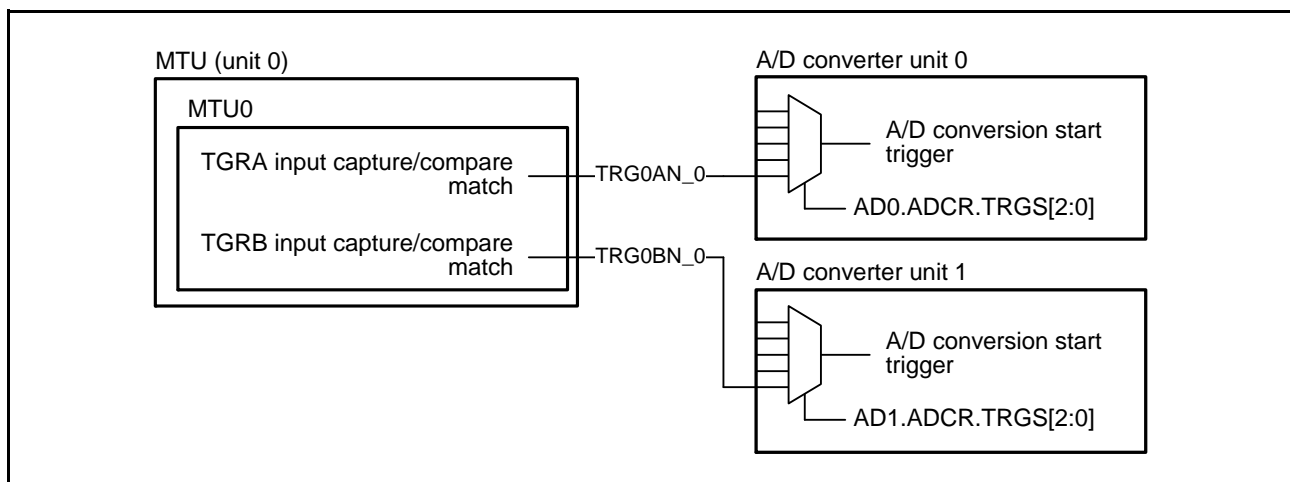


Figure 35.8 Connection of MTU (unit 0) TRG0AN_0 and TRG0BN_0 Outputs and A/D Converter

35.3.6 A/D Converter Activation with TRGAN_0 and TRGAN_1 of MTU

The A/D converter unit 0 and unit 1 can be activated by generating a trigger signal TRGAN_0 when an input capture or compare match occurs in TGRA of the MTU (unit 0) in channels 0 to 4 and a TCNT underflow (trough) occurs in channel 4 in complementary PWM mode. Likewise, the A/D converter unit 0 and unit 1 can be activated by generating a trigger signal TRGAN_1 when an input capture or compare match occurs in TGRA of the MTU (unit 1) in channels 6 to 10 and a TCNT underflow (trough) occurs in channel 10 in complementary PWM mode.

Figure 35.9 shows a connection of MTU (unit 0, unit 1) TRGAN_0 and TRGAN_1 outputs and the A/D converter.

To start up the A/D converter when an input capture or compare match occurs in TGRA of the MTU (unit 0) in channels 0 and 2, ADn.ADCR.TRGS[2:0] bits (n = 0, 1) of the A/D converter unit n (n = 0, 1) should be set to 001b (to select the trigger signal TRGAN_0), and the MTU0.TIER.TTGE bit and the MTU2.TIER.TTGE bit should be set to 1.

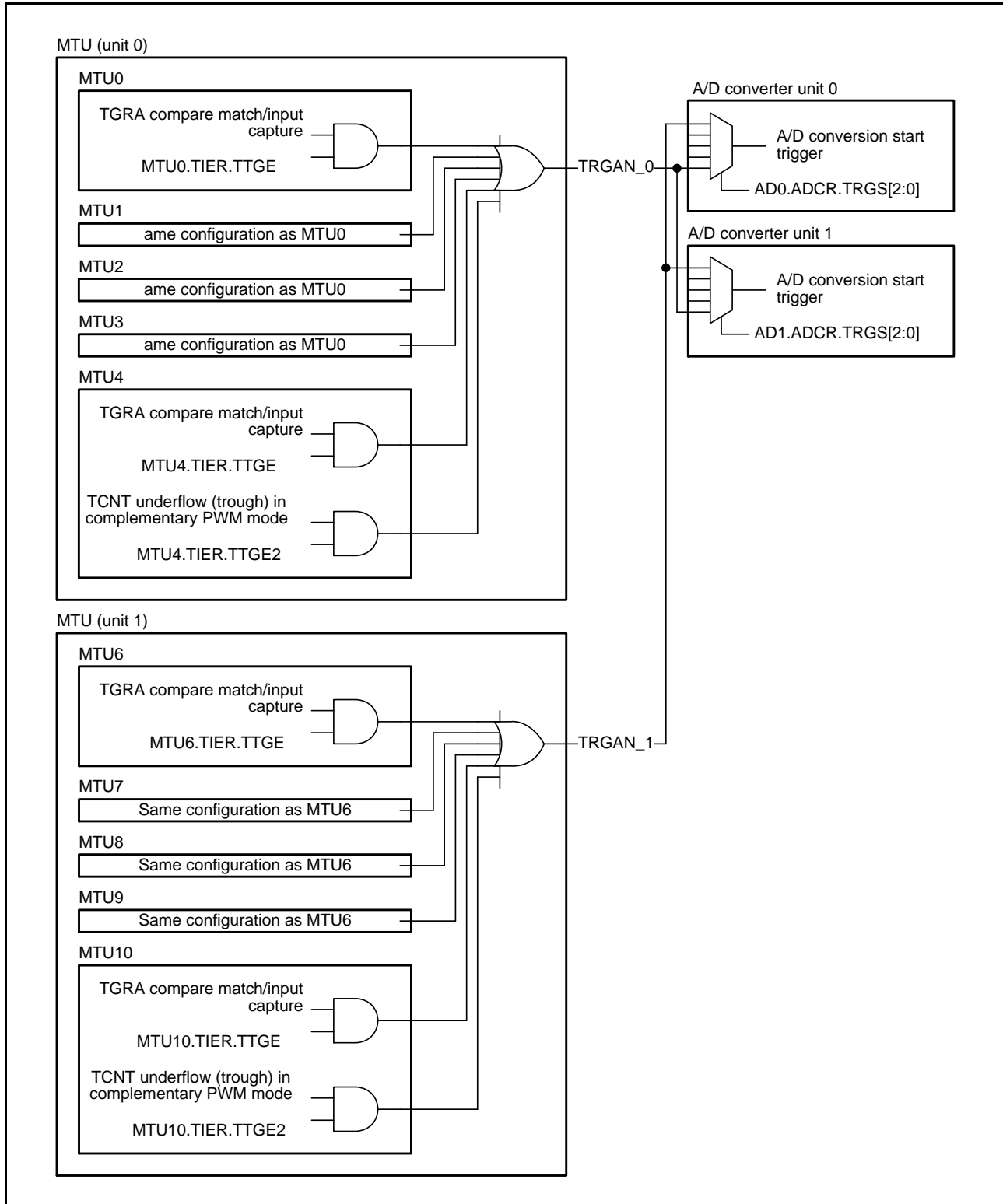


Figure 35.9 Connection of MTU (unit 0, unit 1) TRGAN_0 and TRGAN_1 Outputs and A/D Converter

35.3.7 A/D Converter Activation with TRG4ABN_0 and TRG4ABN_1 of MTU

The A/D converter unit 0 and unit 1 can be activated by generating a trigger signal TRG4ABN_0 when a compare match with the A/D converter start request delaying function of the MTU (unit 0) in channel 4 occurs. Likewise, the A/D converter unit 0 and unit 1 can be activated by generating a trigger signal TRG4ABN_1 when a compare match with the A/D converter start request delaying function of the MTU (unit 1) in channel 10 occurs.

Figure 35.10 shows a connection of MTU (unit 0, unit 1) TRG4ABN_0 and TRG4ABN_1 outputs and the A/D converter.

To start up the A/D converter by the A/D converter start request delaying function of the MTU (unit 0) in channel 4 when the TCNT count matches the TADCORA value, ADn.ADCR.TRGS[2:0] bits (n = 0, 1) of the A/D converter unit n (n = 0, 1) should be set to 110b (to select the trigger signal TRG4ABN_0), the cycle should be set to MTU4.TADCOBRA/B and MTU4.TADCORA/B, and the MTU4.TADCR.UT4AE bit should be set to 1.

For details on the A/D converter start request delaying function, see section 17.3.9, A/D Converter Start Request Delaying Function.

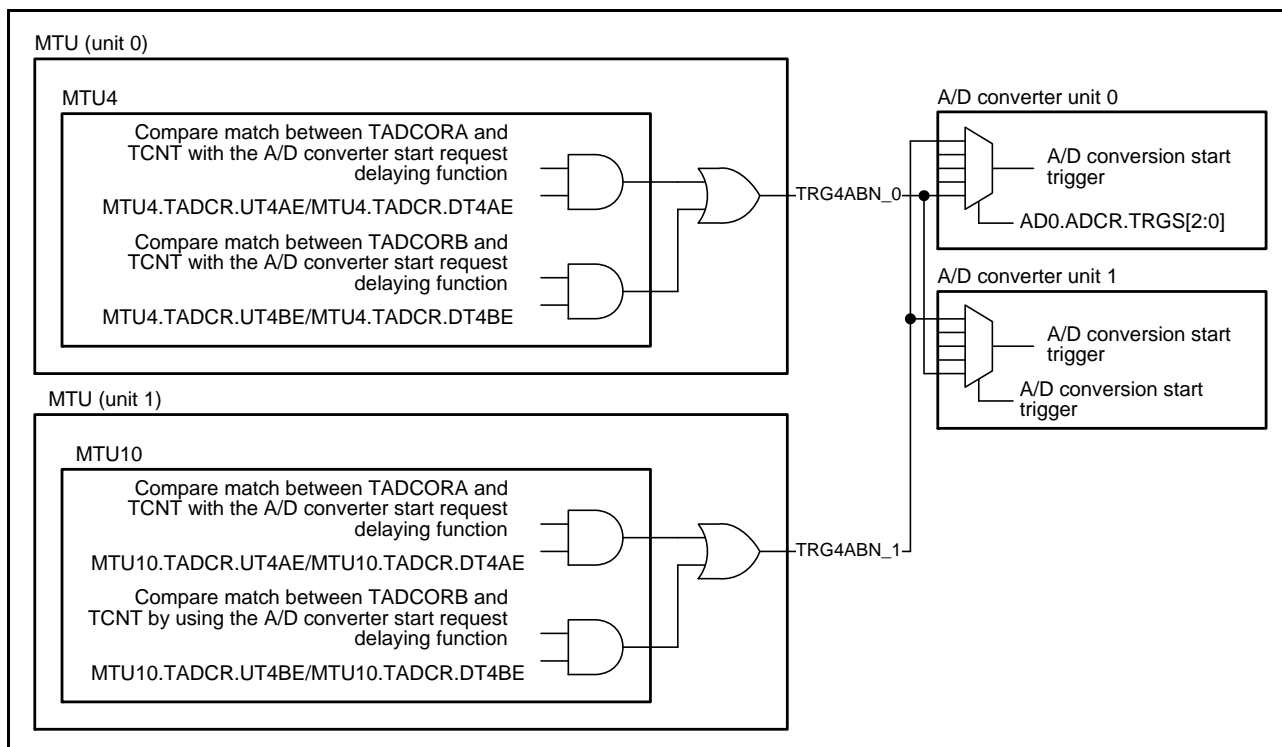


Figure 35.10 Connection of MTU (unit 0, unit 1) TRG4ABN_0 and TRG4ABN_1 Outputs and A/D Converter

35.3.8 A/D Converter Activation with TMTRG0AN_0 of TMR

The A/D converter unit 0 and unit 1 can be activated when a compare match (compare match A) occurs in TCORA of the TMR (unit 0) in channel 0.

Figure 35.11 shows a connection of TMR (unit 0) TMTRG0AN_0 output and the A/D converter.

To start up the A/D converter when a compare match (compare match A) occurs in TCORA of the TMR (unit 0) in channel 0, ADn.ADCR.TRGS[2:0] bits (n = 0, 1) of the A/D converter unit n (n = 0, 1) should be set to 010b (to select the trigger signal TMTRG0AN_0), and the TMR0.TCSR.ADTE bit should be set to 1.

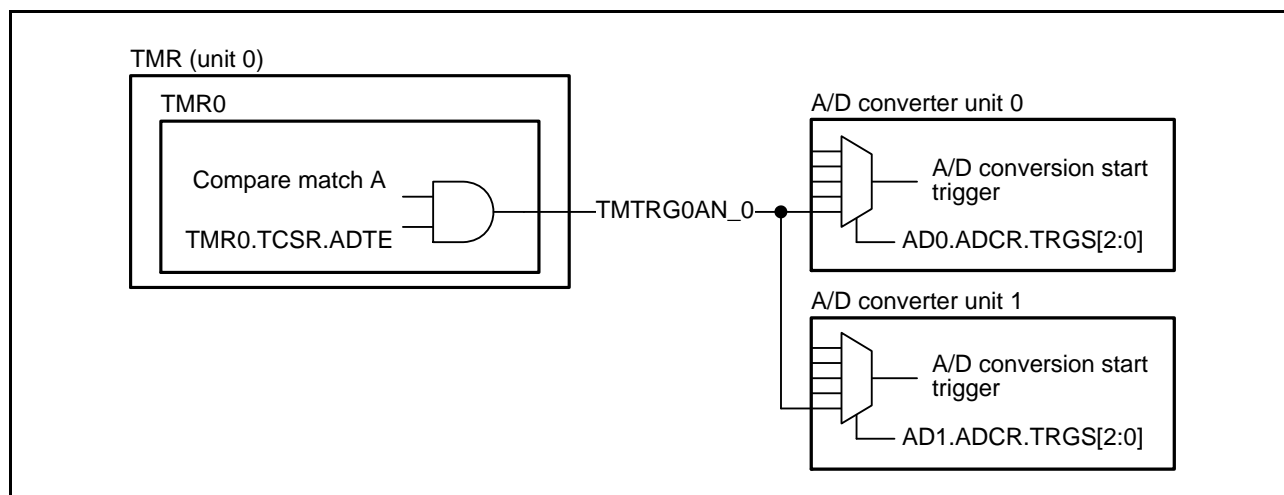


Figure 35.11 Connection of TMR (unit 0) TMTRG0AN_0 Output and A/D Converter

35.4 Interrupt Source

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion while the ADIE bit in ADCSR is set to 1 (after ADI interrupt is enabled by completing A/D conversion).

The data transfer controller (DTC) and DMA controller (DMACA) can be activated by an ADI interrupt. Having the converted data read by the DTC or DMACA in response to an ADI interrupt enables continuous conversion to be achieved without imposing a load on software.

Table 35.8 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Status Flag	DTC Activation	DMACA Activation
AD10	A/D conversion end	ICU.IR98.IR	Possible	Possible
AD11	A/D conversion end	ICU.IR99.IR	Possible	Possible

35.5 A/D Conversion Accuracy Definitions

The RX62N/RX621 Group's A/D conversion accuracy definitions are given below.

- Resolution
The number of A/D converter digital output codes
- Quantization error
The deviation inherent in the A/D converter, given by 1/2 LSB (see Figure 35.12)
- Offset error
The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value 000000000b (000h) to 000000001b (001h) (see Figure 35.13)
- Full-scale error
The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 111111110b (3FEh) to 111111111b (3FFh) (see Figure 35.13)
- Nonlinearity error
The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error (see Figure 35.13).
- Absolute accuracy
The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.

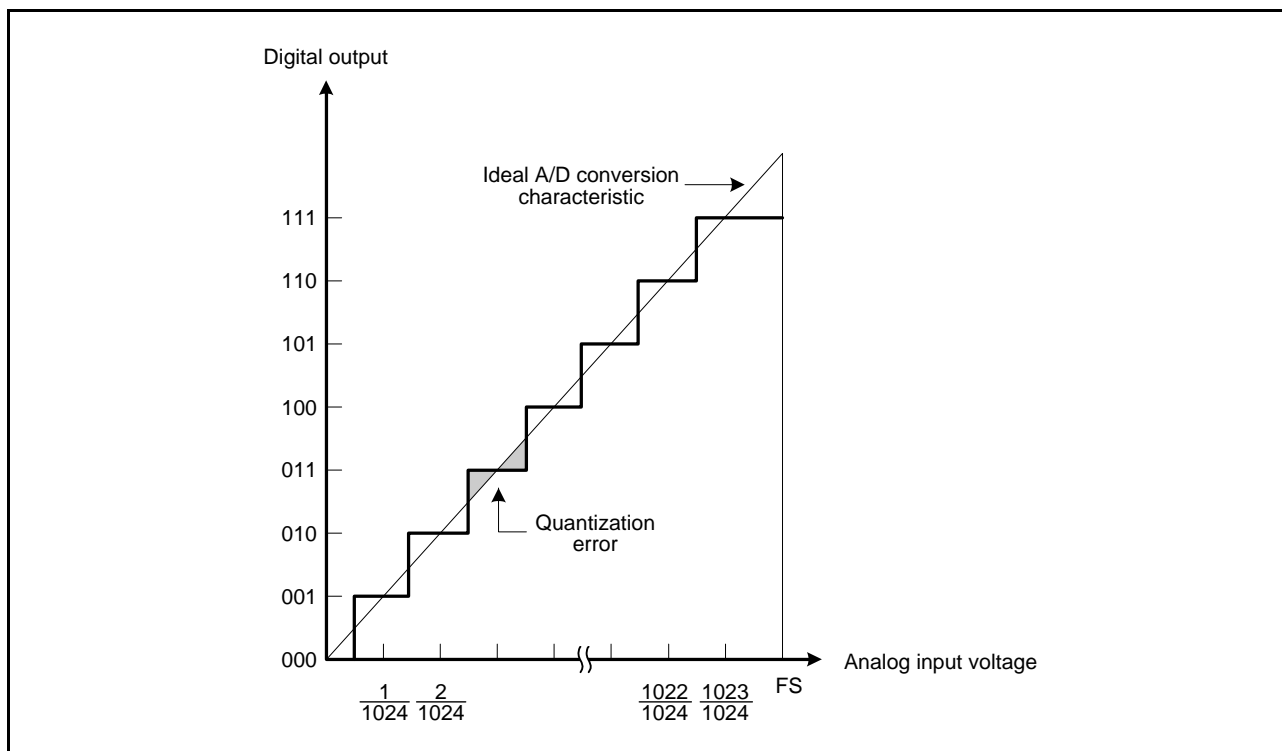


Figure 35.12 A/D Conversion Accuracy Definitions (1)

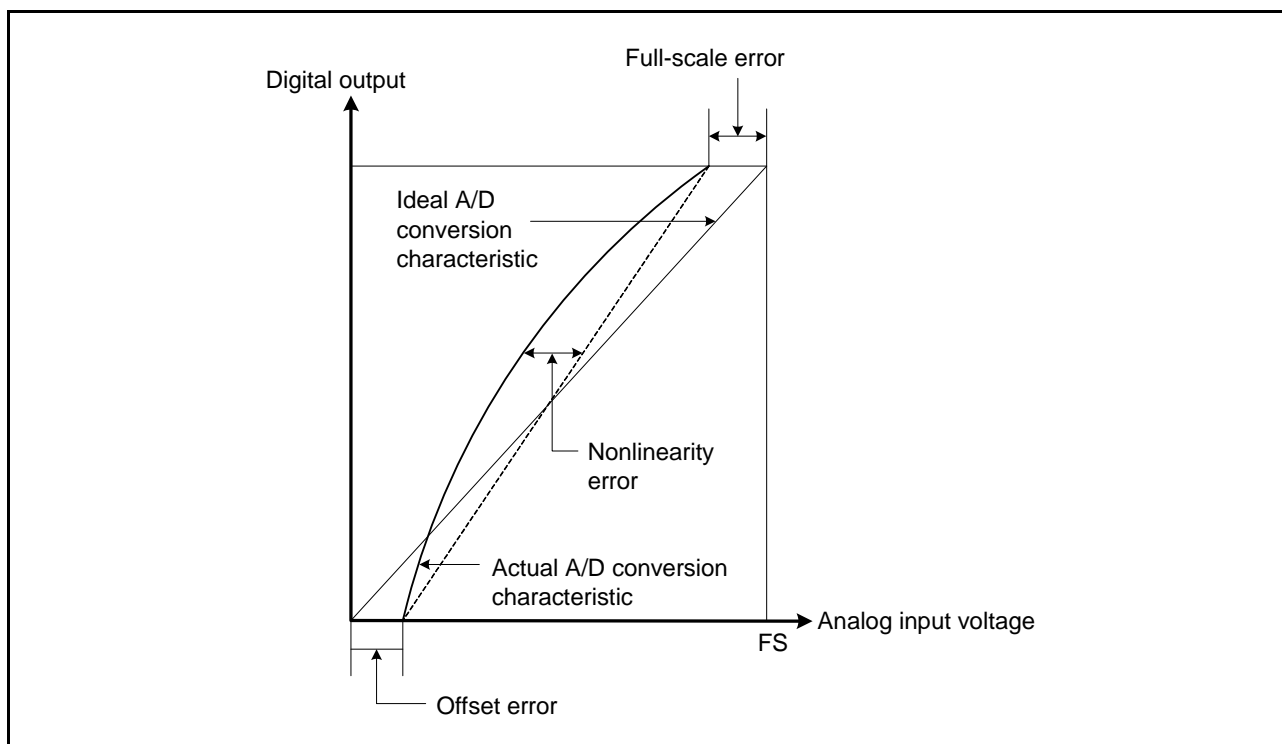


Figure 35.13 A/D Conversion Accuracy Definitions (2)

35.6 Usage Notes

35.6.1 Module Stop Function Setting

Operation of the A/D converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing the module stop state. For details, see section 9, Low Power Consumption.

35.6.2 Notes on Disabling A/D Conversion

To disable A/D conversion when an external trigger or timer has been selected as the condition for starting A/D conversion, set the ADCR.TRGS[2:0] bits to 000b to select the software trigger as the trigger to start A/D conversion, and then set the ADST bit in ADCSR to 0 (to stop A/D conversion).

35.6.3 Notes on Restarting A/D Conversion

Stopping analog input to the A/D converter by clearing the ADST bit in ADCSR to 0, requires one cycle of the ADCLK. If A/D conversion is to be restarted right after the ADST bit was set to 0, set the ADST bit to 1 allow A/D conversion to restart after one clock cycle has elapsed.

35.6.4 Notes on Entering Power-Down States

When the RX62N/RX621 Group enters the module stop state or software standby mode with A/D conversion enabled, the analog power supply current is the same as it is during A/D conversion. If the analog power supply current needs to be reduced in the module stop state or software standby mode, disable A/D conversion. To do so, set the ADST bit in ADCSR to 0, and allow time for disabling of the analog input to the A/D converter.

Follow the procedure given below to ensure that this time is secured.

- (1) Set the ADCR.TRGS[2:0] bits to 000b (software trigger).
- (2) Clear the ADCSR.ADST bit to 0.
- (3) Set the ADCR.CKS[1:0] bits to 11b (PCLK).
- (4) After confirming that the A/D converter has been disabled, place the LSI in the module stop state or software standby mode.

35.6.5 Permissible Impedance of Signal Sources

To realize high-speed conversion $1.0 \mu\text{s}$, the A/D conversion accuracy is guaranteed only when the impedance of the signal sources for analog input signals of the RX62N/RX621 Group circuit is less than or equal to $1.0 \text{ k}\Omega$. If a large external capacitance is provided in the case of conversion in single mode, the input load becomes only the actual internal input resistance ($6.5 \text{ k}\Omega$), so the impedance of the signal source becomes insufficient. However, since the circuit has become a low-pass filter, the rapid and large fluctuations in the analog signal produced by differentiation (for example, greater than $5 \text{ mV}/\mu\text{s}$) become impossible to track (Figure 35.14). Include a low-impedance buffer if high-rate analog signals are to be converted or conversion is to be in scan mode.

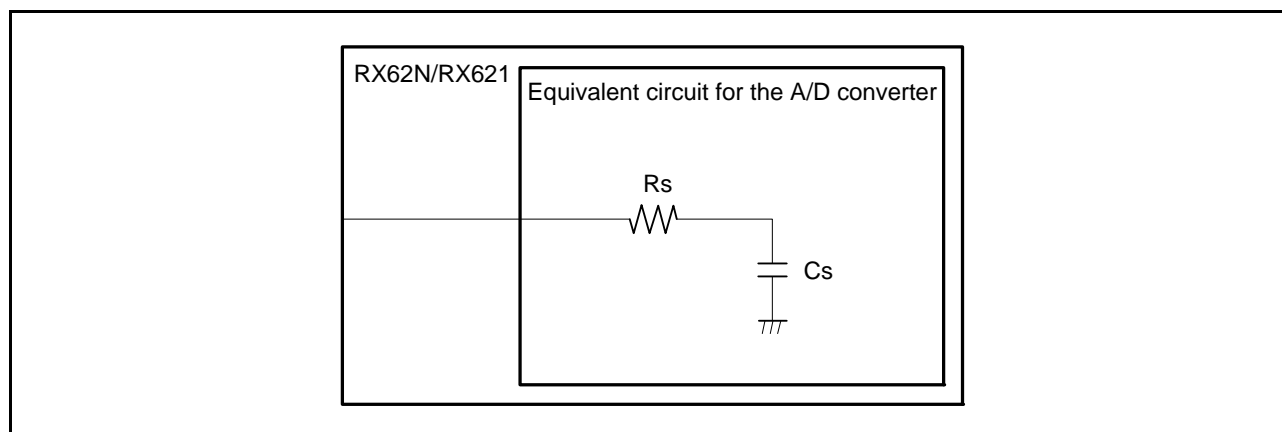


Figure 35.14 Equivalent Circuit for the Internal Circuit of Analog Input Pins

Table 35.9 Specifications of Analog Pins

Item		Min.	Max.	Unit
Permissible signal-source impedance		—	1.0	k Ω
Values in the equivalent circuits for the internal circuits of pins	Rs	—	6.5	k Ω
	Cs	—	6.0	pF

35.6.6 Factors Affecting Absolute Accuracy

Including a capacitor introduces ground coupling. A noisy ground can have a bad effect on absolute accuracy, so be sure to connect the VREFL pins and so on to an electrically stable ground.

Furthermore, a mounted filter circuit can interfere with digital-signal lines on the board, so take care to ensure that the design does not set up an antenna.

35.6.7 Ranges of Settings for Analog Power Supply and Other Pins

Using this LSI circuit with voltages beyond the ranges given below can have a bad effect on LSI reliability.

- Range for the setting of analog input voltages
Keep voltages applied to analog input pins (ANn pins) within the range defined by $VREFL \leq VAN \leq VREFH$.
- Relations between AVCC and VREFL, and between VCC and VSS
Ensure that the relations between AVCC and VREFL and between VCC and VSS are $AVCC = VCC$ and $VREFL = VSS$.
To realize $AVCC = VCC$ and $VREFL = VSS$ at the points where the power-supply lines originate, set up closed loops with wiring runs between power-supply pins that are as short as possible and connect 0.1- μ F capacitors as shown in Figure 35.15. Even if the A/D converters are not in use, ensure that $AVCC = VCC$ and $VREFL = VSS$ by making the same connections.
- Range for the setting of VREFH
Keep the reference voltage on the VREFH pin within the range defined by $VREFH \leq AVCC$.

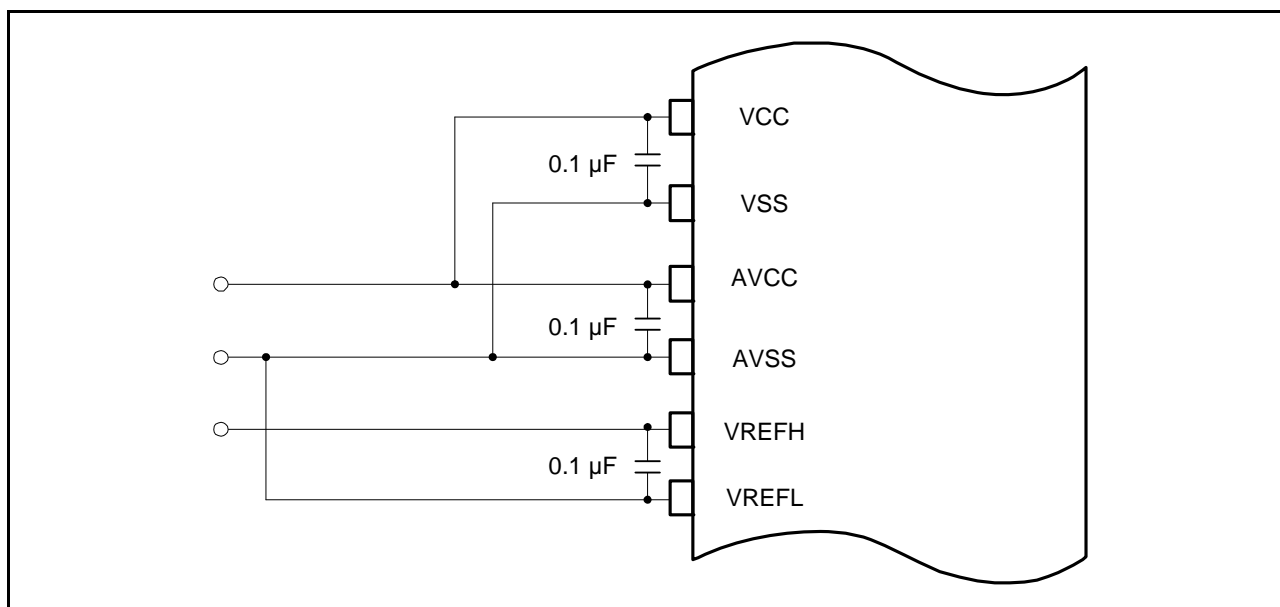


Figure 35.15 Example of Connections for Power Supply Pins

35.6.8 Point for Caution Regarding Board Design

As far as possible, separate the analog circuits from the digital circuits in board design. Furthermore, do not allow the wiring runs for a signal of a digital circuit and a signal of an analog circuit to cross or be in each other's vicinity. Inductive coupling leads to analog circuits operating incorrectly and has a bad effect on the results of analog conversion. Keep the signal lines for analog input pins (AN0 to AN7), the analog reference power supply pin (VREFH), the analog power-supply voltage (AVCC), and analog ground (AVSS) away from digital circuitry. Furthermore, connect the analog reference ground (VREFL) to the stabilized ground (VSS) on the board at a single point.

35.6.9 Point for Caution Regarding Countermeasures for Noise

To prevent destruction of the circuits for the analog input pins by abnormal voltages such as excessively large surges, connect capacitors as shown in Figure 35.16 between AVCC and AVSS, and between VREFH and VREFL; also connect suitable protective circuits to the analog input pins (AN0 to AN7).

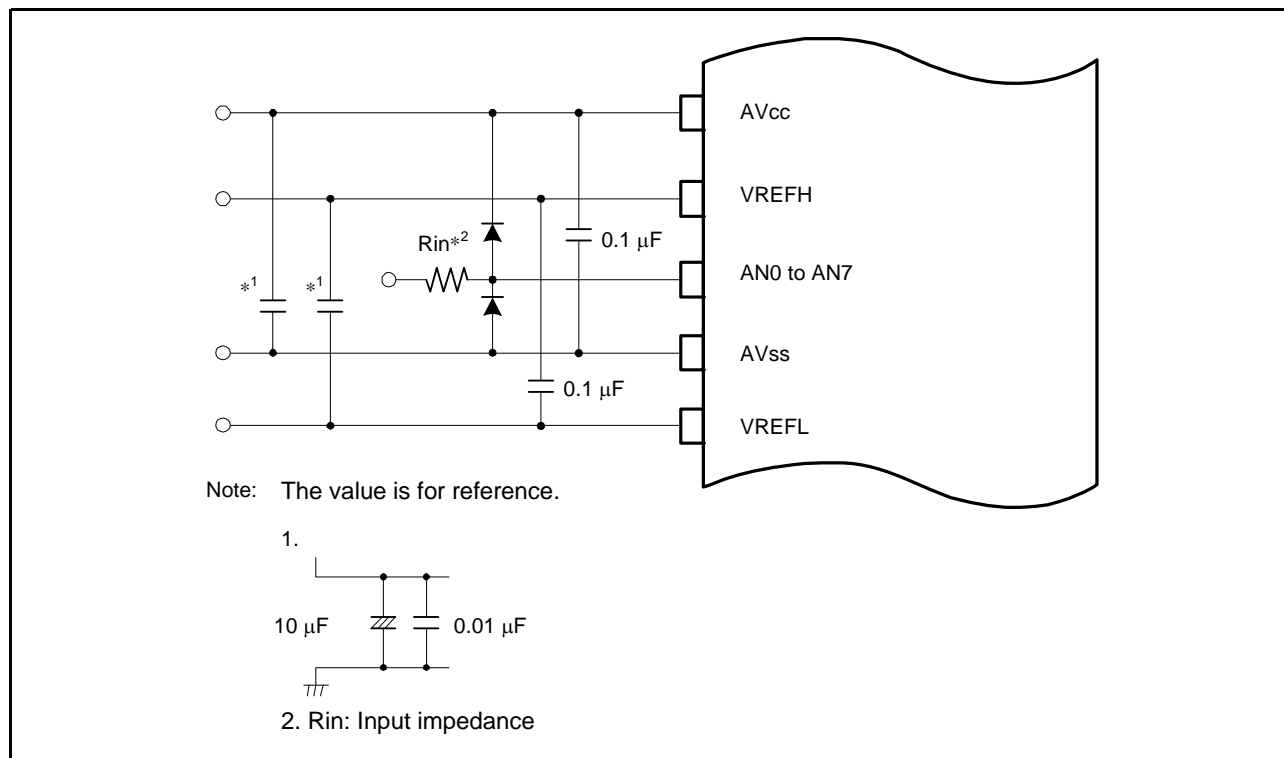


Figure 35.16 Example of a Protective Circuit for Analog Inputs

35.6.10 Realizing High-Speed Conversion

To realize high-speed conversion, connect external 0.1- μF capacitors between the analog input pins (AN0 to AN7) and VREFL. This is shown in Figure 35.17. However, to hide the impedance of the signal source due to the input capacitance of the sample-and-hold circuit of the A/D converter, the externally connected capacitors must be fully charged before the start of conversion.

Furthermore, when the voltages on the analog input pins fluctuate due to scanning and so on, so describing renewal of the charge of the externally connected capacitors, such full charge is beyond the scope of this description.

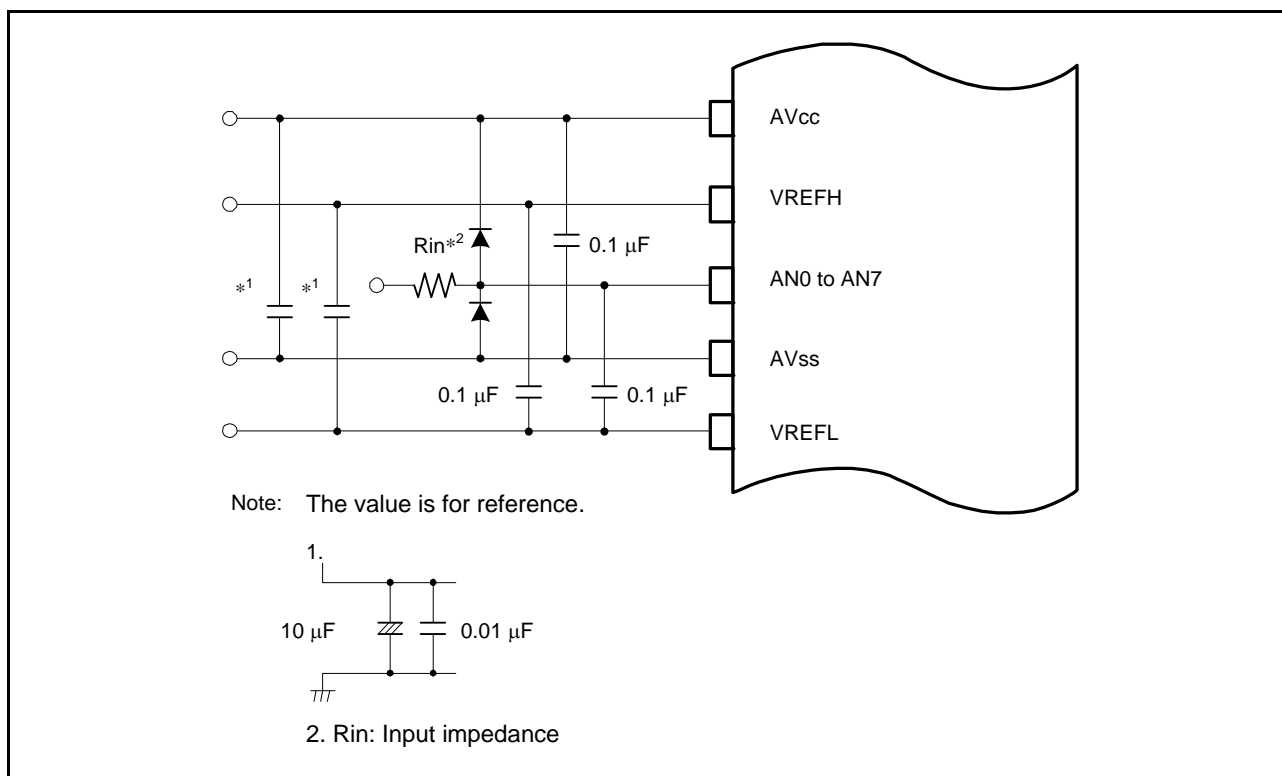


Figure 35.17 Example of an Externally Connected Capacitor for High-Speed Conversion

35.6.11 Selecting 12-Bit A/D Converter or 10-Bit A/D Converter

The 12-bit A/D converter or 10-bit A/D converter should be selected by the MSTPA23, MSTPA22, and MSTPA17 bits in module stop control register A (MSTPCRA).

Setting the MSTPCRA.MSTPA17 bit to 0 selects the 12-bit A/D converter.

Setting the MSTPCRA.MSTPA23 and MSTPCRA.MSTPA22 bits to 0 selects the 10-bit A/D converter.

If the MSTPCRA.MSTPA23, MSTPCRA.MSTPA22, and MSTPCRA.MSTPA17 bits are all set to 0 at the same time, the MSTPCRA.MSTPA17 bit setting takes priority and the MSTPCRA.MSTPA23 and MSTPA22 bit settings are ignored; that is, the 10-bit A/D converter is not selected in this case.

For details, see section 9.2.2, Module Stop Control Register A (MSTPCRA), in section 9, Low Power Consumption.

35.6.12 Notes on Using A/D Converter and D/A Converter Simultaneously

Since the A/D converter and the D/A converter use the same power supply, using them simultaneously may affect A/D conversion accuracy.

If the following registers are set during A/D conversion as below, it may make the accuracy worse by approximately 2 LSB.

- Data is written to the D/A data register m (DADRm) of the D/A converter during A/D conversion.
- Data is written to the D/A control register (DACR) during A/D conversion and when the DADRm register value of the D/A converter is not 00h.

If the above settings affect the conversion accuracy, take any of the following countermeasures.

(1) Average the A/D conversion results using a program.

Example of averaging: Perform A/D conversion four consecutive times for the same pin, and then calculate the average between two values other than the maximum and minimum values of the A/D conversion results.

(2) Discard the A/D conversion result during conversion.

(3) Write data to the DACR register of the D/A converter after setting the DADRm register value of the D/A converter to 00h.

36. D/A Converter

36.1 Overview

The RX62N/RX621 Group includes a two-channel of 10-bit D/A converter.

Table 36.1 lists the specifications of the D/A converter and Figure 36.1 shows a block diagram of the D/A converter.

Table 36.1 Specifications of D/A Converter

Item	Specifications
Resolution	10 bits
Output channels	Two channels
Power-down function	Module stop state can be set for each unit.

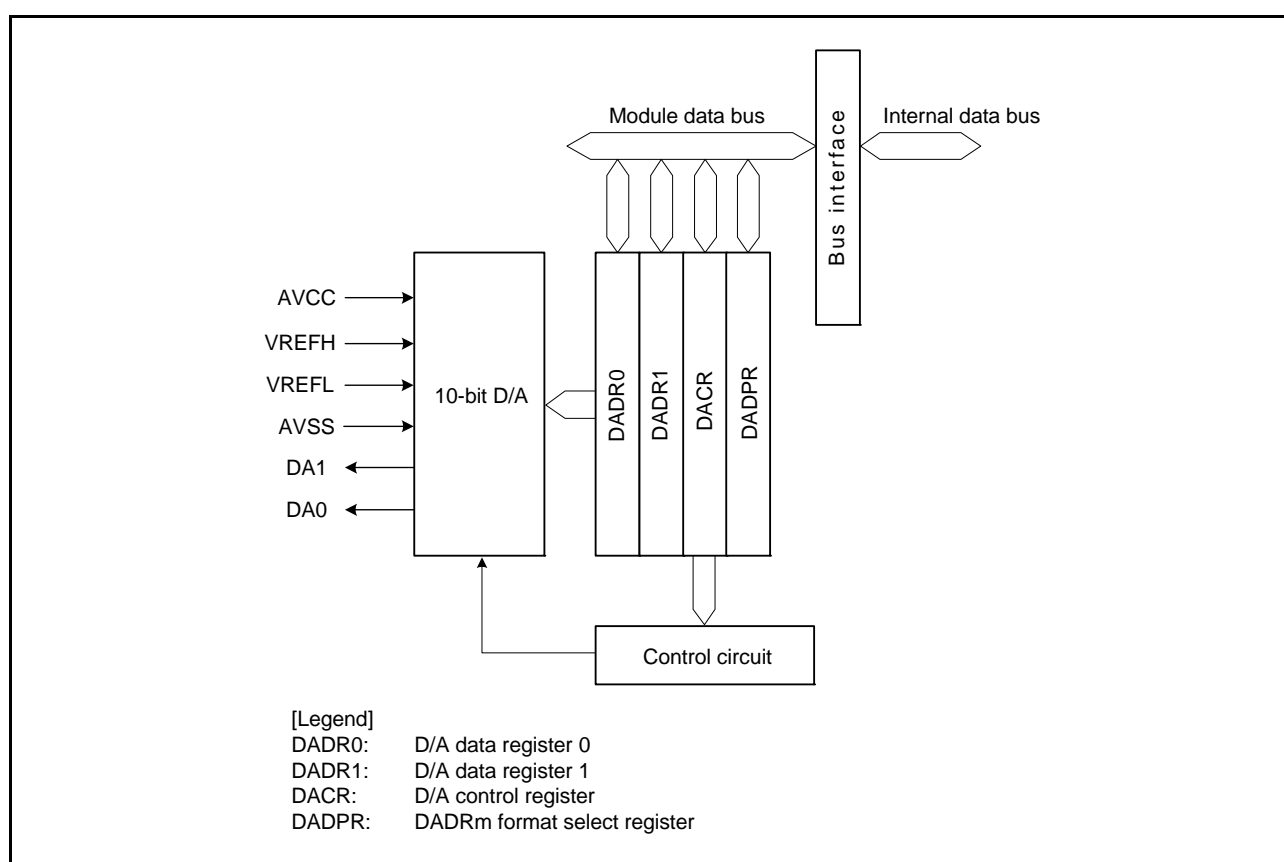


Figure 36.1 Block Diagram of D/A Converter

Table 36.2 lists the pin configuration of the D/A converter.

Table 36.2 Pin Configuration of D/A Converter

Pin Name	I/O	Function
AVCC	Input	Analog circuit power supply pin
AVSS	Input	Analog circuit ground pin
VREFH	Input	D/A converter reference voltage pin
VREFL	Input	D/A converter reference ground pin Connect this pin to the analog reference power supply (0 V).
DA0	Output	Channel 0 analog output pin
DA1	Output	Channel 1 analog output pin

36.2 Register Descriptions

Table 36.3 lists the registers of the D/A converter.

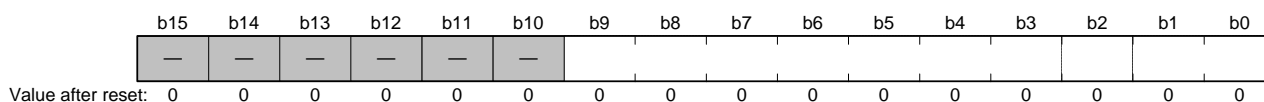
Table 36.3 Registers of D/A Converter

Register Name	Symbol	Value after Reset	Address	Access Size
D/A data register 0	DADR0	0000h	0008 80C0h	16
D/A data register 1	DADR1	0000h	0008 80C2h	16
D/A control register	DACR	1Fh	0008 80C4h	8
DADRm format select register	DADPR	00h	0008 80C5h	8

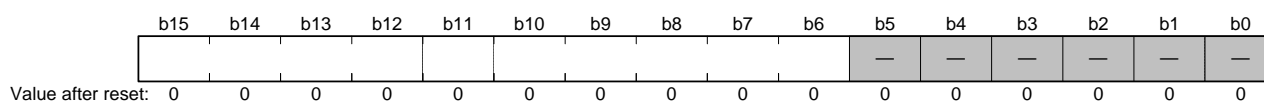
36.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Addresses: 0008 80C0h, DADR1 0008 80C2h

DADPR.DPSEL bit = 0 (Data padded at the LSB end)



DADPR.DPSEL bit = 1 (Data padded at the MSB end)



DADRm registers are 16-bit readable/writable registers, which store data to which D/A conversion is to be performed. Whenever an analog output is enabled, the values in DADRm are converted and output to the analog output pins.

10-bit data can be relocated by setting the DPSEL bit in DADPR.

Bits "—" are always read as 0. The write value should always be 0.

36.2.2 D/A Control Register (DACR)

Address: 0008 80C4h

b7	b6	b5	b4	b3	b2	b1	b0
DAOE1	DAOE0	DAE	—	—	—	—	—
Value after reset:	0	0	0	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 always be 1.	R/W
b5	DAE*1	D/A Enable	0: D/A conversion is independently controlled on channels 0 and 1. 1: D/A conversion on channels 0 and 1 is controlled as a single whole.	R/W
b6	DAOE0	D/A Output Enable 0	0: Analog output of channel 0 (DA0) is disabled. 1: D/A conversion of channel 0 is enabled. Analog output of channel 0 (DA0) is enabled.*2	R/W
b7	DAOE1	D/A Output Enable 1	0: Analog output of channel 1 (DA1) is disabled. 1: D/A conversion of channel 1 is enabled. Analog output of channel 1 (DA1) is enabled.*2	R/W

Note 1. This bit controls D/A conversion in combination with the DAOE0 and DAOE1 bits. The DAOE0 and DAOE1 bits control output of the results of conversion. For details, see Table 36.4.

Note 2. Set the Bm bit (m = 3, 5) in PORT0.DDR for pins used as analog outputs and the corresponding Bm bit (m = 3, 5) in PORT0.ICR to 0. For details, see section 17, I/O Ports.

Table 36.4 Controls of D/A Conversion

b5	b7	b6	Description
DAE	DAOE1	DAOE0	
0	0	0	D/A conversion is disabled.
		1	D/A conversion of channel 0 is enabled. D/A conversion of channel 1 is disabled. Analog output of channel 0 (DA0) is enabled. Analog output of channel 1 (DA1) is disabled.
	1	0	D/A conversion of channel 0 is disabled. D/A conversion of channel 1 is enabled. Analog output of channel 0 (DA0) is disabled. Analog output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is enabled.
1	0	0	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is disabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channel 0 (DA0) is enabled. Analog output of channel 1 (DA1) is disabled.
	1	0	D/A conversion of channels 0 and 1 is enabled. Analog output of channel 0 (DA0) is disabled. Analog output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is enabled.

DACR controls the operation of the D/A converter.

DAE Bit (D/A Enable)

The DAE bit controls D/A conversion in combination with the DAOE0 and DAOE1 bits.

When the DAE bit is 0, D/A conversion is independently controlled on channels 0 and 1. When the DAE bit is 1, D/A conversion on channels 0 and 1 is controlled as a single whole. The DAOE0 and DAOE1 bits control output of the results of conversion.

DAOE0 Bit (D/A Output Enable 0)

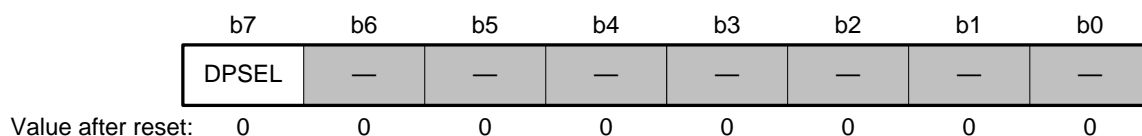
The DAOE0 bit controls the D/A conversion and analog output.

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: 0008 80C5h



Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	DPSEL	DADRm Format Select	0: D/A data register is flushed at the LSB end. 1: D/A data register is flushed at the MSB end.	R/W

DADPR selects the placement of data in the D/A data registers.

DPSEL Bit (DADRm Format Select)

The DPSEL bit selects whether data in the D/A data registers is flushed at the LSB or MSB end.

36.3 Operation

The 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. Write the data for conversion to DADR0.
2. Set the DAOE0 bit in DACR to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time tDCCONV has elapsed. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value is expressed by the following formula:

$$\frac{\text{Setting value of DADR0}}{1024} \times VREFH$$

3. If DADR0 is written to again, the conversion is immediately started. The conversion result is output after the conversion time tDCONV has elapsed.
4. If the DAOE0 bit is cleared to 0, analog output is disabled.

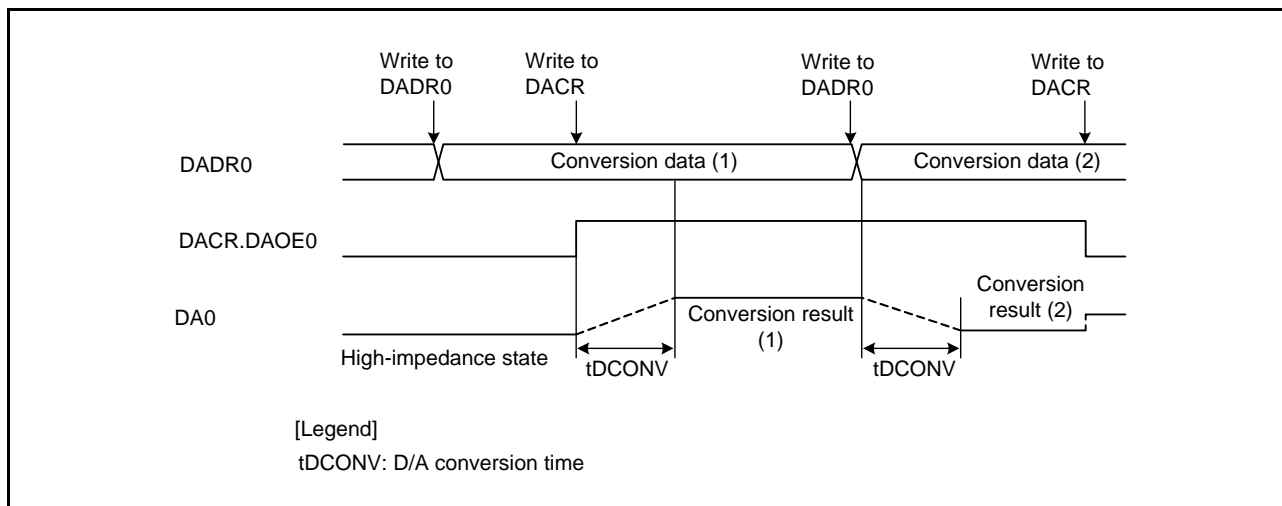


Figure 36.2 Example of D/A Converter Operation

36.4 Usage Notes

36.4.1 Module Stop Function Setting

Operation of the D/A converter can be disabled or enabled by using the module stop control register. The initial setting is for operation of the D/A converter to be halted. Register access is enabled by clearing the module stop state. For details, refer to section 9, Low Power Consumption.

36.4.2 Operation of the D/A Converter in Module Stop State

When the RX62N/RX621 Group enters the module stop state with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in the module stop state, disable D/A conversion by clearing the DAOE1, DAOE0, and DAE bits to 0.

36.4.3 Operation of the D/A Converter in Software Standby Mode

When the RX62N/RX621 Group enters software standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in software standby mode, disable D/A conversion by clearing the DAOE1, DAOE0, and DAE bits to 0.

36.4.4 Note on Entering Deep Software Standby Mode

When the RX62N/RX621 Group enters deep software standby mode with D/A conversion enabled, the outputs of the D/A converter are placed in a high impedance state.

37. RAM

The RX62N/RX621 Group has a high-speed static RAM.

37.1 Overview

Table 37.1 lists the specifications of the RAM.

Table 37.1 Specifications of the RAM

Item	Description
RAM capacity	96 Kbytes (RAM0: 64 Kbytes, RAM1: 32 Kbytes)
RAM address	RAM0: 0000 0000h to 0000 FFFFh RAM1: 0001 0000h to 0001 7FFFh
Access	<ul style="list-style-type: none"> Single-cycle access is possible for both reading and writing. Enabling or disabling of on-chip RAM is selectable.*
Data retention function	Data in RAM0 can be retained during periods in deep standby mode.
Power-down function	The module stop state is independently selectable for RAM0 and RAM1.

Note : * Selectable by the RAME bit in SYSCR1. For details on SYSCR1, see section 3.2.4, System Control Register 1 (SYSCR1).

37.2 Operation

37.2.1 Data Retention

The address space for on-chip RAM is divided into the RAM0 and RAM1 areas. The difference between the two is whether internal power can be supplied in deep software standby mode.

Whether or not the supply of internal power to RAM0 continues in deep software standby mode is selectable by the RAMCUTn bit (n = 2 to 0) in DPSBYCR.

If continuation of the supply of internal power is selected, data in RAM0 are retained during periods in deep software standby mode. The supply of internal power supply to RAM1 is halted at this time, so data are not retained in RAM 1. See section 9, Low Power Consumption, for details on the DPSBYCR.RAMCUTn (n = 0 to 2) bits.

37.2.2 Power-Down Function

Power consumption can be reduced by setting the module stop control register C (MSTPCRC) to stop supply of the clock signal to the on-chip RAM.

If the MSTPC0 bit in MSTPCRC is set to 1, supply of the clock signal to RAM0 is stopped. If the MSTPC1 bit in MSTPCRC is set to 1, supply of the clock signal to RAM1 is stopped.

The respective modules (RAM0 and RAM1) are thus placed in the module stop state by stopping supply of the clock signals. The initial value after a reset is for the RAM to be operational.

RAM is not accessible if it is in the module stop state. A transition to the module stop state should not be made while access to RAM is in progress.

For details on the MSTPCRC registers, see section 9, Low Power Consumption.

38. ROM (Flash Memory for Code Storage)

The RX62N/RX621 Group has two flash-memory modules: a maximum 512-Kbyte ROM for storing code and a 32-Kbyte data flash block for storing data.

This section covers the flash memory for code storage. For the data flash, see section 39, Data Flash Memory (Flash Memory for Data Storage).

38.1 Overview

Table 38.1 lists the specifications of the ROM, and Figure 38.1 show a block diagram of the ROM, data-flash memory (data flash), and related modules.

Table 38.1 Specifications of the ROM

Item	Specifications	
Memory capacity	<ul style="list-style-type: none"> User mat: 512 Kbytes, 384 Kbytes, or 256 Kbytes* User boot mat: 16 Kbytes 	
High-speed reading	A read operation takes one cycle of ICLK	
Programming/erasing method	<ul style="list-style-type: none"> The chip incorporates a dedicated sequencer (FCU) for programming of the ROM. Programming and erasing the ROM are handled by issuing commands to the FCU. The ROM in the erased state can be read as FFFF FFFFh in 32-bit access. 	
BGO (background operation)	<ul style="list-style-type: none"> The CPU is able to execute program code from areas other than the ROM or data flash while the ROM is being programmed or erased. Execution of program code from the ROM is possible while the data flash memory is being programmed or erased. 	
Suspension and resumption	<ul style="list-style-type: none"> The CPU is able to execute program code from the ROM during suspension of programming or erasure. Programming and erasure of the ROM can be restarted (resumed) after suspension. 	
Units of programming and erasure	<ul style="list-style-type: none"> Unit of programming for the user mat and the user boot mat: 256 bytes. Unit of erasure for the user mat: 4 Kbytes (8 blocks), 16 Kbytes (30 blocks) Unit of erasure for the user boot mat: 16 Kbytes 	
On-board programming (three types)	Boot mode	<ul style="list-style-type: none"> The user mat and the user boot mat are programmable via the SCI. The bit rate for SCI transfer between the host and RX62N/RX621 is automatically adjusted.
	USB (user) boot mode	<ul style="list-style-type: none"> Booting up from the user boot mat and programming of the user mat The USB boot program is stored in the user boot mat at shipment; the user mat is programmable via the USB. Programming the user boot mat allows programming of the user mat via an arbitrary interface
	User program	Programming of the user mat under program control
Off-board programming	A PROM programmer can be used to program the user mat and the user boot mat.	
Protection	Software-controlled protection	The FENTRYR.FENTRY0, FWEPROR.FLWE[1:0], and lock bits can be used to prevent unintentional programming.
	Error protection	Prevention of further programming or erasure after the detection of abnormal operations during programming or erasure
Times for programming and erasure, durability (number of times reprogramming is possible)	See section 41, Electrical Characteristics.	

Note: * Each product has different ROM sizes.

Product Code	ROM Size	ROM Addresses
R5F562x8	512 Kbytes	FFF8 0000h to FFFF FFFFh
R5F562x7	384 Kbytes	FFFA 0000h to FFFF FFFFh
R5F562x6	256 Kbytes	FFFC 0000h to FFFF FFFFh

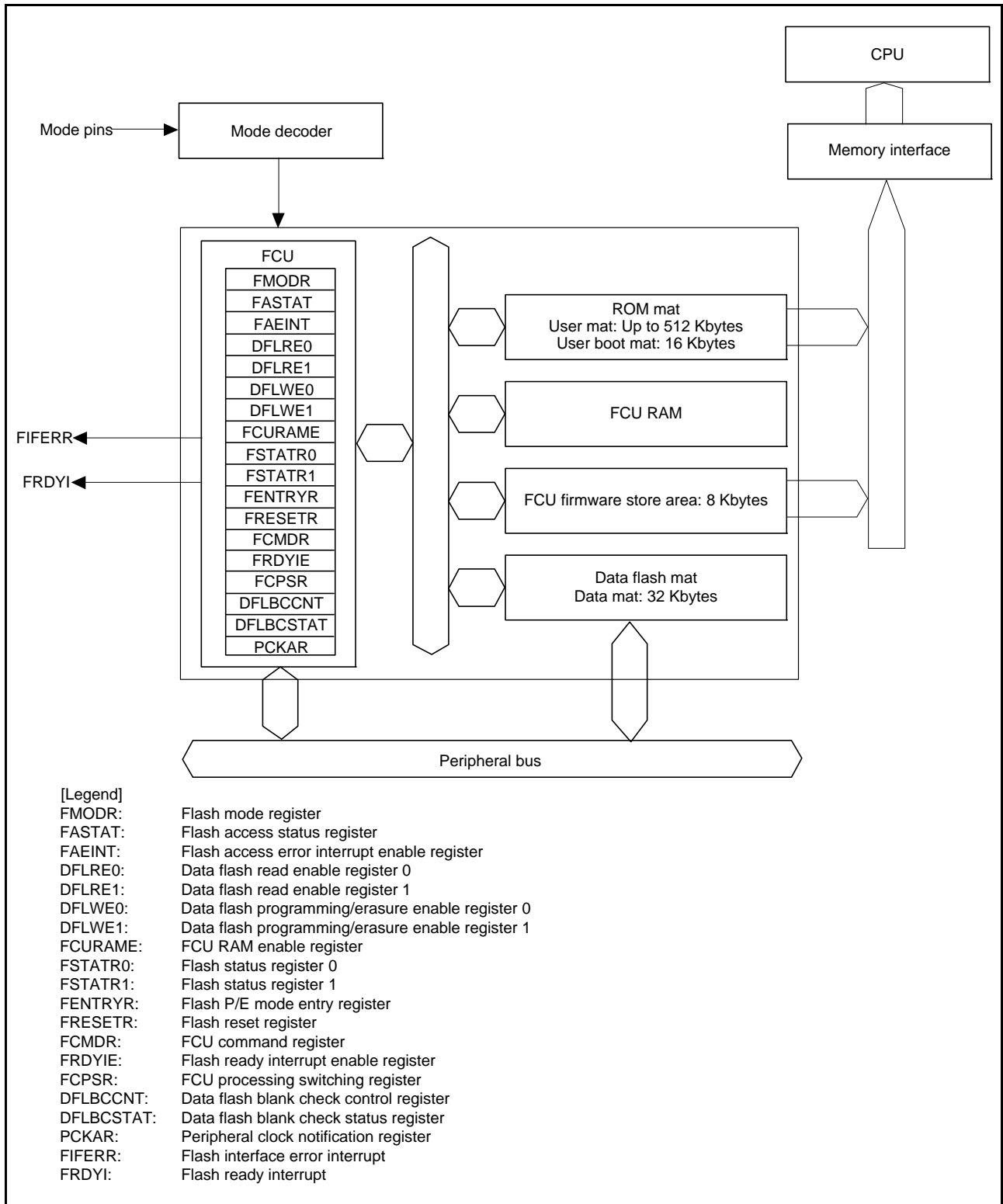


Figure 38.1 Block Diagram of ROM

Input and output pins associated with the ROM are listed in Table 38.2.

Table 38.2 Input and Output Pins Associated with the ROM

Pin Name	I/O	Description
PF2/RxD1-B (176-pin version) P30/RxD1 (145-, 144-,100-, or 85-pin version)	Input	Used in boot mode to receive data via SCI (for host communications)
PF0/TxD1-B (176-pin version) P26/TxD1 (145-, 144-,100-, or 85-pin version)	Output	Used in boot mode to transmit data from SCI (for host communications)
MD1, MD0	Input	Operating mode settings for products of the RX62N/RX621 Groups
USB0_DP, USB0_DM	I/O	Input/output pins for USB data (for use in USB boot mode)
P16/USB0_VBUS	Input	Detection of connection and disconnection of USB cables (for use in USB boot mode)
P35	Input	Selection of USB bus-power mode or self-power mode (for use in USB boot mode)
P14/USB0_DPUPE-B	Output	D+ pull-up control (for use in USB boot mode)

38.2 Register Descriptions

Table 38.3 lists the registers related to ROM. Although some registers have bits related to data flash, this section deals only with the bits related to ROM. For details on the bits related to the data flash, see section 39.2, Register Descriptions in the data flash section.

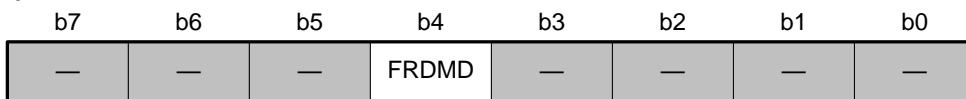
The registers related to the ROM are initialized by a reset.

Table 38.3 Registers Related to ROM

Register Name	Symbol	Value after Reset	Address	Access Size
Flash mode register	FMODR	00h	007F C402h	8
Flash access status register	FASTAT	00h	007F C410h	8
Flash access error interrupt enable register	FAEINT	9Bh	007F C411h	8
Flash ready interrupt enable register	FRDYIE	00h	007F C412h	8
FCU RAM enable register	FCURAME	0000h	007F C454h	16
Flash status register 0	FSTATR0	80h	007F FFB0h	8
Flash status register 1	FSTATR1	0xh	007F FFB1h	8
Flash P/E mode entry register	FENTRYR	0000h	007F FFB2h	16
Flash protection register	FPROTR	0000h	007F FFB4h	16
Flash reset register	FRESETR	0000h	007F FFB6h	16
FCU command register	FCMDR	FFFFh	007F FFBAh	16
FCU processing switching register	FCPSR	0000h	007F FFC8h	16
Flash P/E status register	FPESTAT	0000h	007F FFCCCh	16
Peripheral clock notification register	PCKAR	0000h	007F FFE8h	16
Flash write erase protection register	FWEPROR	02h	0008 C289h	8

38.2.1 Flash Mode Register (FMODR)

Address: 007F C402h



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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	FRDMD	FCU Read Mode Select	0: Memory Area Read Method Memory area read mode is set to read a lock bit of ROM in ROM lock bit read mode. 1: Register Read Method Register read mode is set to read a lock bit of ROM using the lock bit read 2 command.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

FMODR is a register to specify the method for the reading of lock bits.

When on-chip ROM is disabled, the data read from FMODR is 00h and writing is disabled.

FMODR is initialized by a reset.

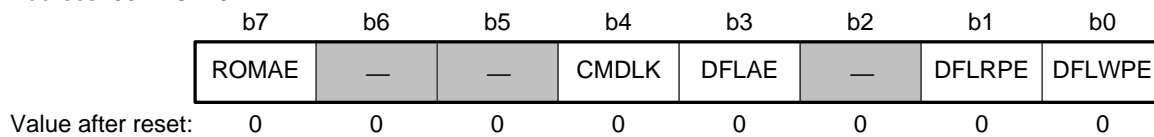
FRDMD Bit (FCU Read Mode Select)

This bit is used to specify the method for the reading of lock bits.

If the blank checking command for the data flash is to be used, this bit has to be set for the register read mode (see section 39, Data Flash Memory (Flash Memory for Data Storage)).

38.2.2 Flash Access Status Register (FASTAT)

Address: 007F C410h



Bit	Symbol	Bit Name	Description	R/W
b0	DFLWPE	Data Flash Programming/Erase Protection Violation	See section 39, Data Flash Memory (Flash Memory for Data Storage).	R/(W)*
b1	DFLRPE	Data Flash Read Protection Violation	See section 39, Data Flash Memory (Flash Memory for Data Storage).	R/(W)*
b2	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b3	DFLAE	Data Flash Access Violation	See section 39, Data Flash Memory (Flash Memory for Data Storage).	R/(W)*
b4	CMDLK	FCU Command Lock	0: FCU is not in the command-locked state 1: FCU is in the command-locked state	R
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	ROMAE	ROM Access Violation	0: No ROM access error 1: ROM access error	R/(W)*

Note : * Only 0 can be written after reading 1 to clear the flag.

FASTAT is a register to check if the access to the ROM/data flash is allowed.

When on-chip ROM is disabled, the data read from FASTAT is 00h and writing is disabled. When one of the bits in FASTAT is set to 1, the FCU is placed in the command-locked state (see section 38.8.2, Error Protection). To clear the command-locked state, a status clear command must be issued to the FCU after setting FASTAT to 10h.

FASTAT is initialized by a reset.

CMDLK Bit (FCU Command Lock)

This bit indicates that the FCU is in the command-locked state (see section 38.8.2, Error Protection).

[Setting condition]

- After the FCU detects an error and enters the command-locked state

[Clearing condition]

- After the FCU processes a status clear command under conditions where FASTAT is set to 10h

ROMAE Bit (ROM Access Violation)

This bit indicates whether a ROM access violation occurred.

When the ROMAE bit is set to 1, the ILGLERR bit in FSTATR0 is set to 1, placing the FCU in the command-locked state.

[Setting conditions]

- A read command is issued for ROM programming/erasure addresses 00F8 0000h to 00FF FFFFh when the FCU is in ROM P/E normal mode and the FENTRYR.FENTRY0 bit is set to 1.
- A command is issued for ROM programming/erasure addresses 00F8 0000h to 00FF FFFFh when the FENTRY0 bit is set to 0.
- A read command is issued for ROM read addresses FFF8 0000h to FFFF FFFFh when FENTRYR is set to the ROM P/E mode.

[Clearing condition]

- When 0 is written after reading 1

38.2.3 Flash Access Error Interrupt Enable Register (FAEINT)

Address: 007F C411h

b7	b6	b5	b4	b3	b2	b1	b0
ROMAEIE	—	—	CMDLKIE	DFLAEIE	—	DFLRPEIE	DFLWPEIE

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

Bit	Symbol	Bit Name	Description	R/W
b0	DFLWPEIE	Data Flash Programming/Erase Protection Violation Interrupt Enable	See section 39, Data Flash Memory (Flash Memory for Data Storage).	R/W
b1	DFLRPEIE	Data Flash Read Protection Violation Interrupt Enable	See section 39, Data Flash Memory (Flash Memory for Data Storage).	R/W
b2	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b3	DFLAEIE	Data Flash Access Violation Interrupt Enable	See section 39, Data Flash Memory (Flash Memory for Data Storage).	R/W
b4	CMDLKIE	FCU Command Lock Interrupt Enable	0: FIFERR interrupt requests disabled when the CMDLK bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the CMDLK bit in FASTAT is set to 1	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	ROMAEIE	ROM Access Violation Interrupt Enable	0: FIFERR interrupt requests disabled when the ROMAE bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the ROMAE bit in FASTAT is set to 1	R/W

FAEINT is a register to enable and disable the flash interface error interrupt (FIFERR).

When on-chip ROM is disabled, the data read from FAEINT is 00h and writing is disabled.

FAEINT is initialized by a reset.

CMDLKIE Bit (FCU Command Lock Interrupt Enable)

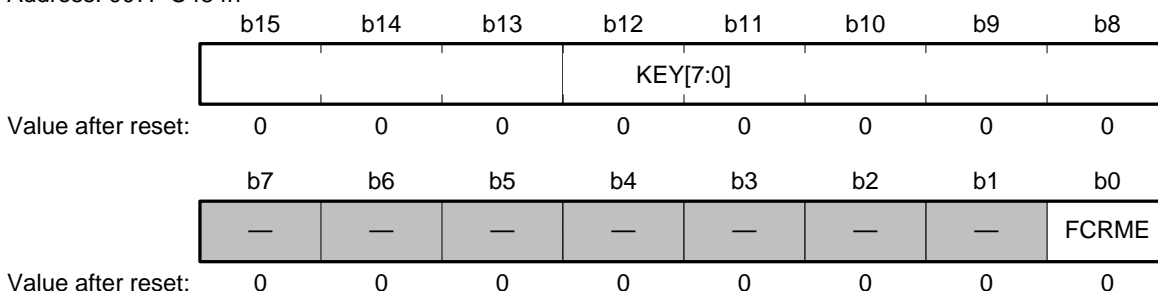
This bit is used to enable or disable FIFERR interrupt requests when an FCU command lock occurs and the CMDLK bit in FASTAT is set to 1.

ROMAEIE Bit (ROM Access Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when a ROM access violation occurs and the ROMAE bit in FASTAT is set to 1.

38.2.4 FCU RAM Enable Register (FCURAME)

Address: 007F C454h



Bit	Symbol	Bit Name	Description	R/W
b0	FCRME	FCU RAM Enable	0: Access to the FCU RAM disabled 1: Access to the FCU RAM enabled	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	This bit is used to enable or disable rewriting of the FCRME bit.	R/(W)*

Note : * Write data is not retained.

FCURAME is a register to enable and disable an access to the FCU RAM area.

Only specific values written to the upper byte in word access are valid. Data written to the upper byte is not retained.

When on-chip ROM is disabled, the data read from FCURAME is 00h and writing is disabled.

FCURAME is initialized by a reset.

FCRME Bit (FCU RAM Enable)

This bit is used to enable and disable an access to the FCU RAM.

Data written to the FCRME bit is valid only when it is written in word access and the KEY[7:0] bits are C4h. When programming data to the FCU RAM, set FENTRYR to 0000h and stop the FCU.

KEY[7:0] Bits (Key Code)

These bits are used to enable or disable rewriting of the FCRME bit.

Data written to the KEY[7:0] bits is not retained.

38.2.5 Flash Status Register 0 (FSTATR0)

Address: 007F FFB0h

b7	b6	b5	b4	b3	b2	b1	b0
FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	—	ERSSPD	PRGSPD

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

Bit	Symbol	Bit Name	Description	R/W
b0	PRGSPD	Programming Suspend Status	0: Other than the status described below 1: During programming suspend processing or programming suspended	R
b1	ERSSPD	Erase Suspend Status	0: Other than the status described below 1: When erasure suspend processing or erasure suspended	R
b2	—	Reserved	These bits are always read as 0 and cannot be modified.	R
b3	SUSRDY	Suspend Ready	0: P/E suspend commands cannot be received 1: P/E suspend commands can be received	R
b4	PRGERR	Programming Error	0: Programming terminates normally 1: An error occurs during programming	R
b5	ERSERR	Erase Error	0: Erasure terminates normally 1: An error occurs during erasure	R
b6	ILGLERR	Illegal Command Error	0: FCU detects no illegal command or ROM/data flash access 1: FCU detects an illegal command or ROM/data flash access	R
b7	FRDY	Flash Ready	0: During programming/erasure, During suspending programming/erasure, During the lock bit read 2 command processing, During the peripheral clock notification command processing During the blank check processing of data flash (See section 39, Data Flash Memory (Flash Memory for Data Storage)). 1: Processing described above is not performed	R

FSTATR0 is a register to check the FCU status.

When on-chip ROM is disabled, the data read from FSTATR0 is 00h.

FSTATR0 is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

PRGSPD Bit (Programming Suspend Status)

This bit is used to indicate that the FCU enters the programming suspend processing state or programming suspended state (see section 38.7, Suspending Operation).

[Setting condition]

- The FCU has initiated a write suspend command.

[Clearing condition]

- The FCU has accepted a resume command.

ERSSPD Bit (Erasure Suspend Status)

This bit is used to indicate that the FCU enters the erasure suspend processing state or erasure suspended state (see section 38.7, Suspending Operation).

[Setting condition]

- The FCU has initiated an erasure suspend command.

[Clearing condition]

- The FCU has accepted a resume command.

SUSRDY Bit (Suspend Ready Status)

This bit is used to indicate whether the FCU can receive a P/E suspend command.

[Setting condition]

- After starting programming/erasure process, the FCU enters a state in which P/E suspend commands can be received.

[Clearing conditions]

- The FCU has accepted a P/E suspend command.
- During programming/erasure process, the FCU enters the command-locked state.

PRGERR Bit (Programming Error)

This bit is used to indicate the result of the ROM/data flash programming process by the FCU.

When the PRGERR bit is set to 1, the FCU is placed in the command-locked state (see section 38.8.2, Error Protection).

[Setting condition]

- An error occurs during programming.
- A programming command is issued to areas protected by a lock bit.

[Clearing condition]

- After the FCU processes a status clear command

ERSERR Bit (Erasure Error)

This bit is used to indicate the result of the ROM/data flash erasure process by the FCU.

When the ERSERR bit is set to 1, the FCU is placed in the command-locked state (see section 38.8.2, Error Protection).

[Setting condition]

- An error occurs during erasure.
- A block erase command is issued to areas protected by a lock bit.

[Clearing condition]

- After the FCU processes a status clear command

ILGLERR Bit (Illegal Command Error)

This bit is used to indicate that the FCU detects any illegal command or ROM/data flash access.

When the ILGLERR bit is set to 1, the FCU is placed in the command-locked state (see section 38.8.2, Error Protection).

[Setting conditions]

- The FCU detects an illegal command.
- The FCU detects an illegal ROM/data flash access (one of the ROMAЕ, DFLAE, DFLRPE, and DFLWPE bits in FASTAT is 1).
- The setting of FENTRYR is invalid.

[Clearing condition]

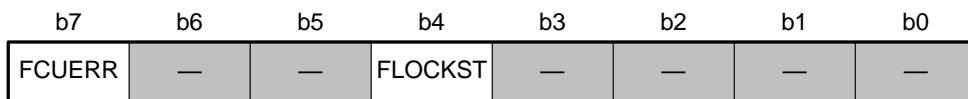
- After the FCU processes a status clear command under conditions where FASTAT is set to 10h

FRDY Bit (Flash Ready)

This bit is used to check the processing status of the FCU.

38.2.6 Flash Status Register 1 (FSTATR1)

Address: 007F FFB1h



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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	The read value is undefined and these bits cannot be modified.	R
b3, b2	—	Reserved	These bits are always read as 0 and cannot be modified.	R
b4	FLOCKST	Lock Bit Status	0: Protected 1: Not protected	R
b6, b5	—	Reserved	These bits are always read as 0 and cannot be modified.	R
b7	FCUERR	FCU Error	0: No error occurs in the FCU processing 1: An error occurs in the FCU processing	R

FSTATR1 is a register to check the FCU status.

When on-chip ROM is disabled, the data read from FSTATR1 is 00h.

FSTATR1 is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

FLOCKST Bit (Lock Bit Status)

This bit is to reflect the read data of a lock bit when using the lock bit read 2 command.

When the FRDY bit in FSTATR0 is set to 1 after a lock bit read 2 command is issued, valid data is stored in the FLOCKST bit. The value of the FLOCKST bit is retained until the completion of the next lock bit read 2 command.

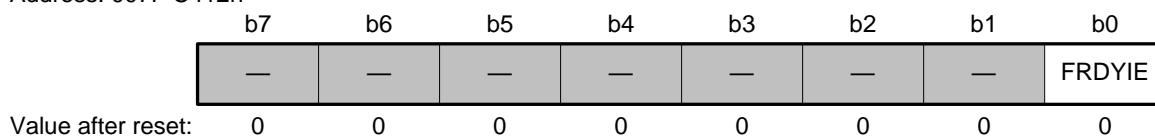
FCUERR Bit (FCU Error)

This bit is used to indicate that an error occurs in the FCU internal processing.

When the FCUERR bit is set to 1, set the FRESET bit in FRESETR to 1 to initialize the FCU. Additionally, recopy the FCU firmware from the FCU firmware area to the FCU RAM area.

38.2.7 Flash Ready Interrupt Enable Register (FRDYIE)

Address: 007F C412h



Bit	Symbol	Bit Name	Description	R/W
b0	FRDYIE	Flash Ready Interrupt Enable	0: FRDYI interrupt requests disabled 1: FRDYI interrupt requests enabled	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

FRDYIE is a register to enable and disable the flash ready interrupt (FRDYI) output.

When on-chip ROM is disabled, the data read from FRDYIE is 00h and writing is disabled.

FRDYIE is initialized by a reset.

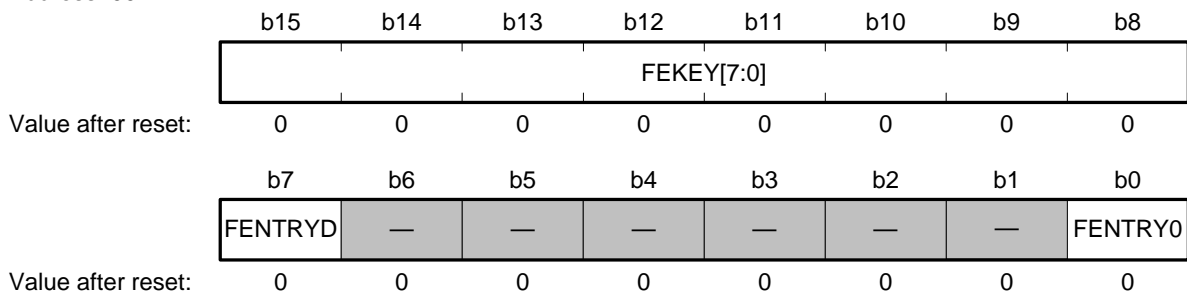
FRDYIE Bit (Flash Ready Interrupt Enable)

This bit is to enable/disable a FRDYI interrupt request when programming/erasure is completed.

If the FRDYIE bit is set to 1, a flash ready interrupt request (FRDYI) is generated when execution of the FCU command has completed (FSTATR0.FRDY bit changes from 0 to 1).

38.2.8 Flash P/E Mode Entry Register (FENTRYR)

Address: 007F FFB2h



Bit	Symbol	Bit Name	Description	R/W
b0	FENTRY0	ROM P/E Mode Entry 0	0: Products with 512, 384, or 256 Kbytes are in ROM read mode. 1: Products with 512, 384, or 256 Kbytes are in ROM P/E mode.	R/W
b6 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	FENTRYD	Data Flash P/E Mode Entry	See section 39, Data Flash Memory (Flash Memory for Data Storage).	R/W
b15 to b8	FEKEY[7:0]	Key Code	These bits enable or disable rewriting of the FENTRYD and FENTRY0 bits.	R/(W)*

Note : * Write data is not retained.

FENTRYR is a register to place the ROM/data flash in P/E mode.

To place the ROM/data flash in P/E mode so that the FCU can accept commands, either the FENTRYD or FENTRY0 bit must be set to 1. Note that if both bits are set to 1, the ILGLERR bit in FSTATR0 is set and the FCU enters the command-locked state.

Only specific values written to the upper byte in word access are valid. Any other writing causes the register to be initialized. Data written to the upper byte is not retained.

When on-chip ROM is disabled, the data read from FENTRYR is 0000h and writing is disabled.

FENTRYR is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

FENTRY0 Bit (ROM P/E Mode Entry 0)

This bit is used to place 512 Kbytes of ROM (read addresses: FFF8 0000h to FFFF FFFFh, programming/erasure addresses 00F8 0000h to 00FF FFFFh) in P/E mode.

[Writing-enable conditions (when all of the following conditions are met)]

- On-chip ROM is enabled.
- The FRDY bit in FSTATR0 is set to 1.
- AAh is written to the FEKEY[7:0] bits in word access.

[Setting condition]

- When the writing-enable conditions are met, FENTRYR is set to 0000h, and 1 is written to the FENTRY0 bit

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FEKEY[7:0] bits are other than AAh.
- When the writing-enable conditions are met, 0 is written to the FENTRY0 bit.
- When the writing-enable conditions are met and FENTRYR is other than 0000h, data is written to FENTRYR.

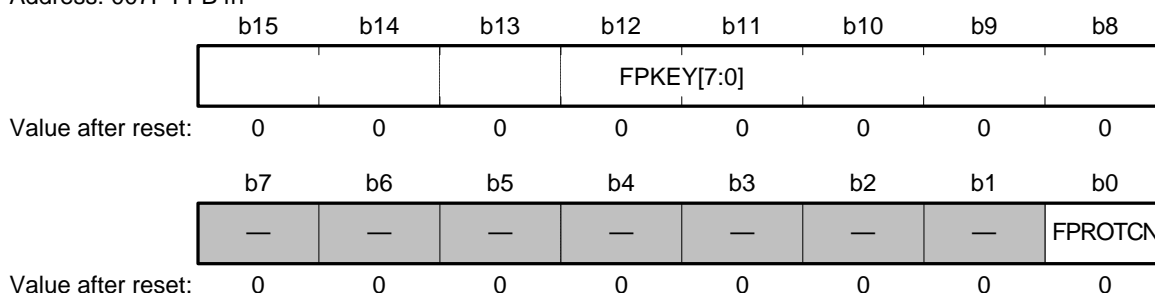
FEKEY[7:0] Bits (Key Code)

These bits enable or disable rewriting of the FENTRYD and FENTRY0 bits.

Data written to the FEKEY[7:0] bits is not retained.

38.2.9 Flash Protection Register (FPROTR)

Address: 007F FFB4h



Bit	Symbol	Bit Name	Description	R/W
b0	FPROTCN	Lock Bit Protection Cancel	0: Protection with a lock bit enabled 1: Protection with a lock bit disabled	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b15 to b8	FPKEY[7:0]	Key Code	These bits are used to enable or disable rewriting of the FPROTCN bit.	R/(W)*

Note : * Write data is not retained.

FPROTR is a register used to enable/disable the programming/erasure protection function with a lock bit.

Only specific values written to the upper byte in word access are valid. Any other writing causes the register to be initialized. Data written to the upper byte is not retained.

When on-chip ROM is disabled, the data read from FPROTR is 0000h and writing is disabled.

FPROTR is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

FPROTCN Bit (Lock Bit Protection Cancel)

This bit is used to enable/disable the programming/erasure protection with a lock bit.

[Setting condition]

- The FPKEY[7:0] bits are set to 55h, and the FPROTCN bit is set to 1 in word access when the value of FENTRYR is other than 0000h.

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FPKEY[7:0] bits are other than 55h.
- The FPKEY[7:0] bits are set to 55h, and the FPROTCN bit is set to 0 in word access.
- The value of FENTRYR is 0000h.

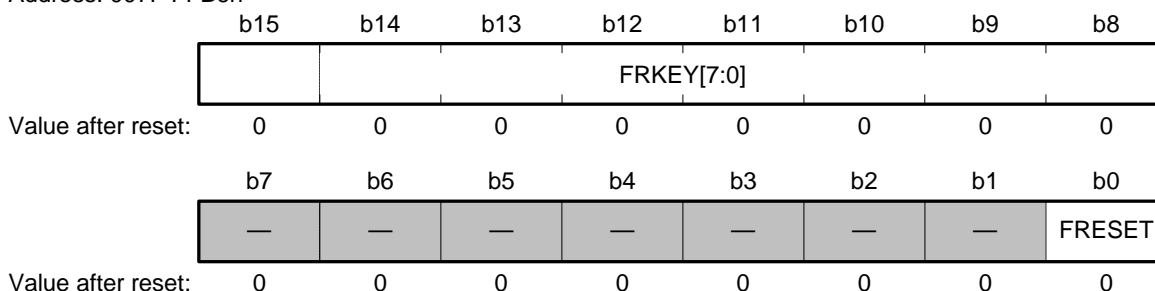
FPKEY[7:0] Bits (Key Code)

These bits are used to enable or disable rewriting of the FPROTCN bit.

Data written to the FPKEY[7:0] bits is not retained.

38.2.10 Flash Reset Register (FRESETR)

Address: 007F FFB6h



Bit	Symbol	Bit Name	Description	R/W
b0	FRESET	Flash Reset	0: FCU is not reset 1: FCU is reset	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b15 to b8	FRKEY[7:0]	Key Code	These bits are used to enable or disable rewriting of the FRESET bit.	R/(W)*

Note : * Write data is not retained.

FRESETR is a register to initialize the FCU.

Only specific values written to the upper byte in word access are valid. Data written to the upper byte is not retained.

When on-chip ROM is disabled, the data read from FRESETR is 0000h and writing is disabled.

FRESETR is initialized by a reset.

FRESET Bit (Flash Reset)

When the FRESET bit is set to 1, programming/erasure operations for the ROM/data flash are forcibly terminated, and the FCU is initialized.

High voltage is applied to the memory of the ROM/data flash during programming/erasure. To ensure time required for dropping the voltage applied to the memory, keep the FRESET bit set to 1 for tRESW2 (see section 41, Electrical Characteristics) when initializing the FCU. While the FRESET bit is kept 1, prohibit the ROM/data flash from being read. Additionally, when the FRESET bit is set to 1, the FCU commands cannot be used because FENTRYR is initialized.

Writing of the FRESET bit is enabled only in word access and when the FRKEY[7:0] bits are CCh.

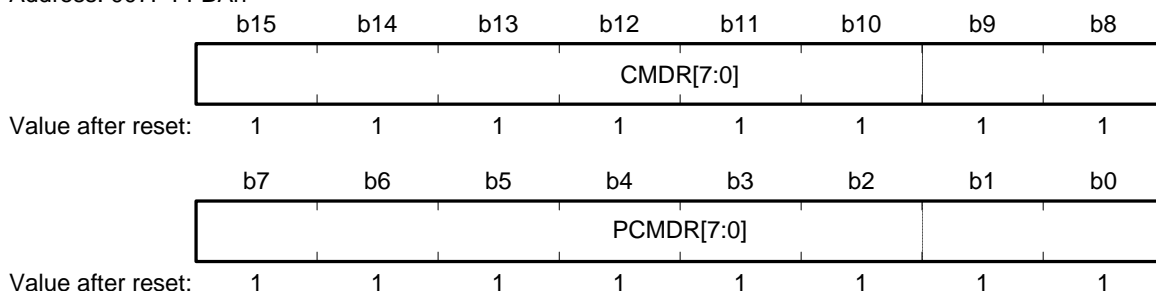
FRKEY[7:0] Bits (Key Code)

These bits are used to enable or disable rewriting of the FRESET bit.

Data written to the FRKEY[7:0] bits is not retained.

38.2.11 FCU Command Register (FCMDR)

Address: 007F FFBAh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PCMDR[7:0]	Precommand	Store the command immediately before the last command received by the FCU.	R
b15 to b8	CMDR[7:0]	Command	Store the last command received by the FCU.	R

FCMDR is a register to store commands received by the FCU.

When on-chip ROM is disabled, the data read from FCMDR is 0000h and writing is disabled.

FCMDR is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

Table 38.4 lists the states of FCMDR after receiving each command. For details on the blank check processing, see section 39.6, Programming and Erasing the Data Flash Memory.

Table 38.4 States of FCMDR after Receiving Each Command

Command	CMDR	PCMDR
P/E Normal mode transition	FFh	Previous command
Status read mode transition	70h	Previous command
Lock bit read mode transition (lock bit read 1)	71h	Previous command
Peripheral clock notification command	E9h	Previous command
Programming	E8h	Previous command
Block erase	D0h	20h
P/E suspend	B0h	Previous command
P/E resume	D0h	Previous command
Status register clear	50h	Previous command
Lock bit read 2 blank check	D0h	71h
Lock bit programming	D0h	77h

38.2.12 FCU Processing Switching Register (FCPSR)

Address: 007F FFC8h

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

Bit	Symbol	Bit Name	Description	R/W
b0	ESUSPMD	Erasure Suspend Mode	0: suspension priority mode 1: Erasure priority mode	R/W
b15 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

FCPSR is a register to select the method of suspending the FCU erasure processing.

When on-chip ROM is disabled, the data read from FCPSR is 0000h and writing is disabled.

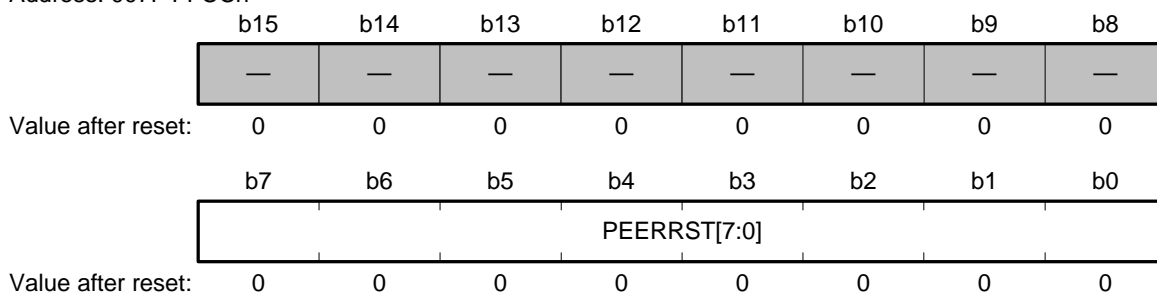
FCPSR is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

ESUSPMD Bit (Erasure Suspend Mode)

This bit is to select the erasure suspend mode for when a P/E suspend command is issued while the FCU executes the erasure processing for the ROM/data flash (see section 38.7, Suspending Operation).

38.2.13 Flash P/E Status Register (FPESTAT)

Address: 007F FFCh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PEERRST[7:0]	P/E Error Status	01h: Programming error against areas protected by a lock bit 02h: Programming error due to sources other than the lock bit protection 11h: Erasure error against areas protected by a lock bit 12h: Erasure error due to sources other than the lock bit protection (Values other than above are reserved)	R
b15 to b8	—	Reserved	These bits are always read as 0 and cannot be modified.	R

FPESTAT is a register to indicate the result of the programming/erasure processing for the ROM/data flash.

When on-chip ROM is disabled, the data read from FPESTAT is 0000h and writing is disabled.

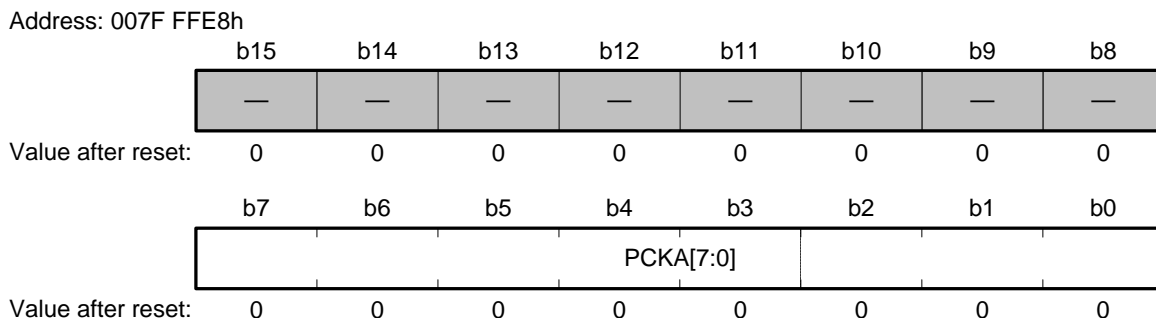
FPESTAT is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

PEERRST[7:0] Bits (P/E Error Status)

These bits are used to indicate the reason of an error that occurs during the programming/erasure processing for the ROM/data flash.

The value of the PEERRST[7:0] bits is valid only when the ERSERR or PRGERR bit in FSTATR0 is 1 and the FRDY bit in FSTATR0 is 1. The value of the reason of the past error is retained in the PEERRST[7:0] bits when the ERSERR and PRGERR bits are 0.

38.2.14 Peripheral Clock Notification Register (PCKAR)



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PCKA[7:0]	Peripheral Clock Notification	These bits are used to set the peripheral clock (PCLK) at the programming/erasure for the ROM/data flash.	R/W
b15 to b8	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

PCKAR is a register to notify the sequencer of the frequency setting data of the peripheral clock (PCLK) at the programming/erasure for the ROM/data flash. This setting is used to control programming and erasure times.

When on-chip ROM is disabled, the data read from PCKAR is 0000h and writing is disabled.

PCKAR is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

PCKA[7:0] Bits (Peripheral Clock Notification)

These bits are used to set the peripheral clock (PCLK) at the programming/erasure for the ROM/data flash.

Set the PCKA[7:0] bits to the PCLK frequency and issue a peripheral clock notification command before programming/erasure. Do not change the frequency during the programming/erasure processing for the ROM/data flash.

- Calculate the setting value as follows:
- Convert an operating frequency represented in MHz units to binary and write it to the PCKA[7:0] bits.
For example, when the operating frequency of the peripheral clock is 35.9 MHz, the setting value is calculated as follows:
- Round up 35.9
- Convert 36 to binary and set the upper bits and lower bits of the PCKA[7:0] bits to 00h and 24h (0010 0100b).

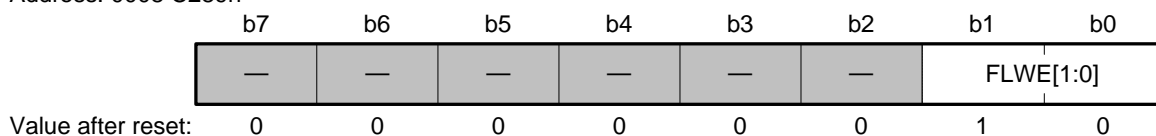
Note 1. When the PCKA[7:0] bits are set to values outside the range from 8 to 50 MHz, do not issue a programming command to the ROM/data flash.

Note 2. When the PCKA[7:0] bits are set to a frequency that is different from the actual frequency, the data of the ROM/data flash may crash.

Note 3. Please note that programming time depends on the frequency to some extent even if the PCKA[7:0] bits are used.

38.2.15 Flash Write Erase Protection Register (FWEPROR)

Address: 0008 C289h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	FLWE[1:0]	Flash Write Erase	b1 and b0 0 0: Write/erase disabled 0 1: Write/erase enabled 1 0: Write/erase disabled (initial value) 1 1: Write/erase disabled	R/W
b7 to b2	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

FWEPROR is a readable/writable register to protect the execution of the flash write/erase with software.

FWEPROR is initialized in software standby mode or deep software standby.

FLWE[1:0] Bits (Flash Write Erase)

These bits protect the execution of the flash write/erase with software.

38.3 Configuration of Memory Mats for the ROM

The ROM of products in the RX62N/RX621 Group is configured of a maximum 512-Kbyte user mat and 16-Kbyte user boot mat. The address range occupied by these mats is shown in Figure 38.2.

Note that for the user mat, the address range for reading differs from the address range for programming and erasure.

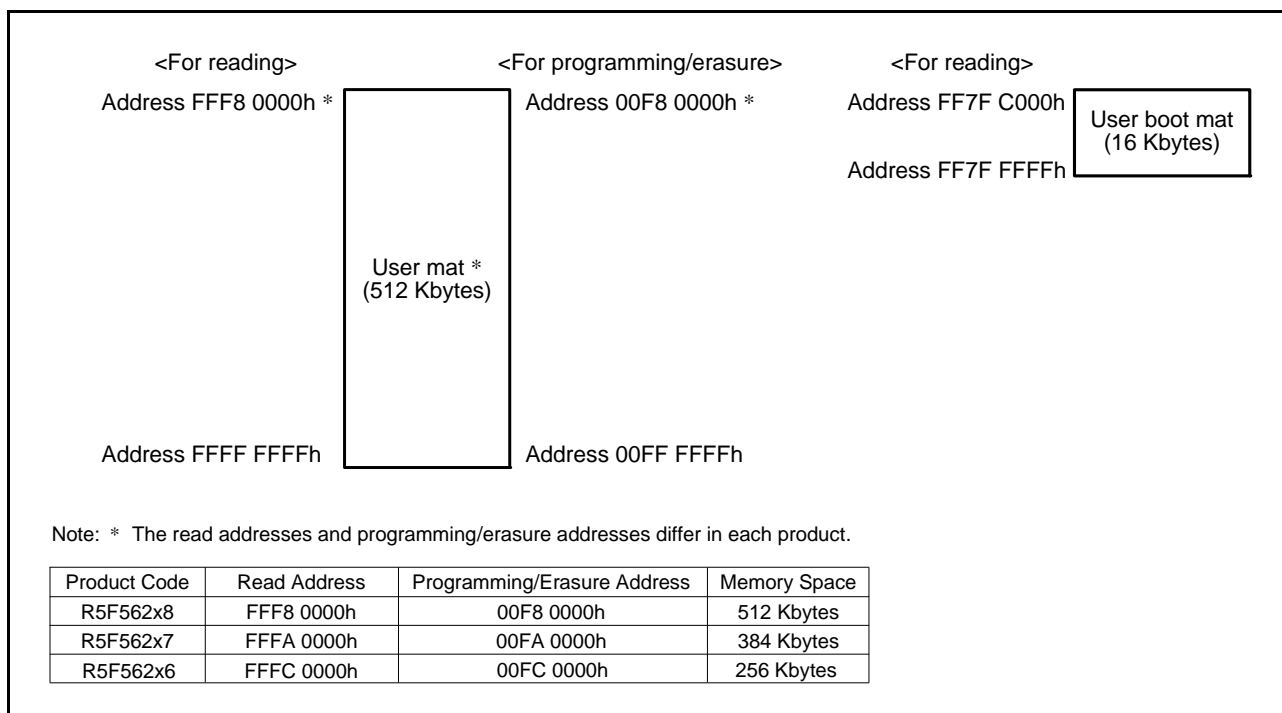


Figure 38.2 Memory Mat Configuration of ROM

38.4 Block Configuration

The configuration of erasure blocks for the user mat is shown in Figure 38.3. As units of erasure, the user mat is divided into 8 blocks of 4 Kbytes each and 30 blocks of 16 Kbytes each. For programming, the user mat consists of the 256-byte units starting from 00h as the lower-order byte of the address.

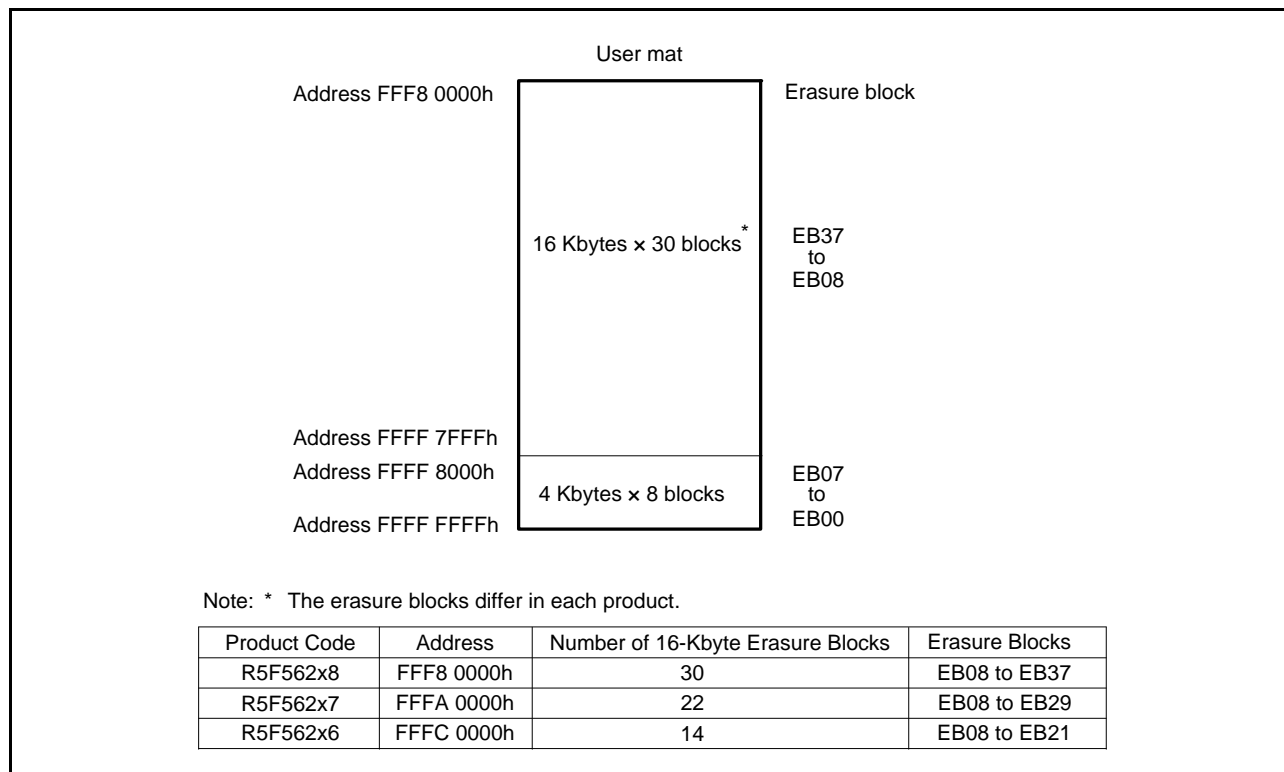


Figure 38.3 Configuration of Erasure Blocks for the User Mat

38.5 Operating Modes Associated with the ROM

Figure 38.4 is a diagram of the operating-mode transitions for the RX62N/RX621 Group.

On release from the reset state, transitions are in accord with the levels on the MD0 and MD1 pins, as shown in Figure 38.4.

For more information on the connections between the settings of the levels on the MD0 and MD1 pins and the operating mode for the RX62N/RX621 Group, refer to section 3, Operating Modes.

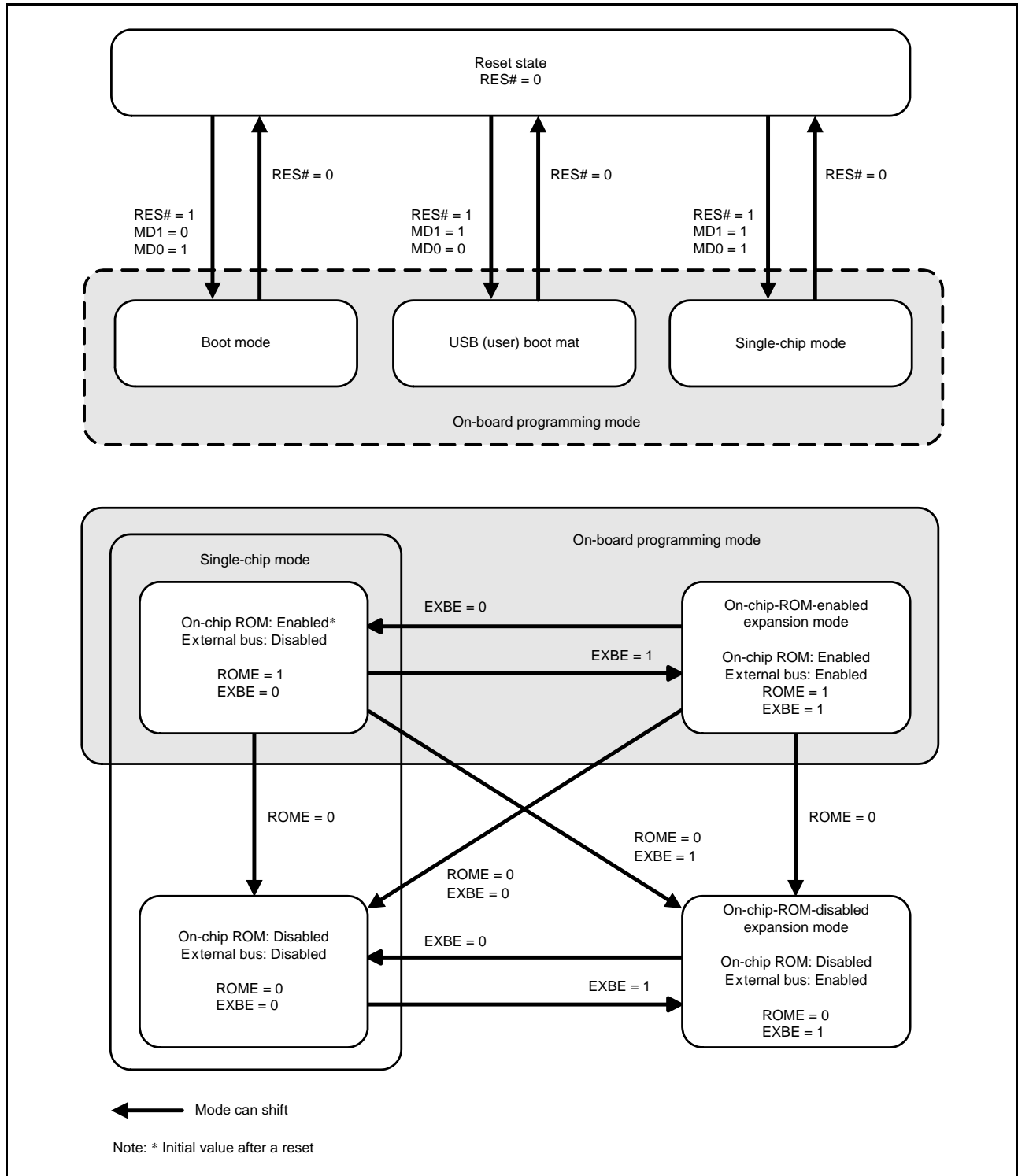


Figure 38.4 Transitions between Operating Modes in Terms of the ROM

Reading, programming, and erasing of the ROM in an on-board device can proceed if the device is in boot, USB (user) boot, or single-chip mode (with on-chip ROM enabled), or in on-chip-ROM-enabled expansion mode.

Which mats are programmable and erasable, the mat from which booting up proceeds after a reset, etc., differs with the mode. The differences between modes are indicated in Table 38.5.

Table 38.5 Differences between Modes

Item	Boot Mode	USB (user) Boot Mode	Single-Chip Mode (with On-chip ROM Enabled) or On-chip-ROM-Enabled Expansion Mode
Environment for programming and erasure	On-board programming	On-board programming	On-board programming
Programmable and erasable mat	User mat User boot mat	User mat	User mat
Division into erasure blocks	Possible*1	Possible*1	Possible
Target mat for booting after a reset	Mat containing the embedded program*2	User boot mat	User mat

Note 1. The entire ROM may be erased at the time of booting up. Specified blocks can subsequently be erased. For details, refer to section 38.9.2, ID Code Protection, section 38.9.4, State Transitions in Boot Mode, and section 38.10.2, State Transitions.

Note 2. Not available to users.

- The user boot mat can be programmed or erased only in boot mode.
- In boot mode, a host is able to program or read the user mat, user boot mat, or data mat via an SCI.
- USB (user) boot mode is activated from the user boot mat. At shipment, the USB boot program is stored in the user boot mat, allowing programming or reading the user mat or data mat via the USB. Programming the user boot mat in boot mode allows programming or reading the user mat or data mat via an arbitrary interface.
- In boot mode, on-chip RAM is employed for the embedded program for use in boot mode. For this reason, preserving the contents of on-chip RAM is not possible in this case.

38.6 Programming and Erasing the ROM

The ROM is programmed and erased by issuing commands to a dedicated sequencer (FCU) for programming and erasure. The FCU has five modes. For programming and erasure, the mode is changed and then commands for programming and erasure are issued.

The mode transitions required to program or erase the ROM and the system of commands are described below. The descriptions apply in common to boot, USB (user) boot, and single-chip mode (with on-chip ROM enabled), and to on-chip-ROM-enabled expansion mode.

38.6.1 FCU Modes

The FCU has five modes or sets of modes. Transitions between modes are caused by modifying FENTRYR or issuing FCU commands. Figure 38.5 is a diagram of the FCU mode transitions.

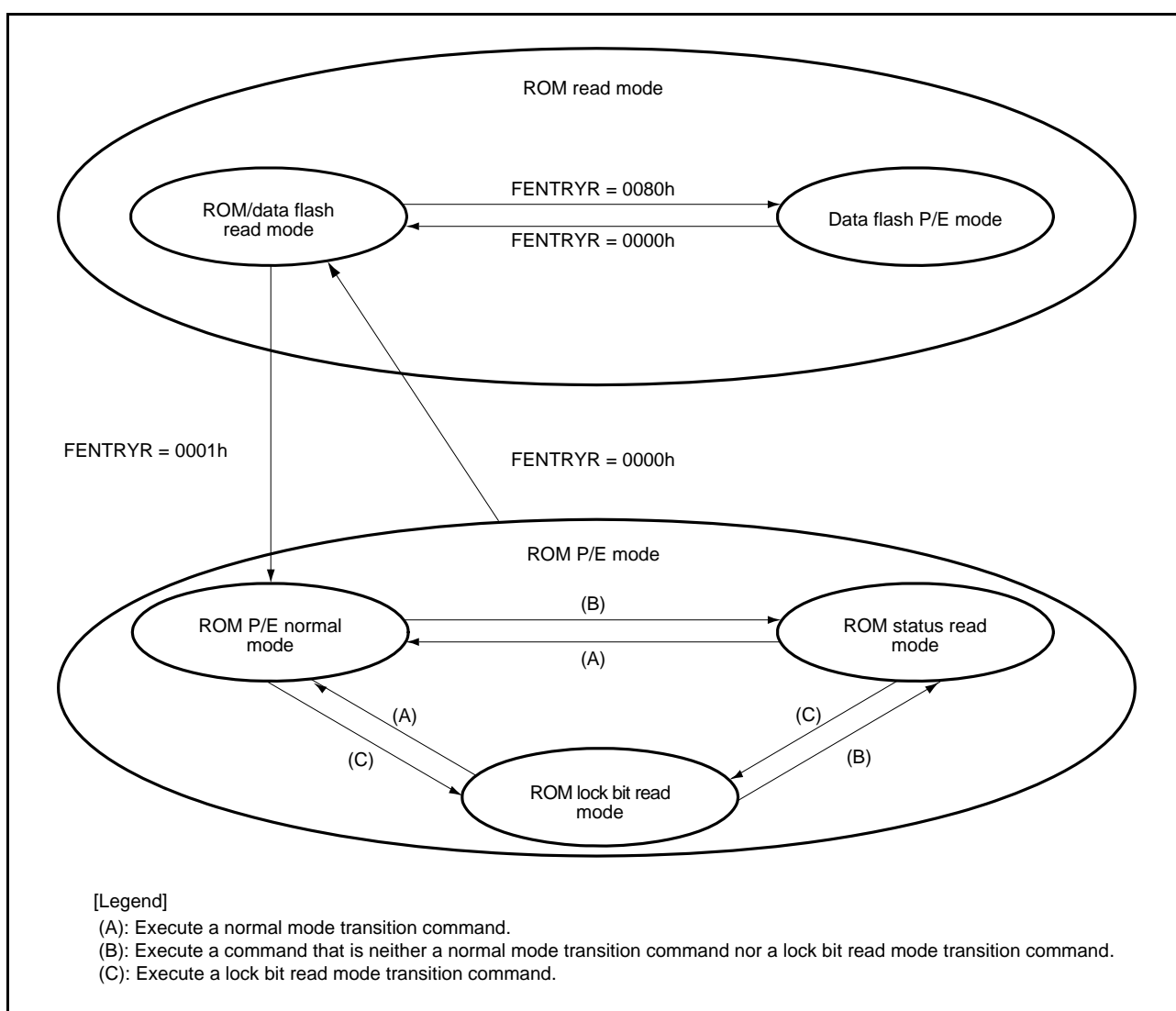


Figure 38.5 Mode Transitions of the FCU (Associated with the ROM)

38.6.1.1 ROM Read Modes

The ROM read modes are for high-speed reading of the ROM. Access to an address for reading can be accomplished in one cycle of ICLK.

ROM/data-flash read mode and data-flash P/E mode are the two kinds of ROM reading mode.

(1) ROM/Data Flash Read Mode

This mode is for reading the ROM and data-flash memory. The FCU does not accept commands.

The FCU enters this mode when the FENTRY0 bit in FENTRYR is set to 0, and the FENTRYD bit in FENTRYR is set to 0.

(2) Data Flash P/E Modes

These modes are for programming and erasure of the data-flash memory. High-speed reading of the ROM is also possible. Although the FCU accepts FCU commands related to the data-flash memory in this mode, it does not accept FCU commands related to the ROM. The FCU enters these modes when the FENTRY0 bit in FENTRYR is set to 0, or the FENTRYD bit in FENTRYR is set to 1.

For details on the data flash P/E modes, see section 39.6.1, FCU Modes

38.6.1.2 ROM P/E Modes

The ROM P/E modes are for programming and erasure of the ROM. High-speed reading of the ROM is not possible in these modes. Read access to an address within the range for reading causes a ROM-access violation, and the FCU enters the command-locked state (see section 38.8.2, Error Protection).

ROM P/E normal mode, ROM status read mode, and ROM lock-bit read mode are the three ROM P/E modes.

(1) ROM P/E Normal Mode

The transition to ROM P/E normal mode is the first transition in the process of programming or erasing the ROM. The FCU enters this mode when the FENTRYD bit in FENTRYR is set to 0, and the FENTRY0 bit in FENTRYR is set to 1 in ROM read mode, or when the normal mode transition command is received in ROM P/E modes. Table 38.8 lists the acceptable commands in this mode.

Read access to an address within the range for programming and erasure while the FENTRY0 bit in FENTRYR is set to 1 causes a ROM-access violation, and the FCU enters the command-locked state (see section 38.8.2, Error Protection).

(2) ROM Status Read Mode

The ROM status read mode is for reading information on the state of the ROM. The FCU enters this mode when a command other than the normal mode transition and lock bit read mode transition command is received in ROM P/E modes.

ROM status read mode encompasses the states where the FRDY bit in FSTATR0 is 0 and the command-locked state after an error has occurred. Table 38.8 lists the acceptable commands in this mode.

Read access to an address within the range for programming and erasure while the FENTRY0 bit in FENTRYR is 1 will actually read out the value of FSTATR0.

(3) ROM Lock-Bit Read Mode

The ROM lock-bit read mode is for reading the values of the lock bits of the ROM. The FCU enters this mode when a lock-bit read mode transition command is received in ROM P/E modes. Table 38.8 lists the acceptable commands in this mode.

In read access to an address within the range for programming and erasure while the FENTRY0 bit in FENTRYR is 1, all bits of the value read out have the value of the lock bit of the erasure block that includes the accessed address.

38.6.2 FCU Commands

FCU commands consist of commands for mode transitions of the FCU and commands for programming and erasure. Table 38.6 lists the FCU commands for use with the ROM.

Table 38.6 FCU Commands for Use with the ROM

Command	Description
P/E normal mode transition	Changes the mode to normal mode (see section 38.6.3, Connections between FCU Modes and Commands)
Status read mode transition	Changes the mode to status read mode (see section 38.6.3, Connections between FCU Modes and Commands)
Lock bit read mode transition (lock bit read 1)	Changes the mode to lock bit read mode (see section 38.6.3, Connections between FCU Modes and Commands)
Peripheral clock notification	Sets the frequency of the peripheral clock
Programming	ROM programming (in 256-byte units)
Block erasure	ROM erasure (in block units, with the lock bit being erased simultaneously)
P/E suspension	Suspends programming/erasure
P/E resumption	Resumes programming/erasure
Status register clearing	Clears the ILGLERR, ERSERR and PRGERR bits in FSTATR0 and releases the FCU from the command locked state
Lock bit read 2/blank checking	Reads the lock bit of a specified erasure block (the value of the lock bit is reflected in the FLOCKST bit of FSTATR1)/blank checking of the data-flash memory
Lock bit programming	Programs the lock bit of a specified erasure block

The lock bit read 2 command is also used as the blank-checking command for the data-flash memory. That is, when a lock bit read 2 command is issued for the data flash, blank checking is executed for the data-flash memory (see section 39, Data Flash Memory (Flash Memory for Data Storage)).

Commands for the FCU are issued by write access to addresses within the range for programming and erasure. Table 38.7 shows the formats of the FCU commands. Write access as listed in Table 38.7 and in accord with certain conditions causes the FCU to execute processing for the corresponding command. For details on the conditions for the acceptance of the individual FCU commands, see section 38.6.3, Connections between FCU Modes and Commands. For how to use the FCU commands, see section 38.6.4, FCU Command Usage.

Table 38.7 FCU Command Formats

Command	Number of bus cycles	First Cycle		Second Cycle		Third Cycle		4th to 5th Cycles		6th Cycle		7th to 130th Cycles		131st Cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
P/E normal mode transition	1	RA	FFh	—	—	—	—	—	—	—	—	—	—	—	—
Status read mode transition	1	RA	70h	—	—	—	—	—	—	—	—	—	—	—	—
Lock bit read mode transition (lock bit read 1)	1	RA	71h	—	—	—	—	—	—	—	—	—	—	—	—
Peripheral clock setting	6	RA	E9h	RA	03h	RA	0F0Fh	RA	0F0Fh	RA	D0h	—	—	—	—
Programming	131	RA	E8h	RA	80h	W A	WDn	RA	WDn	RA	WDn	RA	WDn	RA	D0h
Block erasure	2	RA	20h	BA	D0h	—	—	—	—	—	—	—	—	—	—
P/E suspension	1	RA	B0h	—	—	—	—	—	—	—	—	—	—	—	—
P/E resumption	1	RA	D0h	—	—	—	—	—	—	—	—	—	—	—	—
Status register clearing	1	RA	50h	—	—	—	—	—	—	—	—	—	—	—	—
Lock bit read 2	2	RA	71h	BA	D0h	—	—	—	—	—	—	—	—	—	—
Lock bit programming	2	RA	77h	BA	D0h	—	—	—	—	—	—	—	—	—	—

[Legend] Address column RA: ROM programming/erasure address
 When the FENTRY0 bit in FENTRYR is 1: An address from 00F8 0000h to 00FF FFFFh
 WA: ROM programming-destination address
 Start address for programming of 256 bytes of data
 BA: ROM erasure block address
 An address within the target erasure block (specified as an address in the range for programming and erasure)
 Data column WDn: nth word of data for programming (n = 1 to 128)

38.6.3 Connections between FCU Modes and Commands

The sets of FCU commands that can be accepted in each of the FCU modes are fixed. Furthermore, which commands are acceptable in a given FCU mode also depends on the state of the FCU.

Issuing of an FCU command must follow checking of the FCU's state after transitions of the FCU mode.

Commands that are acceptable in the various FCU modes and states are listed in Table 38.8. Issuing a command that is not currently acceptable leads to the FCU being placed in the command-locked state (see section 38.8.2, Error Protection).

Issuing of an FCU command must follow checking of the values of the FRDY, ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and of the FCUERR bit in FSTATR1 after transitions of the FCU mode. Furthermore, the CMDLK bit in FASTAT can be checked to see if an error has occurred. The value of the CMDLK bit in FASTAT is the logical OR of the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FCUERR bit in FSTATR1.

Table 38.8 Acceptable Commands and the State and Mode (ROM P/E Mode) of the FCU

	P/E Normal Mode			Status Read Mode							Lock-Bit Read Mode		
	Programming suspended	Erase suspended	Other state	Programming or erase	Processing to suspend programming or	Lock bit read 2 processing	Programming suspended	Erase suspended	Command-locked state	Other state	Programming suspended	Erase suspended	Other state
FSTATR0.FRDY bit	1	1	1	0	0	0	1	1	0/1	1	1	1	1
FSTATR0.SUSRDY bit	0	0	0	1	0	0	0	0	0	0	0	0	0
FSTATR0.ERSSPD bit	0	1	0	0	0/1	0	0	1	0	0	0	1	0
FSTATR0.PRGSPD bit	1	0	0	0	0/1	0	1	0	0	0	1	0	0
FASTAT.CMDLK bit	0	0	0	0	0	0	0	0	1	0	0	0	0
Normal mode transition	A	A	A	X	X	X	A	A	X	A	A	A	A
Status read transition	A	A	A	X	X	X	A	A	X	A	A	A	A
Lock-bit read transition (lock bit read 1)	A	A	A	X	X	X	A	A	X	A	A	A	A
Peripheral clock setting	X	X	A	X	X	X	X	X	X	A	X	X	A
Programming	X	*	A	X	X	X	X	*	X	A	X	*	A
Block erasure	X	X	A	X	X	X	X	X	X	A	X	X	A
P/E suspension	X	X	X	A	X	X	X	X	X	X	X	X	X
P/E resumption	A	A	X	X	X	X	A	A	X	X	A	A	X
Status register clearing	A	A	A	X	X	X	A	A	A	A	A	A	A
Lock bit read 2	A	A	A	X	X	X	A	A	X	A	A	A	A
Lock bit programming	X	*	A	X	X	X	X	*	X	A	X	*	A

[Legend]

A: Acceptable

*: Only programming is acceptable for blocks other than the block where erasure was suspended

X: Not acceptable

38.6.4 FCU Command Usage

The set of FCU commands consists of commands for FCU mode transitions, actually programming or erasing the ROM, error processing, and suspension and resumption. The following passages describe the various commands. For a description of the modes and states where the respective commands are acceptable, see section 38.6.3, Connections between FCU Modes and Commands.

38.6.4.1 Mode Transitions

This subsection covers the commands for mode transitions. For an illustration of the various transitions between modes, see Figure 38.5.

(1) Switching to ROM P/E Mode

A transition to ROM P/E mode is required before executing an FCU command for the ROM becomes possible.

Setting the FENTRY0 bit in FENTRYR to 1 causes a transition to ROM P/E mode for programming and erasure of the corresponding address range.

Before actually proceeding to program or erase the ROM, enable programming and erasure by writing 01h as a byte to FWEPROR (see section 38.2.15, Flash Write Erase Protection Register (FWEPROR)).

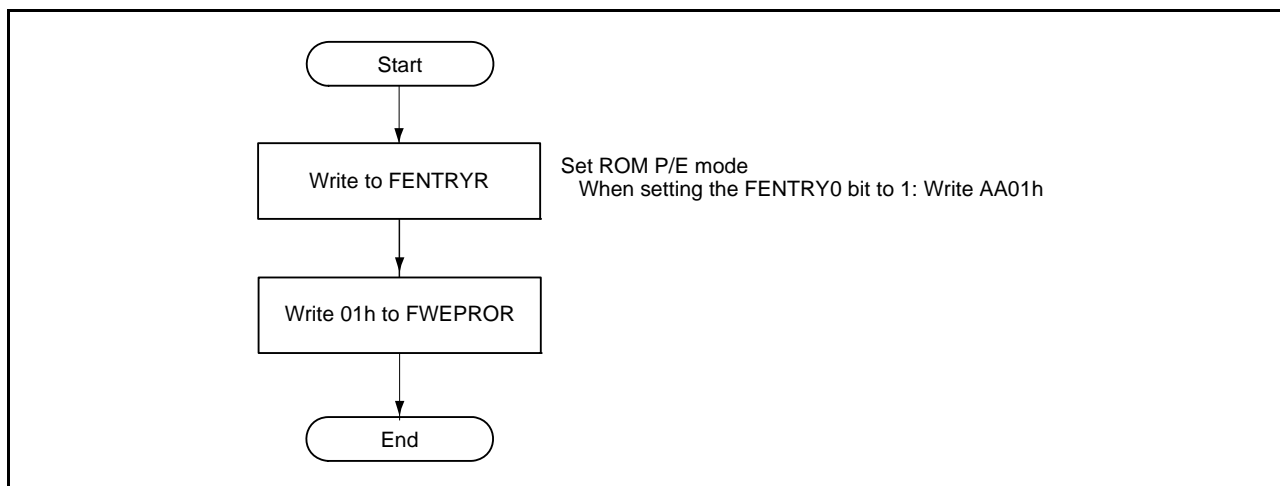


Figure 38.6 Procedure for Transition to ROM P/E Mode

(2) Switching to ROM Read Mode

High-speed reading of the ROM requires clearing of the FENTRY0 bit in FENTRYR, which places the FCU in ROM read mode.

Writing of 02h to FWEPROR is also required to disable programming and erasure (see section 38.2.15, Flash Write Erase Protection Register (FWEPROR)).

Before switching the FCU from P/E mode to read mode, ensure that all processing of FCU commands has been completed and that the FCU has not detected an error.

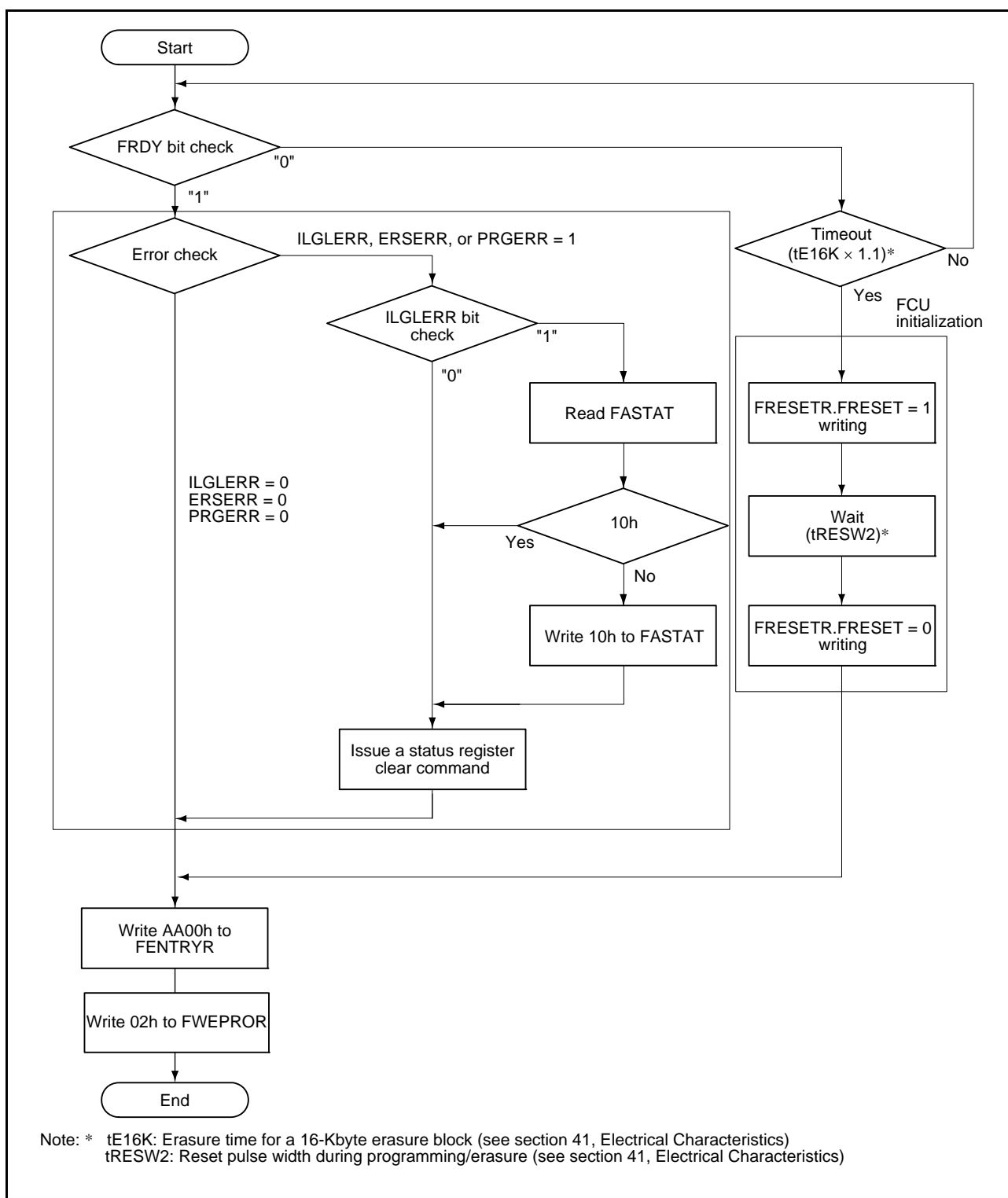


Figure 38.7 Procedure for Transition to ROM Read Mode

(3) Switching to ROM P/E Normal Mode

Two methods are available for the transition to ROM P/E normal mode: setting the FENTRYR register while the FCU is in ROM read mode (see section 38.6.1, FCU Modes), or issuing the normal mode transition command while the FCU is in ROM P/E mode (see Figure 38.8). The normal mode transition command is issued by writing FFh as a byte to an address in the range for programming and erasure of the ROM.

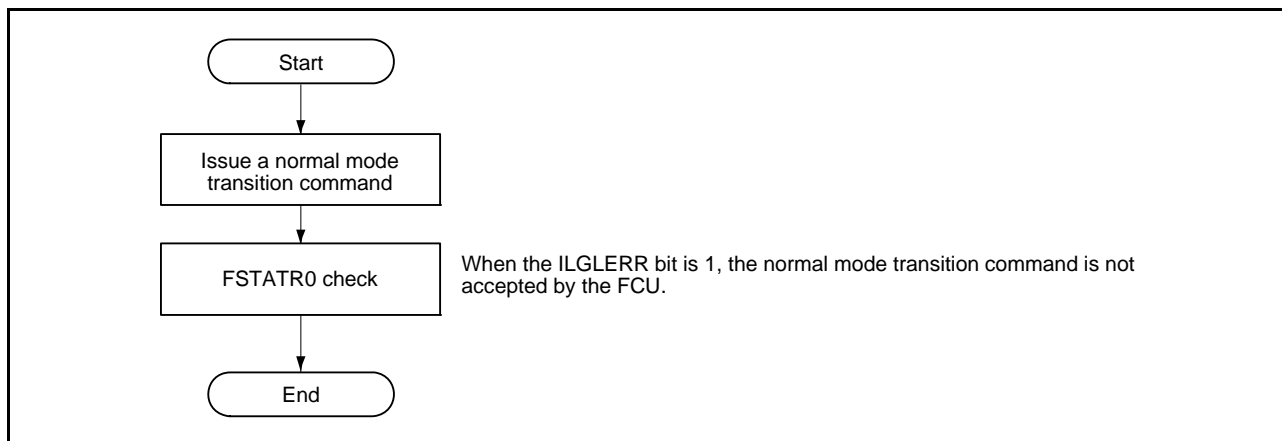


Figure 38.8 Procedure for Transition to ROM P/E Normal Mode

(4) Switching to ROM Status Read Mode

Issuing an FCU command other than a normal mode transition or lock bit read mode transition command places the FCU in ROM status read mode. The same transition can be obtained by issuing the status read mode transition command. Figure 38.9 shows the procedure for checking the register FSTATR0 as an example. In the example, the status read mode transition command is issued to place the FCU in ROM status read mode, and the value of FSTATR0 is obtained by read access to the area for programming and erasure and then checked.

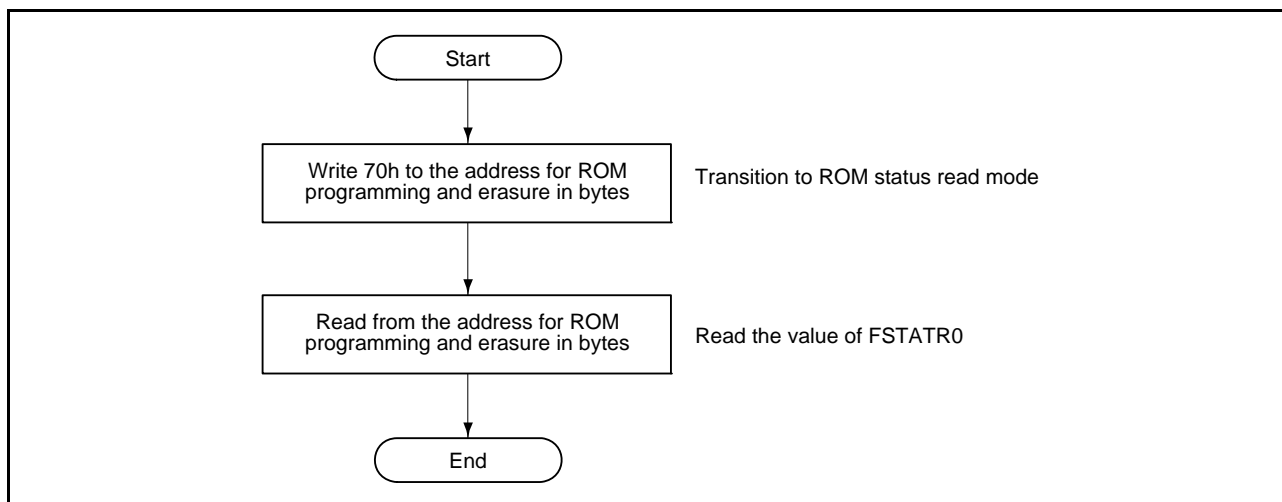


Figure 38.9 Procedure for Transition to ROM Status Read Mode and the Status Checking

(5) Switching to ROM Lock-Bit Read Mode

Clearing the FRDMD bit in FMODR (memory area method) issues a lock bit read mode transition (lock bit read 1) command. After the transition to ROM lock bit read mode, lock bit value are obtained by read access to the area for ROM programming and erasure. All bits of a value thus read out have the value of the lock bit of the erasure block that contains the accessed address (Figure 38.10).

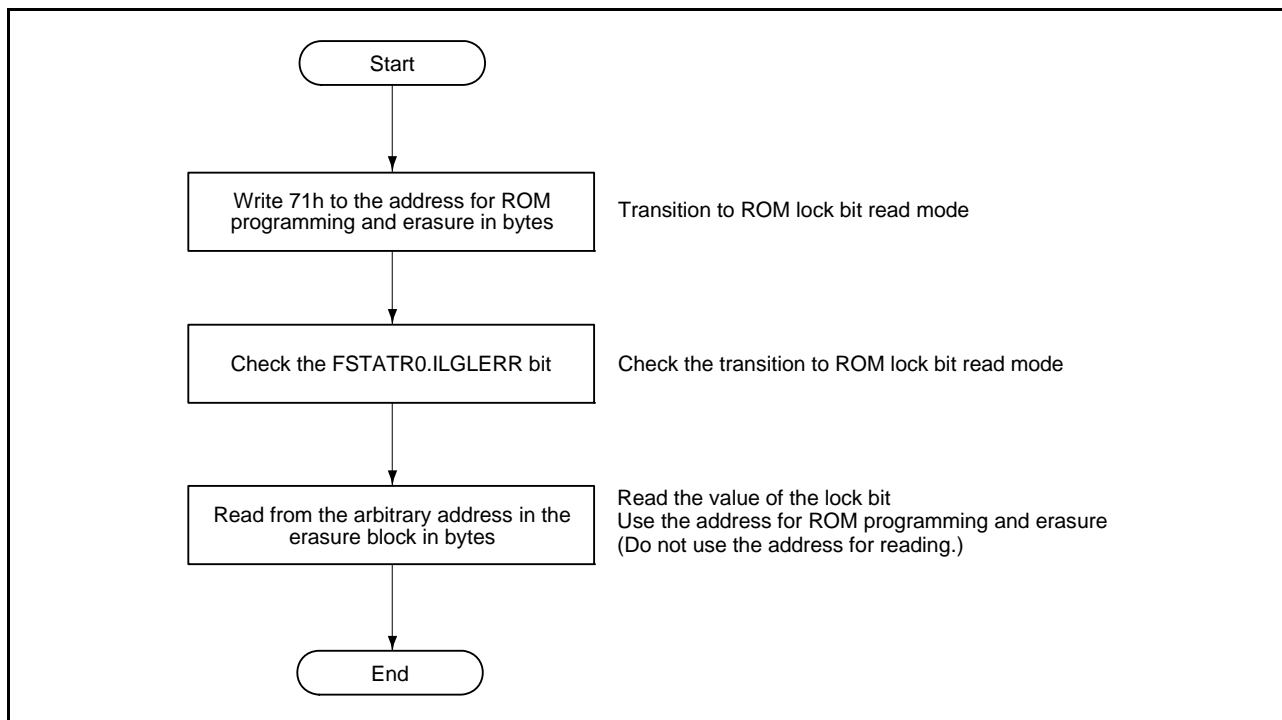


Figure 38.10 Procedure for Transition to ROM Lock-Bit Read Mode and the Lock-Bit Read Method

38.6.4.2 Programming and Erasure Procedures

The following passages describe the flow of procedures for programming or erasing the ROM. For details on the acceptance of commands by the FCU, see section 38.6.3, Connections between FCU Modes and Commands.

Figure 38.11 is a simple flowchart of the procedure for executing FCU commands.

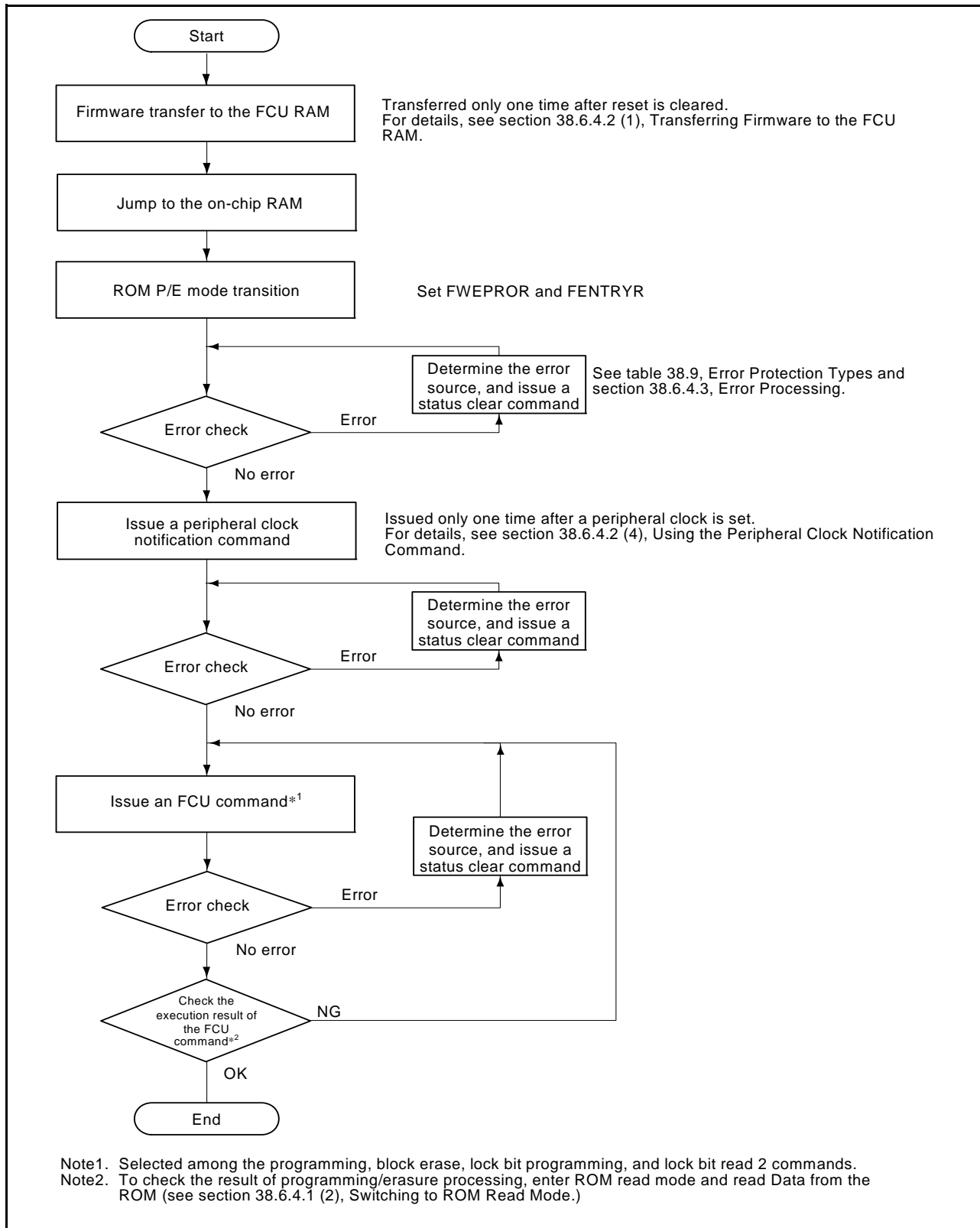


Figure 38.11 Simple Flowchart of the Procedure for Programming and Erasure

(1) Transferring Firmware to the FCU RAM

FCU commands can only be used if the FCU RAM holds the firmware for the FCU. The FCU RAM does not hold the FCU firmware immediately after the chip has been booted up, so the firmware must be copied from the FCU firmware area to the FCU RAM. Furthermore, when the FCUERR bit in FSTATR1 is set to 1, the FCU must be reset and the firmware recopied because the firmware stored in the FCU RAM may have been corrupted.

Figure 38.12 shows the flow of the procedure for transferring firmware to the FCU RAM. Before writing data to the FCU RAM, set FENTRYR to 0000h and stop the FCU. See section 14, DMA Controller (DMACA), for information on setting up the DMACA to handle data transfer.

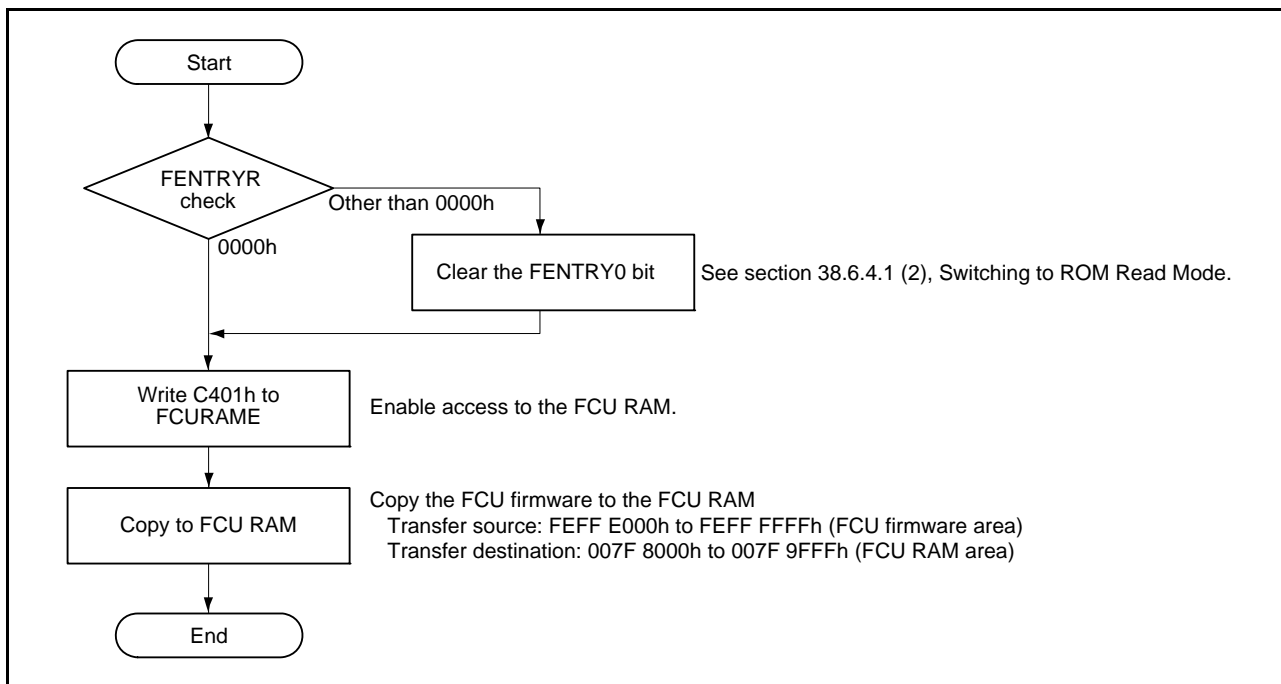


Figure 38.12 Procedure for Firmware Transfer to FCU RAM

(2) Jumping to Locations in On-chip RAM

Since fetching instructions from the ROM is not possible while the ROM is being programmed or erased, code has to be executed from an area other than the ROM. Copy the required program code to on-chip RAM and then make execution jump to the address where the code starts in on-chip RAM.

(3) Transition to ROM P/E Mode

The FCU is placed in ROM P/E mode by the settings of the FENTRY0 bit in FENTRYR and of FWEPROR. Details are given in section 38.6.4.1 (1)Switching to ROM P/E Mode.

(4) Using the Peripheral Clock Notification Command

The peripheral clock is used in programming and erasing the ROM, so the frequency of this clock has to be set in the PCKAR. Frequencies in the range from 8 to 50 MHz are selectable. If a frequency within this range has not been set, the FCU will detect the error and enter the command-blocked state (see section 38.8.2, Error Protection).

The peripheral clock notification command is used after the PCKAR setting has been made. In the first and second cycles for the peripheral clock notification command, respectively, the values E9h and 03h are written to the address range for programming and erasure of the ROM. Word-unit writing is used in the third to fifth cycles of the command.

Accordingly, make sure that the addresses used are aligned with four-byte boundaries. After 0F0Fh has been written three times (as a word) to the address range for programming and erasure of the ROM, the process of the FCU setting the frequency of the peripheral clock starts once the value D0h has been written as a byte in the sixth cycle. The FRDY bit in FSTATR0 can be used to check whether or not the settings have been completed.

Addresses that can be used in the first to sixth cycles differ according to the setting of the FENTRY0 bit in FENTRYR. Ensure that the addresses suit the setting of the bit. If issuing of the command is attempted with an erroneous combination of the setting of the bit and specified addresses, the FCU will detect the error and enter the command-blocked state (see section 38.8.2, Error Protection).

Furthermore, if the setting for the peripheral clock in use will not be changed from this setting after release from the reset state, this setting is also valid for the next FCU command.

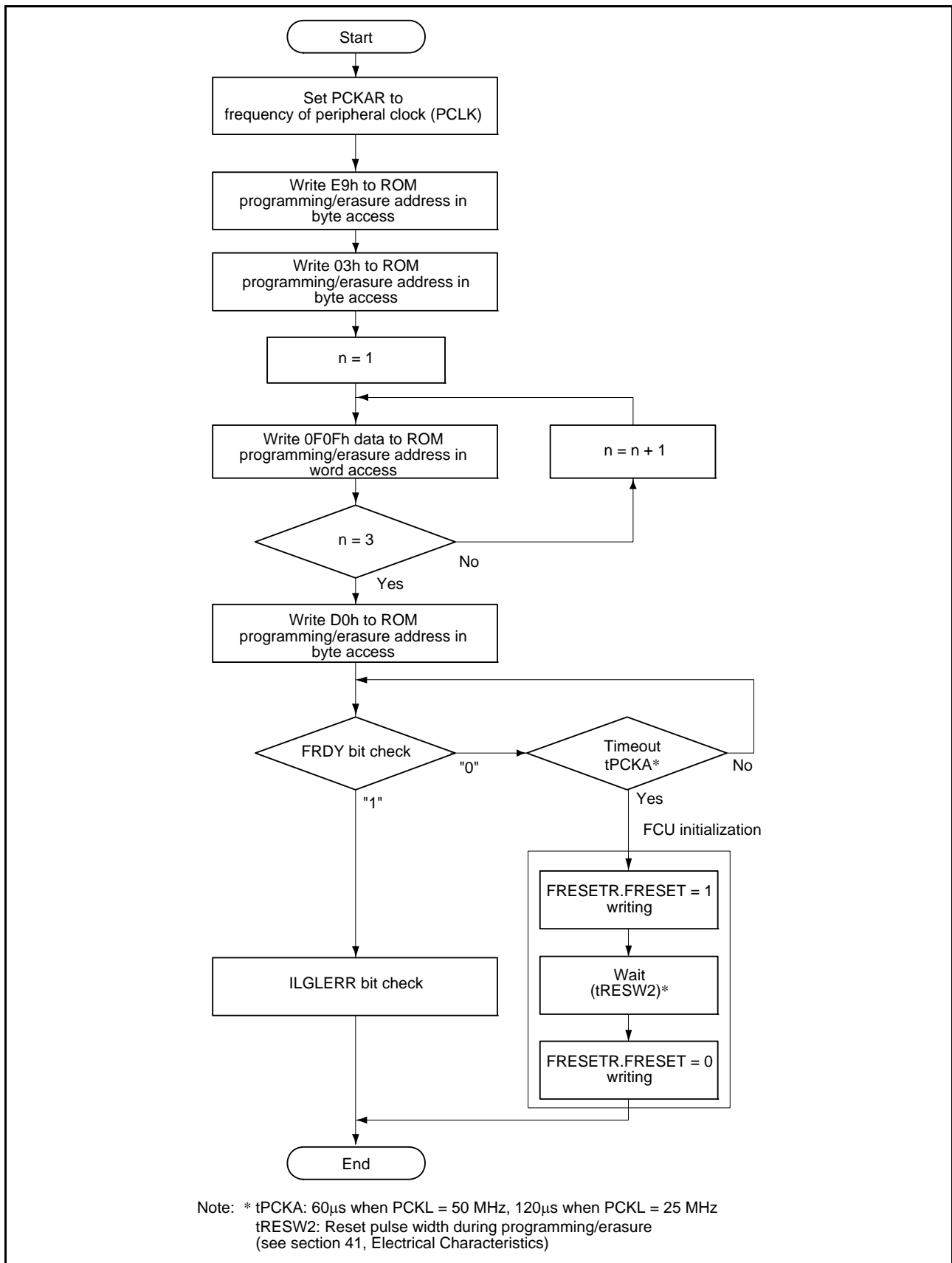


Figure 38.13 Using the Peripheral Clock Notification Command

(5) Programming

The programming command is used to write data to the ROM.

In the first and second cycles for the peripheral clock notification command, respectively, the values E8h and 80h are written to the address range for programming and erasure of the ROM. In the third cycle, write the actual data to be programmed, as a word unit, to the target address for programming. For this first address, always use an address that is on a 256-byte boundary. In the fourth to the 130th cycles, write the data for programming in 127 word-unit rounds to the address range for programming and erasure of the ROM. Once the value D0h has been written as a byte to the address range for programming and erasure of the ROM in the 131st cycle, the FCU begins the actual process of programming the ROM. The FRDY bit in FSTATR0 can be used to check whether or not the programming has been completed.

Addresses that can be used in the first to 131st cycles differ according to the setting of the FENTRY0 bit in FENTRYR. Ensure that the addresses suit the setting of the bit. If issuing of the command is attempted with an erroneous combination of the setting of the bit and specified addresses, the FCU will detect the error and enter the command-blocked state (see section 38.8.2, Error Protection).

In cases where the target range in the third to 130th cycles includes addresses that do not require programming, use FFFFh as the data for programming to those addresses. To execute a programming with lock bit protection disabled, proceed with programming after setting the FPROTCN bit in FPROTR.

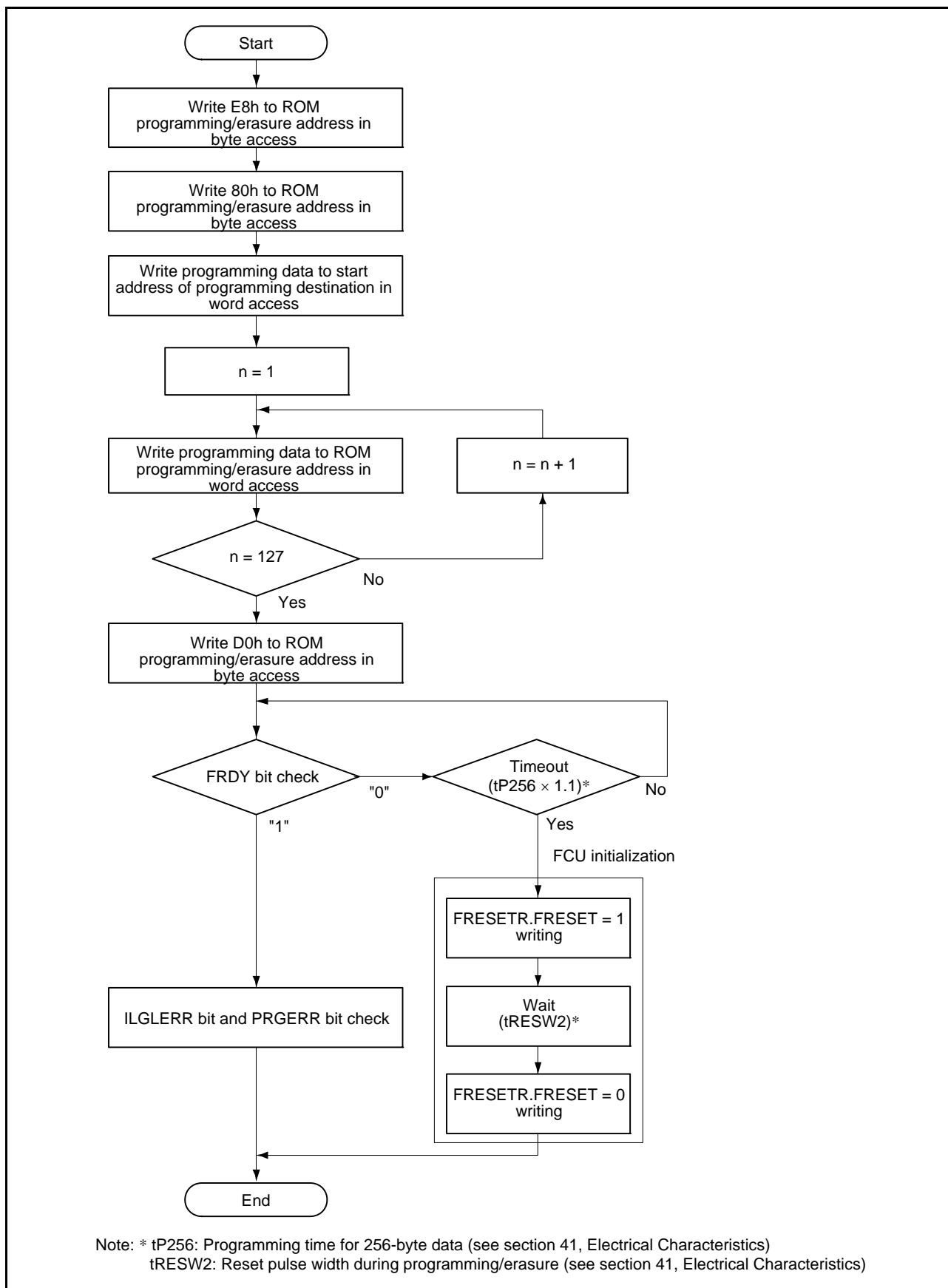


Figure 38.14 Procedure for ROM Programming

(6) Erasure

To erase data from the ROM, use the block erase command.

Write 20h to the ROM programming/erasure address in byte access in the first cycle of the block erase command. When D0h is written to an arbitrary address in an erasure target block in byte access in the second cycle, the FCU start the erasure processing for the ROM. Whether erasure is completed can be checked with the FRDY bit in FSTATR0. The ROM in the erased state can be read by the CPU as FFFF FFFFh in 32-bit access.

To execute an erasure with lock bit protection disabled, set the FPROTCN bit in FPROTR before erasure.

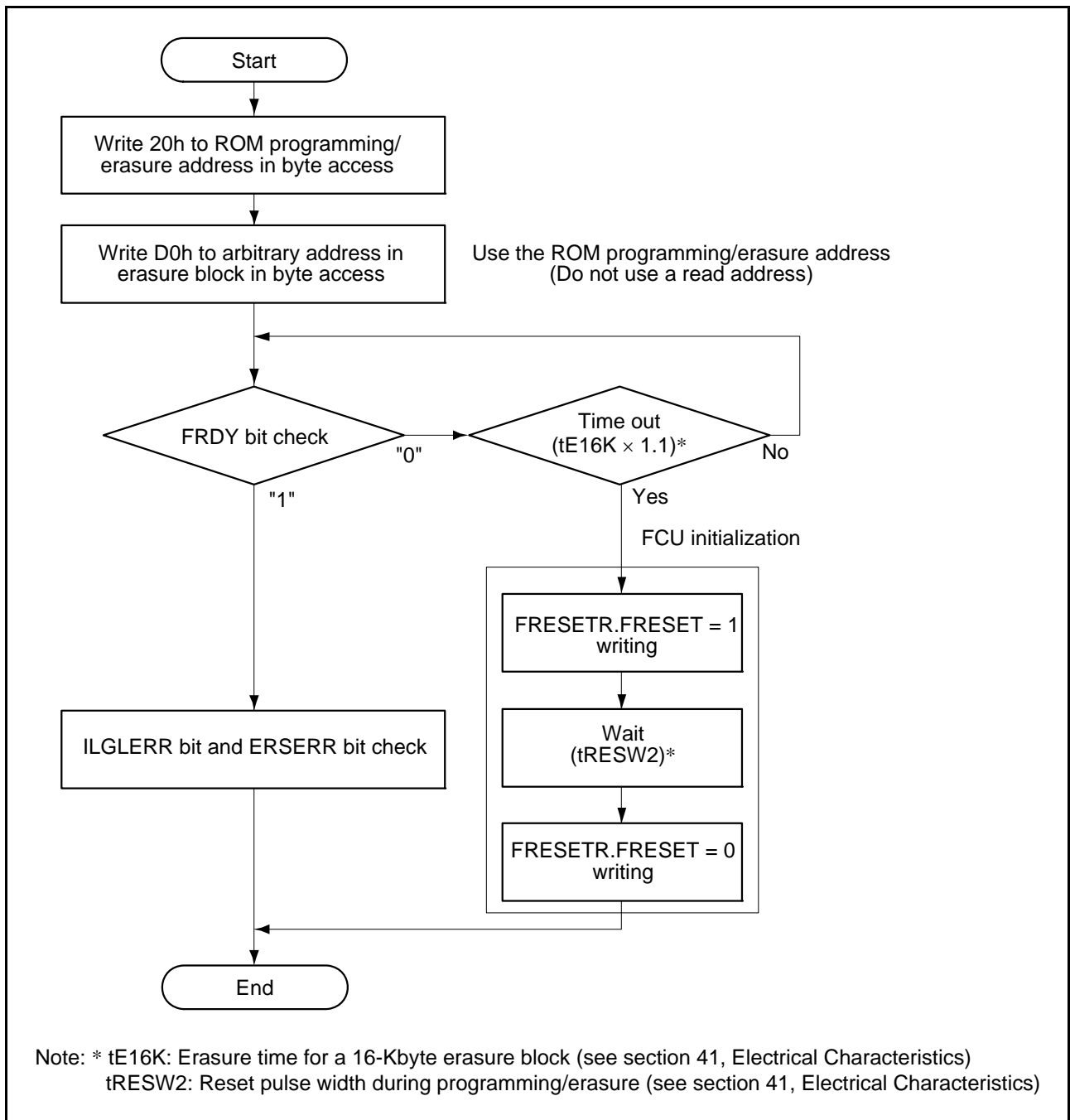


Figure 38.15 Procedure for ROM Erasure

(7) Programming/Erasing to Lock Bit

Each erasure block in the user mat includes a lock bit. To write to a lock bit, use the lock bit programming command. In the first cycle of the lock bit programming command, 77h is programmed to the ROM programming/erasure address in byte access. When D0h is programmed to an arbitrary address in an erasure block whose lock bit is to be programmed in the second cycle in byte access, the FCU start the programming processing of the lock bit. Whether programming is completed can be checked with the FRDY bit in FSTATR0.

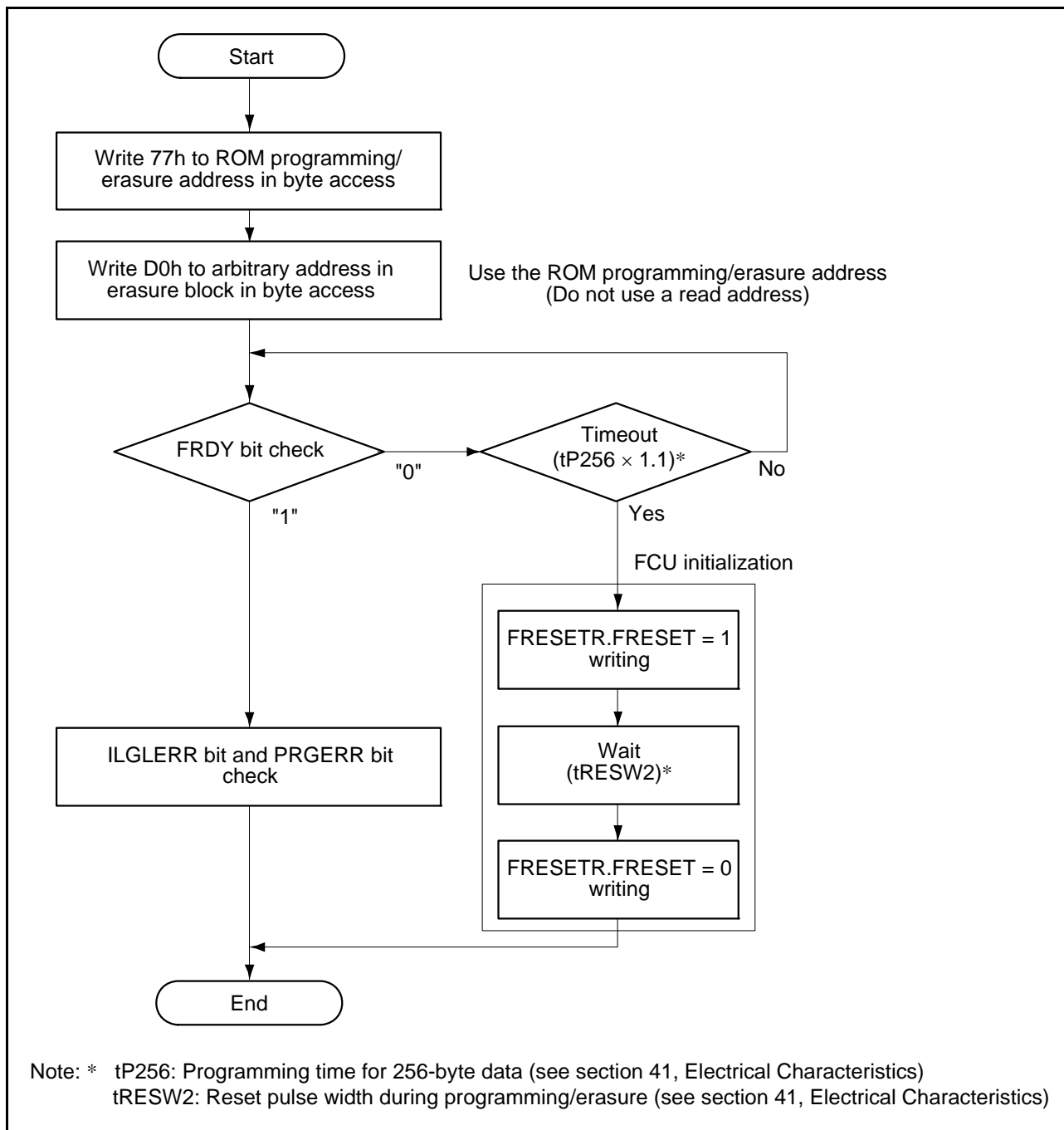


Figure 38.16 Procedure for Writing to the Lock Bit

To erase a lock bit, use the block erase command.

When the FPROTCN bit in FPROTR is 0, erasure blocks whose lock bit is set to 0 cannot be erased. When erasing a lock bit, issue a block erase command with the FPROTCN bit set to 1. Using the block erase command erases all data in the erasure block. It is impossible to erase only a lock bit.

(8) Reading Lock Bits

Lock bits can be read out by either reading from a memory area or reading from a register.

The lock bit read 2 command is issued in the case of the register-reading method (i.e. when the FRDMD bit in FMODER is 1). This command is issued to an address within the block for which the lock bit is to be read out; the address range is that for programming and erasing the ROM. In the first and second cycles of the lock bit read 2 command, the values 71h and D0h are written as bytes; once these values have been written, the value of the lock bit for the specified erasure block is copied to the FLOCKST bit in FSTATR1.

In the case of the memory area reading method (i.e. when the FRDMD bit in FMODER is 0), the FCU is placed in lock-bit reading mode, and the lock bit is obtained by reading from an address within the address range for programming and erasure of the ROM. Details are given in section 38.6.4 (5)Switching to ROM Lock-Bit Read Mode.

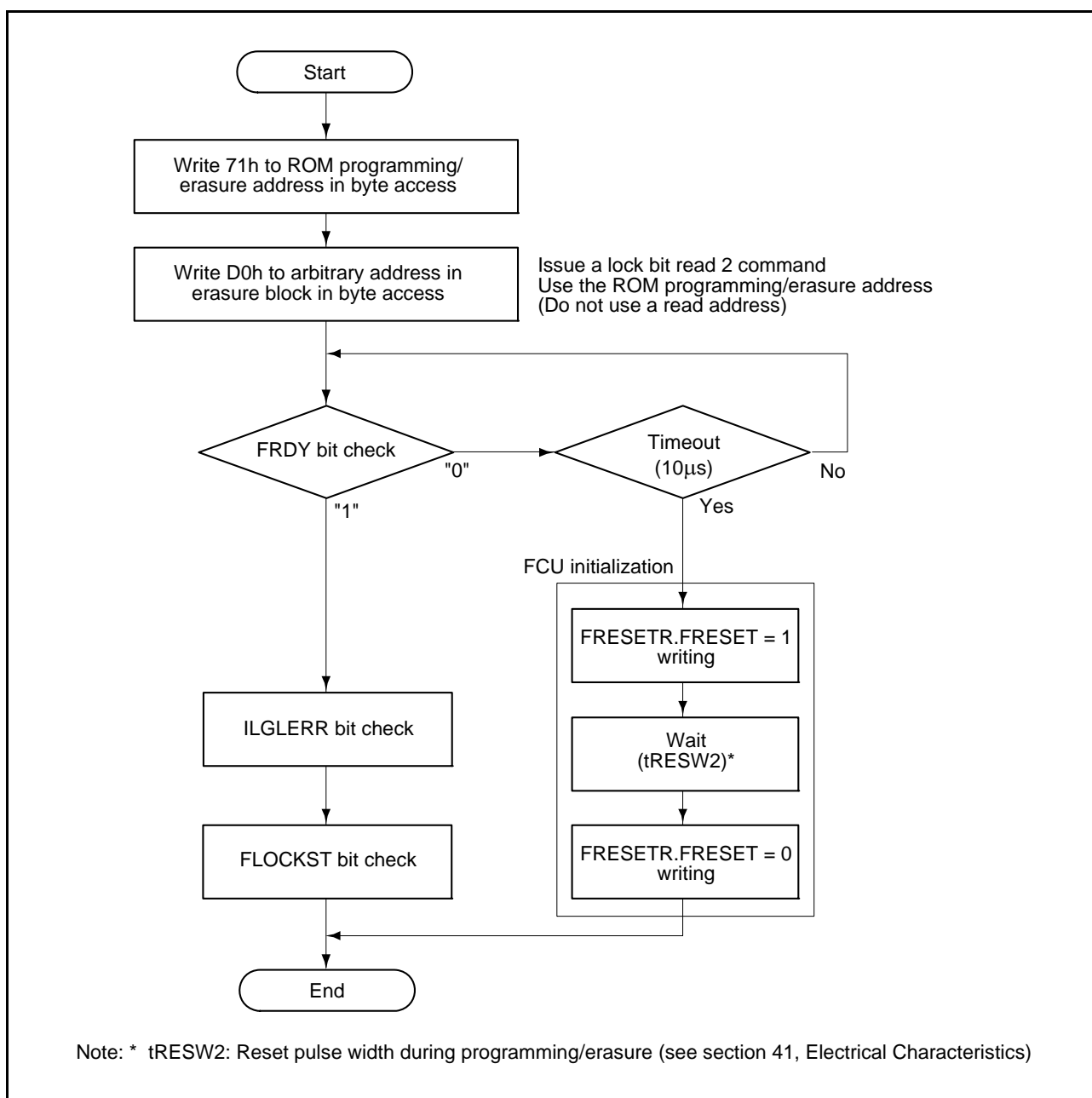


Figure 38.17 Procedure for Reading Lock Bit in Register Read Mode

38.6.4.3 Error Processing

The following passages describe the flow of error processing. For details on errors, see section 38.8, Protection.

(1) Checking Flash Status Register 0 (FSTATR0)

To check FSTATR0, read FSTATR0 directly or read the ROM programming/erasure address in ROM status read mode. For the reading in ROM status read mode, see section 38.6.4.1 (4)Switching to ROM Status Read Mode.

(2) Clearing Flash Status Register 0 (FSTATR0)

To clear the ILGLERR, ERSERR and PRGERR bits in FSTATR0, use the status register clear command.

When one of the ILGLERR, ERSERR and PRGERR bits in FSTATR0 is 1, the FCU is placed in the command-locked state and receives no FCU commands other than the status register clear command. If the ILGLERR is 1, also check the values of the ROMAЕ, DFLAE, DFLRPE, and DFLWPE bits in FASTAT. Even if issuing a status register clear command without clearing these bits, the ILGLERR bit is not cleared.

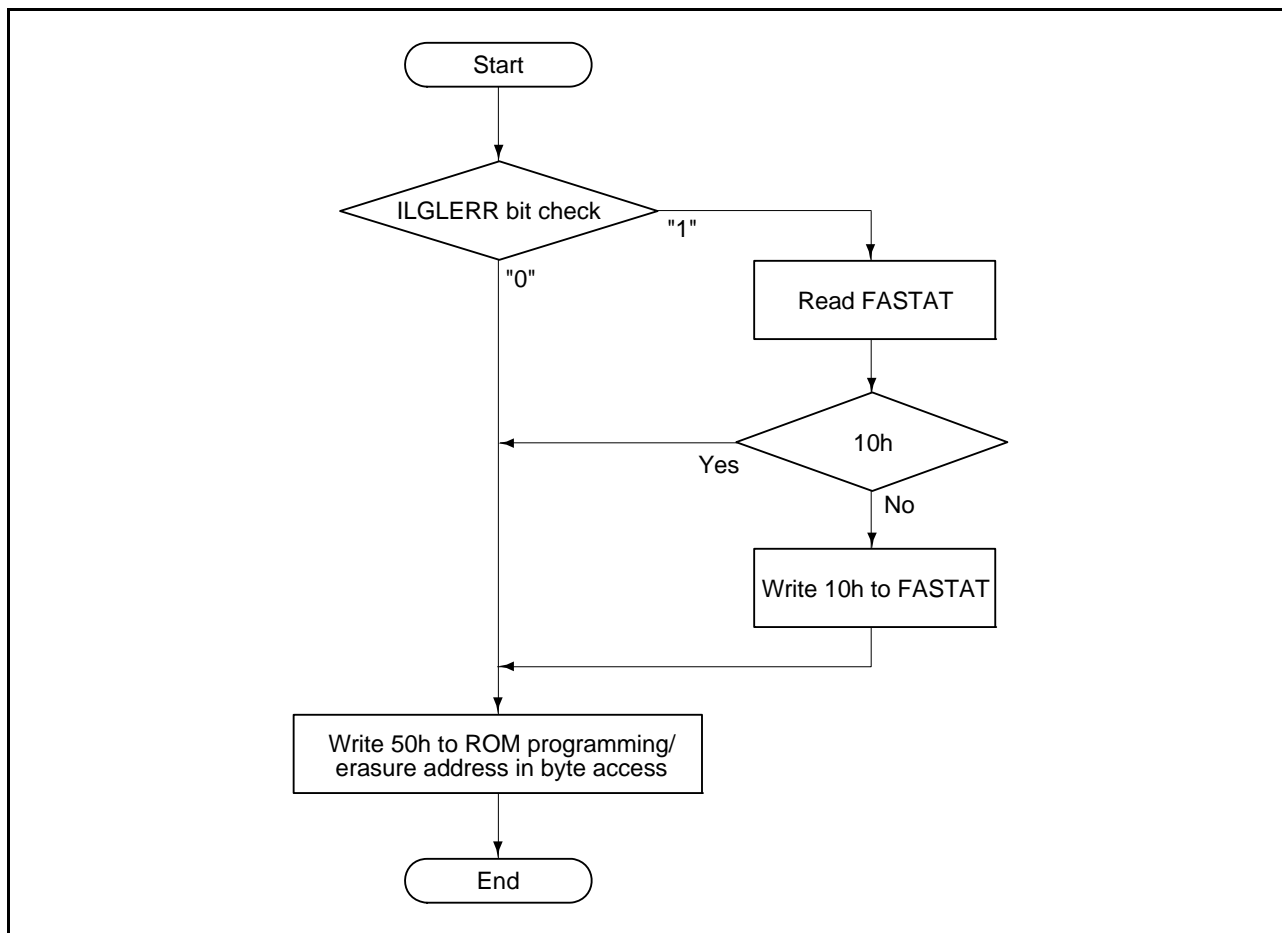


Figure 38.18 Procedure for Clearing FSTATR0

(3) Initializing the FPU

When a timeout leads to the FRDY bit in FSTATR0 not being set to 1 after an FCU command has been issued, FRESETR must be used to initialize the FCU. This is also necessary when the FCUERR bit in FSTATR1 has been set. In either case, maintain the FRESET bit in FRESETR at logical one over a period of at least tRESW2 (see section 41, Electrical Characteristics). Disable reading from the ROM and data-flash memory over this period. In addition, while the FRESET bit is 1, FCU commands are disabled because the FENTRYR register is initialized. Restart the processing from the start, as shown in Figure 38.11.

38.6.4.4 Suspension and Resumption

(1) Suspending Programming or Erasure

To suspend programming/erasure for the ROM, use the P/E suspend command.

When issuing a P/E suspend command, check that the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FCUERR bit in FSTATR1 are 0, and the execution of programming/erasure is normally performed. To confirm that the suspend command can be received, also check that the SUSRDY bit in FSTATR0 is 1. After issuing a P/E suspend command, read FSTATR0 and FSTATR1 to confirm that no error occurs.

If an error occurs during programming/erasure, at least one of the ILGLERR, PRGERR, ERSERR, and FCUERR bits is set to 1. When programming/erasure processing has finished during the interval from when it is checked that the SUSRDY bit is 1 to when a P/E suspend command is received, the ILGLERR bit is set to 1 because the issued P/E suspend command is detected as an illegal command.

When programming/erasure processing has finished simultaneously with the reception of a P/E suspend command, no error occurs and the suspended state is not entered (the FRDY bit in FSTATR0 is 1 and the ERSSPD and PRGSPD bits in FSTATR0 are 0). When a P/E suspend command is received and then the programming/erasure suspend processing finishes normally, the FCU enters the suspended state, the FRDY bit is set to 1, and the ERSSPD or PRGSPD bit is set to 1. After issuing a P/E suspend command, check that the ERSSPD or PRGSPD bit is 1 and the FCU enters the suspended state, and then decide the subsequent flow. When issuing a P/E resume command in the subsequent flow although the FCU does not enter the suspended state, an illegal command error occurs and the FCU is placed in the command-locked state (see section 38.8.2, Error Protection).

If the erasure suspended state is entered, programming to blocks other than an erasure target can be performed.

Additionally, the programming and erasure suspended states can change to ROM read mode by clearing FENTRYR.

For details on FCU operations at the reception of a P/E suspend command, see section 38.7, Suspending Operation.

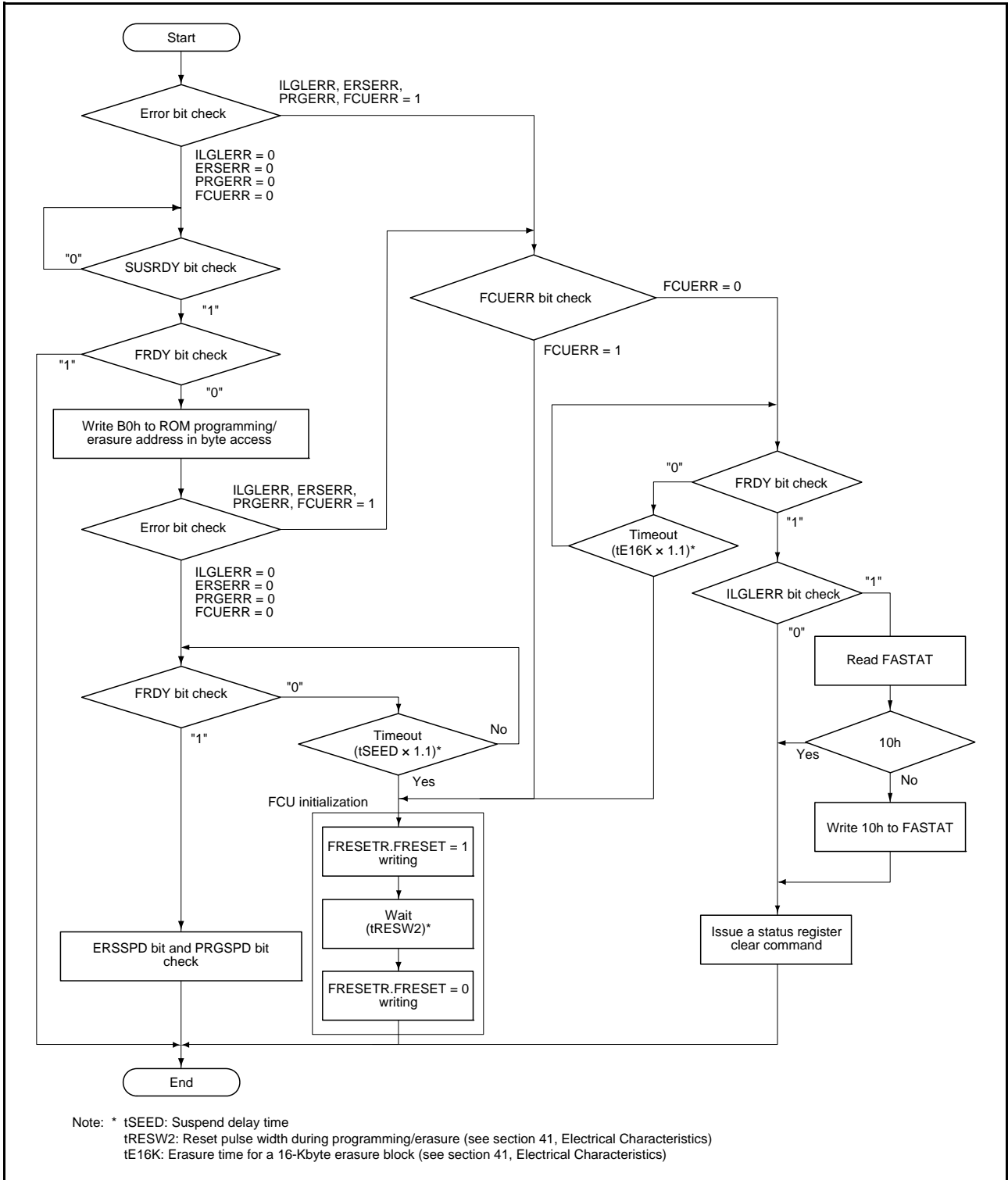


Figure 38.19 Procedure for Programming/Erasure Suspension

(2) Resuming Programming or Erasure

To resume a suspended programming/erasure processing, use the P/E resume command. When the settings of FENTRYR are changed during suspension, reset FENTRYR to the value immediately before a P/E suspend command is issued, and then issue a P/E resume command.

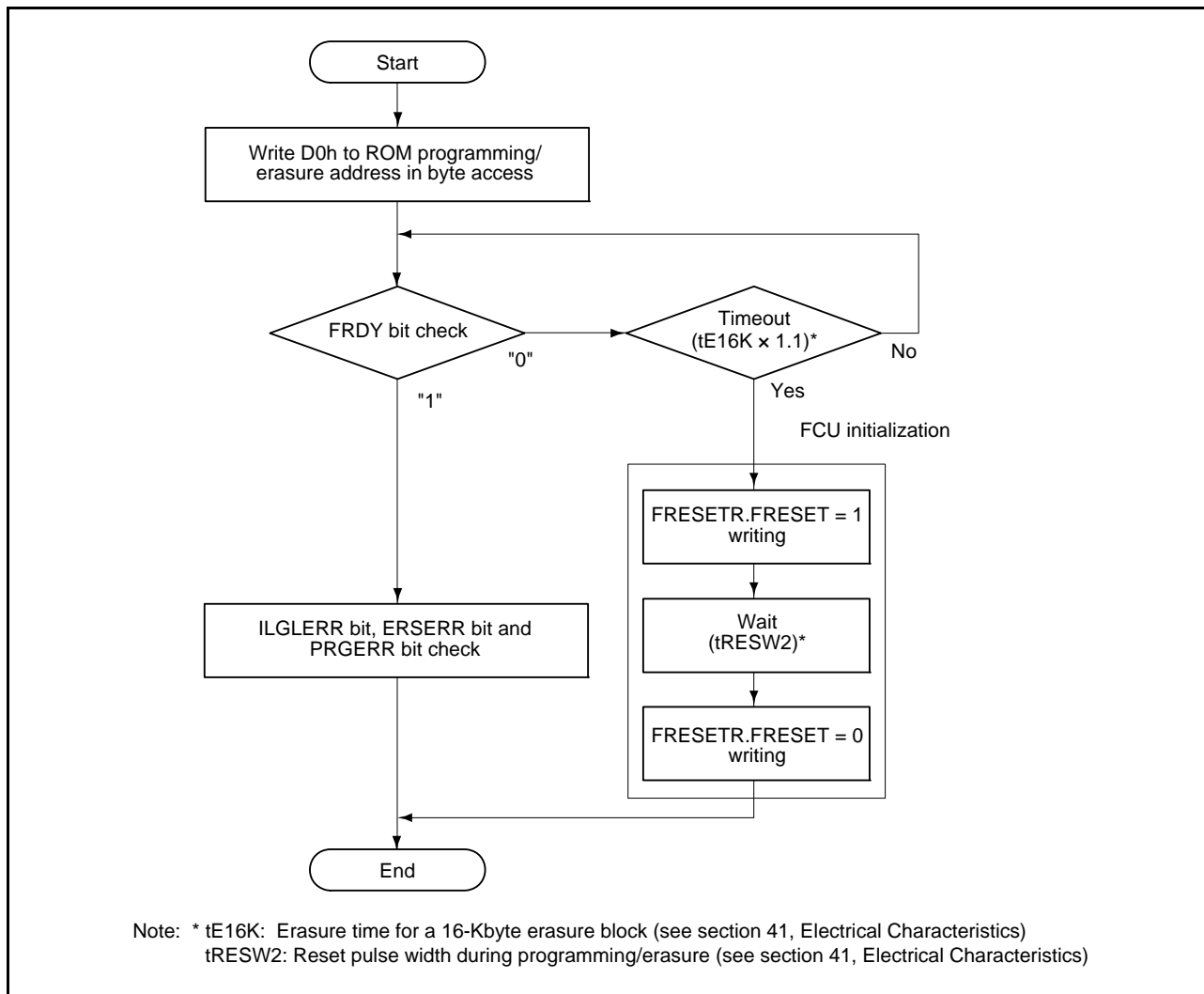


Figure 38.20 Procedure for Resuming Programming or Erasure

38.7 Suspending Operation

The ROM cannot be read out during programming/erasure. The ROM can be read out by suspending the ROM programming/erasure with the P/E suspend command. The P/E suspend command includes one programming mode and two erasure modes (suspension priority mode and erasure priority mode). The P/E resume command that resumes suspended programming/erasure processing is also provided.

38.7.1 Suspension during Programming

When issuing a P/E suspend command during the ROM programming/erasure, the FCU suspends programming processing. Figure 38.21 shows the suspend operation of programming.

When receiving a programming-related command, the FCU clears the FRDY bit in FSTATR0 to 0 to start programming. If the FCU enters the state in which the P/E suspend command can be received after starting programming, the SUSRDY bit in FSTATR0 is set to 1. When a P/E suspend command is issued, the FCU receives the command and clears the SUSRDY bit to 0. If the FCU receives a P/E suspend command while a programming pulse is being applied, the FCU continues applying the pulse. After specified pulse application time, the FCU finishes pulse application, and starts the programming suspend processing and sets the PRGSPD bit in FSTATR0 to 1. When the suspend processing finishes, the FCU sets the FRDY bit to 1 to enter the programming suspended state. If receiving a P/E resume command in the programming suspended state, the FCU clears the FRDY and PRGSPD bits to 0 and resumes programming.

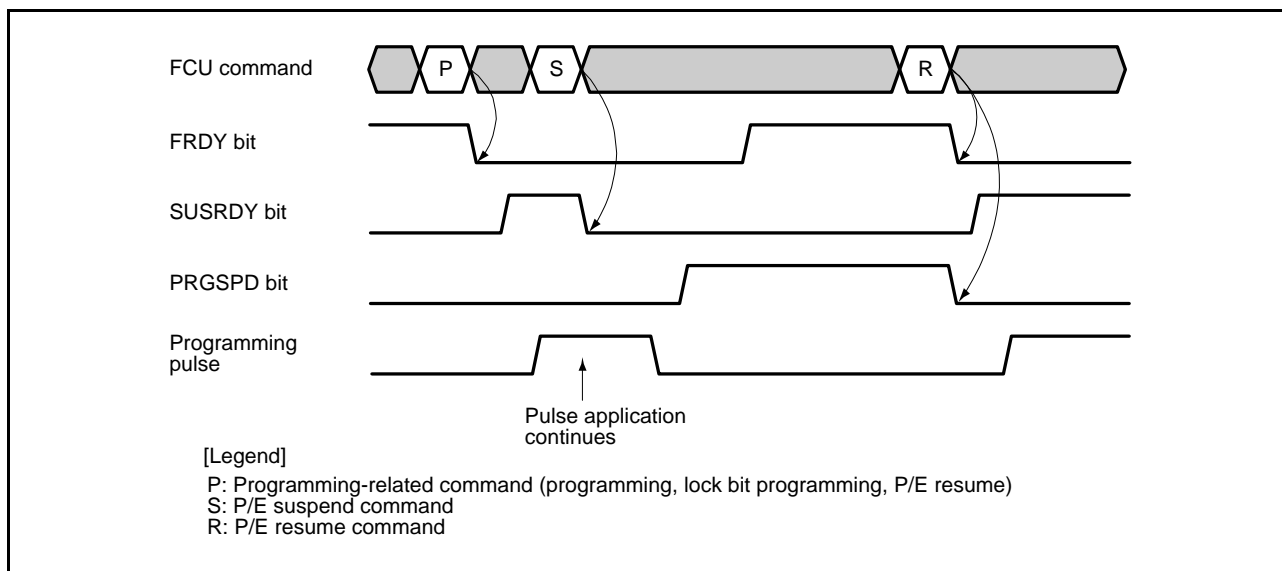


Figure 38.21 Suspension during Programming

38.7.2 Suspension during Erasure (Suspension Priority Mode)

Figure 38.22 shows the suspend operation of erasure when the erasure suspend mode is set to the suspension priority mode (ESUSPMD bit in FCPSR is 0).

When receiving an erasure-related command, the FCU clears the FSTATR0.FRDY bit to 0 to start erasure. If the FCU enters the state in which the P/E suspend command can be received after starting erasure, the FSTATR0.SUSRDY bit is set to 1. When a P/E suspend command is issued, the FCU receives the command and clears the SUSRDY bit to 0. When receiving a suspend command during erasure, the FCU starts the suspend processing and sets the ERSSPD bit in FSTATR0 to 1 even if it is applying an erasure pulse. When the suspend processing finishes, the FCU sets the FRDY bit to 1 to enter the erasure suspended state. If receiving a P/E resume command in the erasure suspended state, the FCU clears the FRDY and ERSSPD bits to 0 and resumes erasure. Operations of the FRDY, SUSRDY, and ERSSPD bits at the suspension and resume of erasure are the same, regardless of the erasure suspend mode.

The setting of the erasure suspend mode affects the control method of erasure pulses. In suspension priority mode, when receiving a P/E suspend command while erasure pulse A that has never been suspended in the past is being applied, the FCU suspends the application of erasure pulse A and enters the erasure suspended state. When receiving a P/E suspend command while reapplying erasure pulse A after erasure is resumed by a P/E resume command, the FCU continues applying erasure pulse A. After specified pulse application time, the FCU finishes erasure pulse application and enters the erasure suspended state. When the FCU receives a P/E resume command next and erasure pulse B starts to be newly applied, and then the FCU receives a P/E suspend command again, the application of erasure pulse B is suspended. In suspension priority mode, delay due to suspend can be minimized because the application of an erasure pulse is suspended one time per pulse and priority is given to the suspension processing.

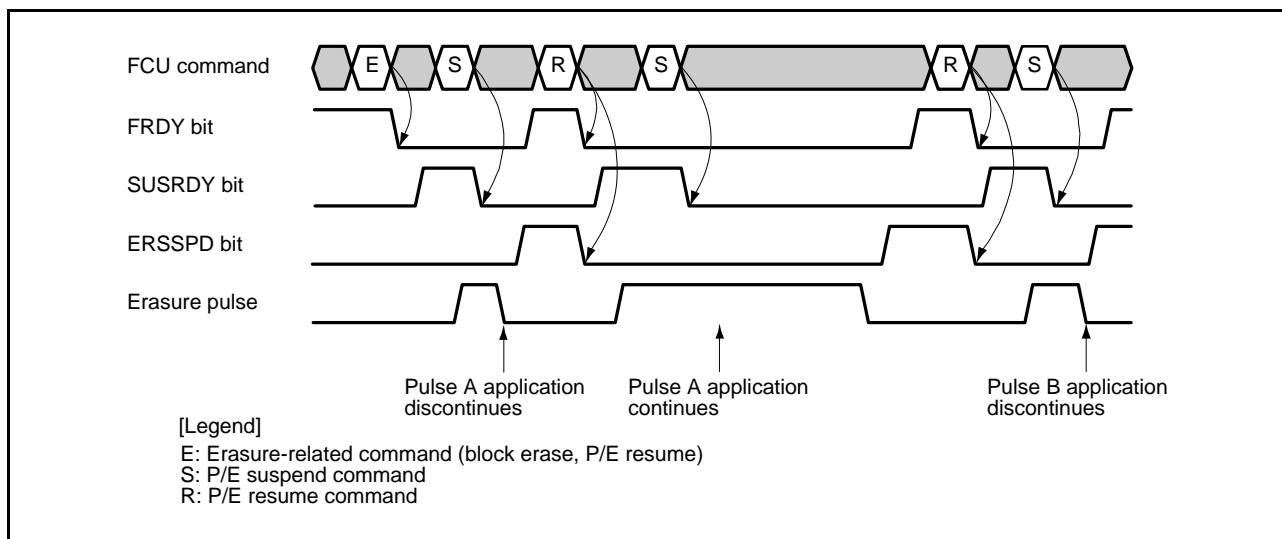


Figure 38.22 Suspension during Erasure (Suspension Priority Mode)

38.7.3 Suspension during Erasure (Erasure Priority Mode)

Figure 38.23 shows the suspend operation of erasure when the erasure suspend mode is set to the erasure priority mode (ESUSPMD bit in FCPSR is 1). The control method of erasure pulses in erasure priority mode is the same as that of programming pulses for the programming suspend processing.

If the FCU receives a P/E suspend command while an erasure pulse is being applied, the FCU definitely continues applying the pulse. In this mode, required time for the whole erasure processing can be reduced as compared with the suspension priority mode because the reapplication of erasure pulses does not occur when a P/E resume command issued.

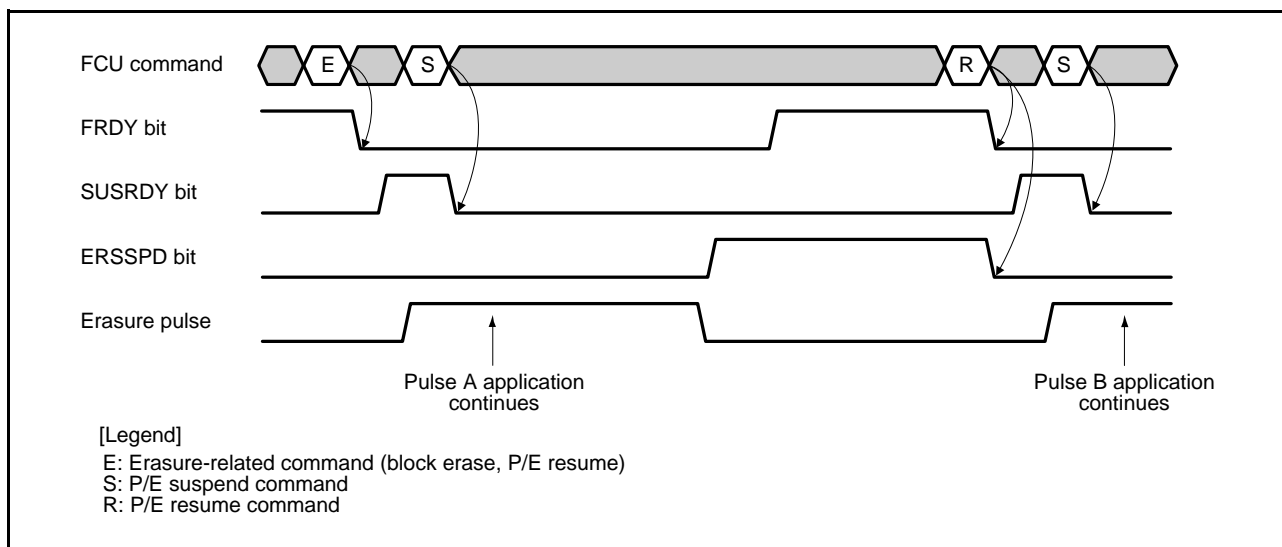


Figure 38.23 Suspension during Erasure (Erasure Priority Mode)

38.8 Protection

Protection against programming/erasure for the ROM includes software protection and error protection.

38.8.1 Software Protection

With the software protection, the ROM programming/erasure is prohibited by the settings of the control register or user mat lock bit. When the software protection is violated and a ROM programming/erasure-related command is issued, the FCU detects an error and enters the command-locked state.

(1) Protection through FWEPROR

If the FLWE[1:0] bits in FWEPROR are not set to 01b, programming cannot be performed in any of the modes.

(2) Protection through FENTRYR

When the FENTRY0 bit in FENTRYR is 0, ROM read mode is selected. Because the FCU command cannot be received in ROM read mode, ROM programming/erasure is prohibited. When an FCU command is issued in ROM read mode, the FCU detects an illegal command error and is placed in the command-locked state (see section 38.8.2, Error Protection).

(3) Protection through Lock Bit

Each erasure block in the user mat includes a lock bit. When the FPROTCN bit in FPROTR is 0, erasure blocks whose lock bit is set to 0 are prohibited from being programmed/erased. To program or erase erasure blocks whose lock bit is set to 0, set the FPROTCN bit to 1. When the lock bit protection is violated and a ROM programming/erasure-related command is issued, the FCU detects a programming/erasure error and enters the command-locked state (see section 38.8.2, Error Protection).

38.8.2 Error Protection

With the error protection, FCU command issuance errors, prohibited access occurrences, and FCU malfunctions are detected, and an FCU command is prohibited from being received (command-locked state). When the FCU enters the command-locked state (FASTAT.CMDLK bit is 1), one or several of the status bits (FSTATR0.ILGLERR, ERSERR, and PRGERR bits, FSTATR1.FCUEERR bit, and FASTST.ROMAE bit) are set to 1 and programming and erasure of the ROM are prohibited. To clear the command-locked state, a status register clear command must be issued with FASTAT set to 10h.

While the CMDLKIE bit in FAEINT is set to 1, if the FCU is placed in the command-locked state (CMDLK bit in FASTAT is set to 1), a flash interface error (FIFERR) interrupt occurs. While the ROMAEIE bit in FAEINT is set to 1, if the ROMAE bit in FASTAT is set to 1, an FIFERR interrupt occurs.

Table 38.9 lists the relationship between the contents of the ROM-related error protection and status bit values (ILGLERR, ERSERR, PRGERR bits in FSTATR0, FCUEERR bit in FSTATR1, ROMAE bit in FASTAT) at error detection. If a command other than the suspend command is issued during programming/erasure and the FCU enters the command-locked state, it continues the programming/erasure. In this state, it is impossible to issue a P/E suspend command and suspend programming/erasure. When a command is issued in the command-locked state, the ILGLERR bit is set to 1.

Table 38.9 Error Protection Types (Types Dedicated to ROM and Types Common to ROM and Data Flash)

Type	Description	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE	CMDLK
FENTRYR setting error	More than one bit is set to 1 among the FENTRYD and FENTRY0 bits in FENTRYR	1	0	0	0	0	1
	The FENTRYR setting at suspension disagrees with that at resume	1	0	0	0	0	1
Illegal command error	Undefined code is specified in the first cycle of an FCU command	1	0	0	0	0	1
	Other than D0h is specified in the last cycle of a multi-cycle FCU command	1	0	0	0	0	1
	The peripheral clock is set to other than 8 to 50 MHz in PCKAR	1	0	0	0	0	1
	A command other than the suspend command is issued during programming/erasure	1	0	0	0	0	1
	A suspend command is issued during processing other than programming/erasure	1	0	0	0	0	1
	A suspend command is issued in the suspended state	1	0	0	0	0	1
	A resume command is issued in other than the suspended state	1	0	0	0	0	1
	A programming/erasure-related (programming/lock bit programming/block erase) command is issued in the programming suspended state	1	0	0	0	0	1
	A block erase command is issued in the erasure suspended state	1	0	0	0	0	1
	A programming or lock bit programming command is issued to an erasure suspend target area in the erasure suspended state	1	0	0	0	0	1
	Other than 80h is specified in the second cycle of the programming command	1	0	0	0	0	1
	A command is issued in the command-locked state	1	0/1	0/1	0/1	0/1	1
Erasure error	An error occurs during erasure	0	1	0	0	0	1
	When the FPROTCN bit in FPROTR is 0, a block erase command is issued to a erasure block whose lock bit is set to 0	0	1	0	0	0	1
Programming error	An error occurs during programming	0	0	1	0	0	1
	When the FPROTCN bit in FPROTR is 0, a programming or lock bit programming command is issued to a erasure block whose lock bit is set to 0	0	0	1	0	0	1
FCU error	An error occurs during FCU internal processing	0	0	0	1	0	1
ROM access violation	When the FENTRY0 bit in FENTRYR is 1 and the FCU is in P/E normal mode, a read command is issued for addresses 00F8 0000h to 00FF FFFFh	1	0	0	0	1	1
	When the FENTRY0 bit in FENTRYR is 0, a command is issued for addresses 00F8 0000h to 00FF FFFFh	1	0	0	0	1	1
	A read command is issued for addresses FFF8 0000h to FFFF FFFFh when FENTRYR is set to the ROM P/E mode.	1	0	0	0	1	1

38.9 Boot Mode

38.9.1 System Configuration

In boot mode, the host sends control commands and data for programming, and the user mat, user boot mat, and data mat are programmed or erased accordingly. An on-chip SCI handles transfer between the host and RX62N/RX621 in asynchronous mode. Tools for the transmission of control commands and the data for programming must be prepared in the host.

When the RX62N/RX621 is activated in boot mode, the program on the mat that holds the embedded program is executed. This program automatically adjusts the bit rate of the SCI and controls programming/erasure by receiving control commands from the host.

Figure 38.24 shows the system configuration for operations in boot mode.

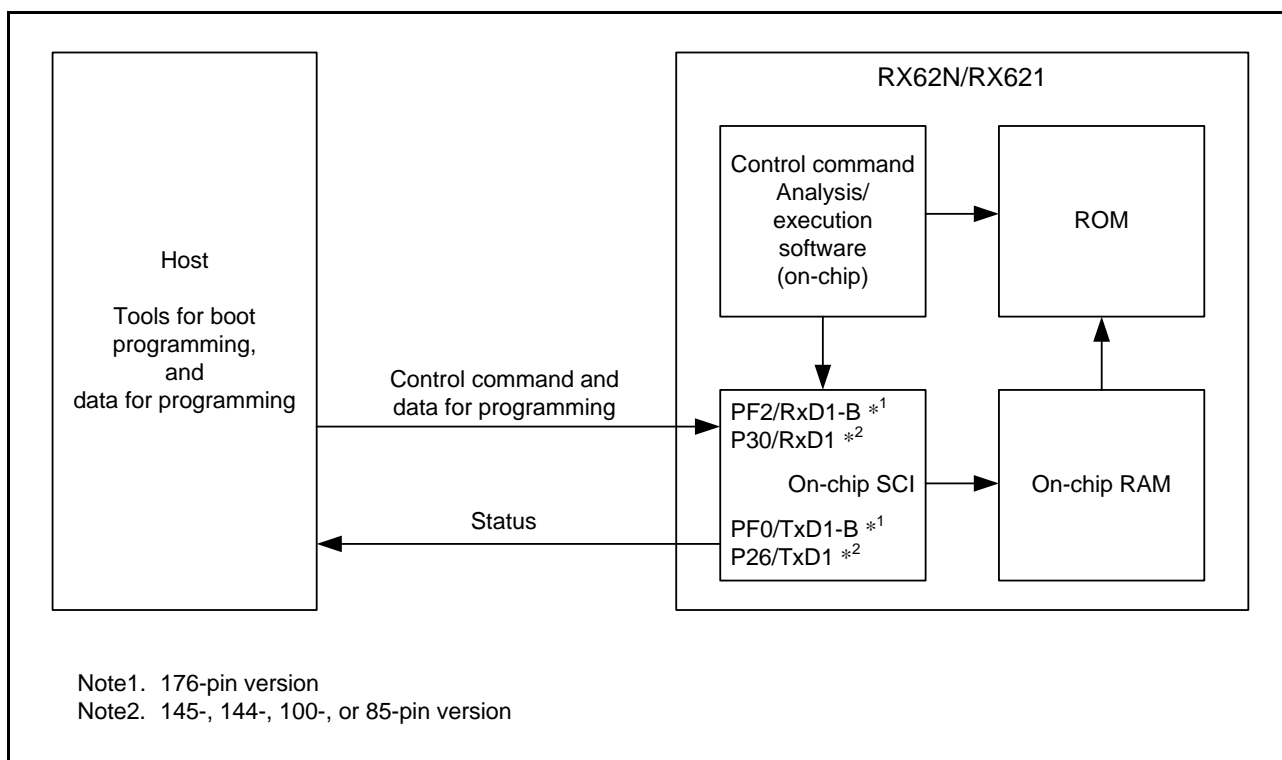


Figure 38.24 System Configuration for Operations in Boot Mode

38.9.2 ID Code Protection

This function is used to prohibit reading/programming/erasure from the host.

Using the control code and ID code written in the ROM, ID code protection is enabled or disabled and ID code protection is judged. When ID code protection is enabled, the code sent from the host is compared with the control code and ID code in the ROM to determine whether they match, and reading/programming/erasure will be enabled only when the two match.

The control code and ID code in the ROM consists of four 32-bit words. Figure 38.25 shows the configuration of the control code and ID code. The ID code should be set in 32-bit units.

	31	24	23	16	15	8	7	0
FFFF FFA0h	Control code		ID code 1		ID code 2		ID code 3	
FFFF FFA4h	ID code 4		ID code 5		ID code 6		ID code 7	
FFFF FFA8h	ID code 8		ID code 9		ID code 10		ID code 11	
FFFF FFACH	ID code 12		ID code 13		ID code 14		ID code 15	

Figure 38.25 Configuration of Control Code and ID Code in ROM

(1) Control Code

The control code determines whether ID code protection is or is not enabled and the method of authentication to use with the host. Table 38.10 shows how the control code determines the method of authentication

Table 38.10 Specifications for ID Code Protection

Control Code	ID Code	State of Protection	Operations at the Time of SCI Connection
45h	As desired	Protection enabled (authentication method 1)	Matching ID code: ID code protection is lifted, and this is followed by a transition to the state of waiting for a host command. Non-matching ID code: Up to two more transitions to the ID code protection waiting state; complete erasure if a non-matching ID code is received for a third time.
52h	Sequences other than 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh	Protection enabled (authentication method 2)	Matching ID code: ID code protection is lifted, and this is followed by a transition to the state of waiting for a host command. Non-matching ID code: Further transition to the ID code protection waiting state
	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh	Protection enabled (authentication method 3)	Always judged to be a non-matching ID code.
Other than the above	—	Protection disabled	Erasure of all blocks

(2) ID Code

The ID code can be set to any desired value. However, if the control code is 52h and the ID code is 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh (from the ID code 1 field), there is no determination of matching and the ID code is always considered to be non-matching. Accordingly, reading, programming, and erasure from the host are prohibited.

(3) Example of Assembler Directives for Setting an ID Code

The following assembler directives set up a control code of 45h and an ID code of 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh (from the ID code 1 field).

```
.SECTION    ID_CODE, CODE
.ORG       0FFFFFFA0h
.LWORD    45010203h
.LWORD    04050607h
.LWORD    08090A0Bh
.LWORD    0C0D0E0Fh
```

38.9.3 UB Code A

The UB code A is 32-bit or 2-word data which determines a program on the user boot mat. The UB code A on the user boot mat is used to determine a program on the user boot mat. If the program is the USB boot program, the user boot mat is not erased by erasure of all blocks at the time of SCI connection. Figure 38.26 shows a structure of the UB code A. The UB code A should be specified in 32-bit units.

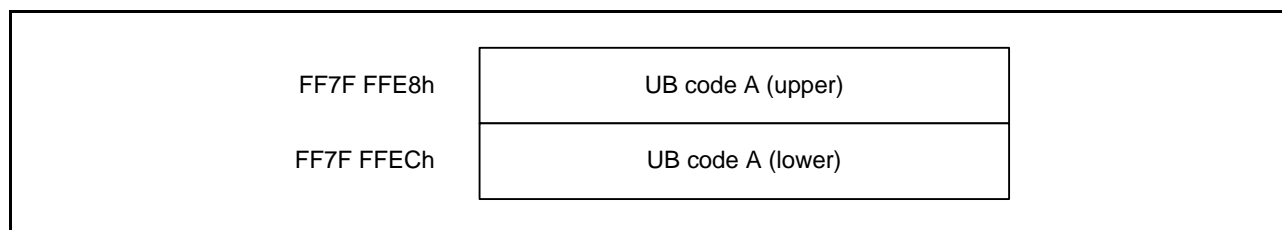


Figure 38.26 UB Code A Configuration

Table 38.11 UB Code A Specifications

UB Code A	Program on User Boot Mat	Erasure of All Blocks at the Time of SCI Connection
55736242h, 6F6F74FFh	USB boot program	User boot mat is not erased
Other than above	Other than USB boot program	User boot mat is erased

38.9.4 State Transitions in Boot Mode

Figure 38.27 is a diagram of the state transitions in boot mode.

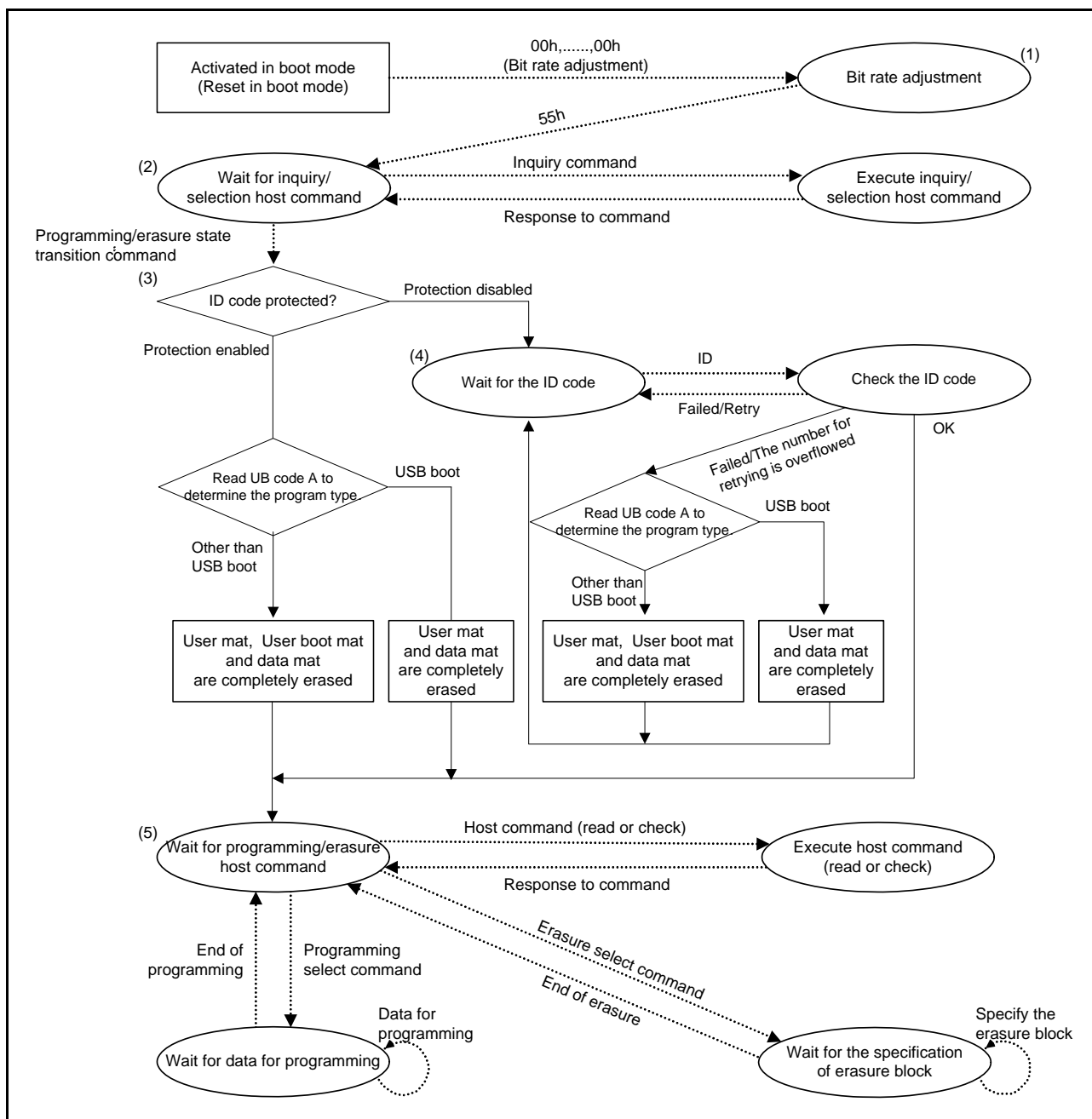


Figure 38.27 State Transitions in Boot Mode

(1) Matching the Bit Rates

When the RX62N/RX621 is activated in boot mode, the bit rate of the SCI is automatically adjusted to match that of the host. On completion of this adjustment, the RX62N/RX621 transmits the value 00h to the host. On subsequent correct reception of the value 55h sent from the host, the RX62N/RX621 enters the state of waiting for a host command for inquiry or selection. For details on matching of the bit rates, see section 38.9.5, Automatic Adjustment of the Bit Rate.

(2) Waiting for a Host Command for Inquiry or Selection

This state is for inquiries on mat size, mat configuration, the addresses where mats start, the state of support etc., and for selection of the device, clock mode, and bit rate. The RX62N/RX621 receives a programming/erasure state transition command issued by the host and then enters the state to determine whether ID code protection is enabled or disabled. For the inquiry/selection host commands, see section 38.9.6, Inquiry/Selection Host Command Wait State.

(3) Judging ID Code Protection

This state is for determining whether ID code protection is enabled or disabled. The control code and ID code written in the ROM are used to determine whether ID code protection is enabled or disabled. When enabled, the state of waiting for the ID code is entered. When disabled, the user mat, user boot mat, and data mat are all completely erased, and the state of waiting for programming and erasure commands from the host is entered. For details on the control code and ID code, see section 38.9.2, ID Code Protection.

(4) Waiting for an ID Code

This state is for waiting for the control code and ID code to be sent from the host. The control code and ID code sent by the host are compared with the code stored in the ROM, and the state of waiting for programming and erasure commands from the host is entered if the two match. If they do not match, the next transition is back to the state of waiting for an ID code. However, if the ID codes fail to match three times in a row and also the state of protection is authentication method 1, the ROM is completely erased, and the state of waiting for an ID code is entered again. A reset is required to release the system from this state due to non-matching ID codes. For details on the control code and ID code, see section 38.9.2, ID Code Protection.

(5) Waiting for a Host Command for Programming or Erasure

In this state, programming and erasure proceed in accordance with commands from the host. In response to the reception of a command, the RX62N/RX621 enters the state of waiting for the data to use in programming, waiting for specification of the erasure block to be erased, or executing the processing of commands for reading, fetching and so on. When the RX62N/RX621 receives a programming selection command, it enters the state of waiting for the data to use in programming. After the host has issued the programming selection command, the process continues with the address where programming is to start and then the data for programming. Setting of FFFF FFFFh as the address where programming is to start indicates the completion of programming, and the next transition is from the state of waiting for the data to use in programming to the state of waiting for programming and erasure commands.

When the RX62N/RX621 receives a programming selection command, it enters the state of waiting for specification of the erasure block to be erased. After the host has issued the programming selection command, the process continues with the number of the erasure block to be erased. Setting of FFh as the number of the erasure block indicates the completion of erasure, and the next transition is from the state of waiting for specification of the erasure block to the state of waiting for programming and erasure commands. Since the user mat, user boot mat, and data mat are all completely erased during the interval between booting up in boot mode and transition to the state of waiting for programming and erasure commands, explicit execution of erasure is not necessary unless newly programmed data are to be erased without a further reset.

Other than the programming and erasure commands, commands from the host for execution in this state include those for sum checking of the user mat and user boot mat, blank checking (to confirm erasure), reading from memory, and acquiring state information.

38.9.5 Automatic Adjustment of the Bit Rate

When the RX62N/RX621 is booted up in boot mode, asynchronous transfer by the SCI is used to measure the periods at low level of consecutive bytes with value 00h that are sent from the host. While the period at low level is being measured, set the host's SCI parameters to eight-bit data, one stop bit, no parity, and a transfer rate of 9,600 bps or 19,200 bps. The RX62N/RX621 calculates the host's SCI bit rate from the measured periods at low level, adjusts its own bit rate accordingly, and then sends a 00h byte to the host. If reception of the value 00h by the host is normal, the host responds by sending the value 55h to the RX62N/RX621. If normal reception of 00h by the host is not possible, the RX62N/RX621 is re-booted in boot mode, and then repeats the process of automatically adjusting the bit rate. If reception of the value 55h by the RX62N/RX621 is normal, it responds by sending E6h to the host, and if normal reception of 55h by the RX62N/RX621 is not possible, it responds by sending FFh to the host.

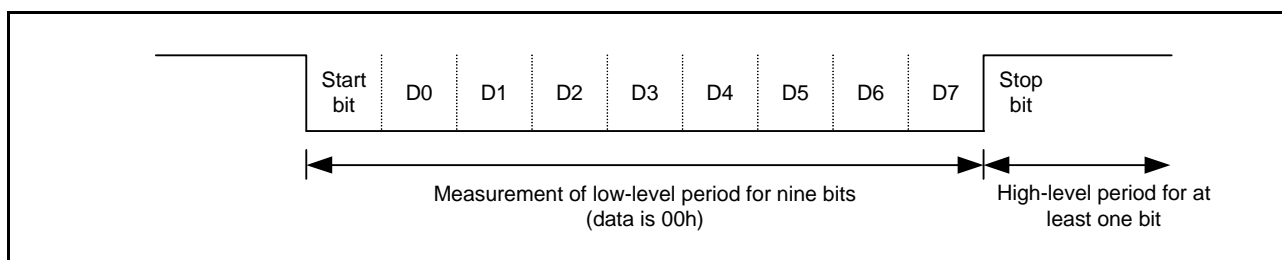


Figure 38.28 Transfer Format Used by the SCI in Automatic Adjustment of the Bit Rate

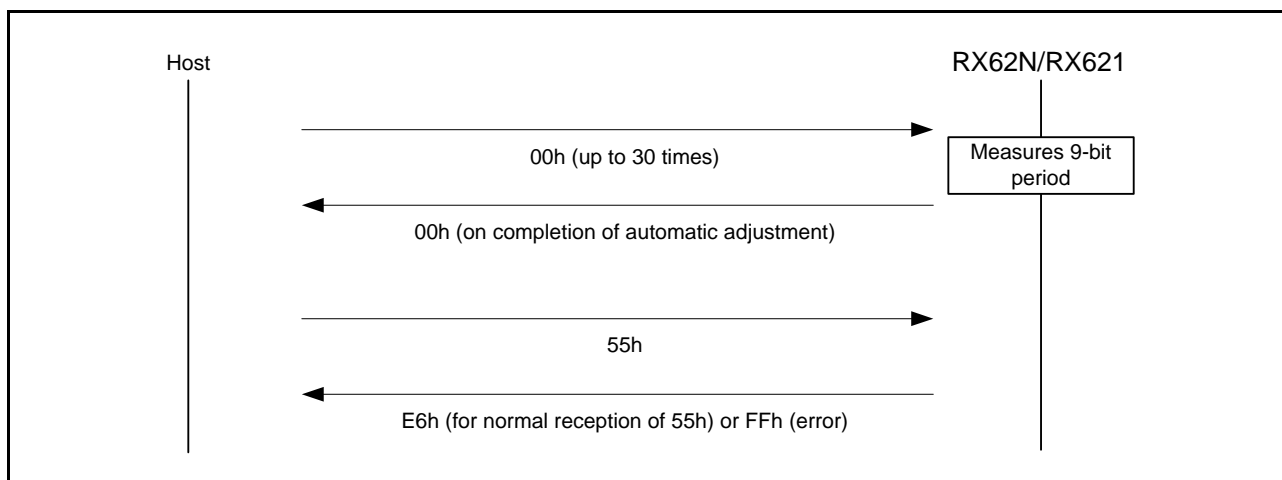


Figure 38.29 Sequence of Transfer between the Host and RX62N/RX621

Since the bit rate of the RX62N/RX621's SCI module depends on the frequency of the peripheral clock, adjustment to match the bit rate of the host will not be possible under some conditions. Accordingly, ensure that SCI transfer is under the conditions given in Table 38.12.

Table 38.12 Conditions for Automatic Bit-Rate Adjustment to be Possible

Bit Rate of the SCI in the Host	Range of Frequency for the EXTAL Signal
9,600 bps	8 to 14 MHz
19,200 bps	8 to 14 MHz

38.9.6 Inquiry/Selection Host Command Wait State

Table 38.13 shows the host commands available in the inquiry/selection host command wait state. The embedded program status inquiry command can also be used in the programming/erasure host command wait state. The other commands can only be used in the inquiry/selection host command wait state.

Table 38.13 Inquiry/Selection Host Commands

Host Command Name	Function
Supported device inquiry	Inquires regarding the device codes and the product codes for the embedded programs
Device selection	Selects a device code
Clock mode inquiry	Inquires regarding the clock mode
Clock mode selection	Notifies the selected clock mode
Multiplication ratio inquiry	Inquires regarding the number of clock types, the number of multiplication/division ratios, and the multiplication /division ratios
Operating frequency inquiry	Inquires regarding the number of clock types and the maximum and minimum operating frequencies
User boot mat information inquiry	Inquires regarding the number of user boot mats and the start and end addresses
Erasure block information inquiry	Inquires regarding the number of blocks and the start and end addresses
Programming size inquiry	Inquires regarding the size of programming data
New bit rate selection	Modifies the bit rate of SCI communications between the host and RX62N/RX621
Programming/erasure state transition	Enters the state for determining ID code protection
Embedded program status inquiry	Inquires regarding the processing state

If the host has sent an undefined command, the RX62N/RX621 returns a response indicating a command error in the format shown below. The command field holds the first byte of the undefined command sent from the host.

Error response

80h	Command
-----	---------

In the inquiry/selection host command wait state, send selection commands from the host in the order of device selection, clock mode selection, and new bit rate selection to set up the RX62N/RX621 according to the responses to inquiry commands. Note that the supported device inquiry and clock mode inquiry commands are the only inquiry commands that can be sent before the clock mode selection command; other inquiry commands must not be issued before the clock mode selection command. If commands are issued in an incorrect order, the RX62N/RX621 returns a response indicating a command error. Figure 38.30 shows an example of the procedure to use inquiry/selection host commands.

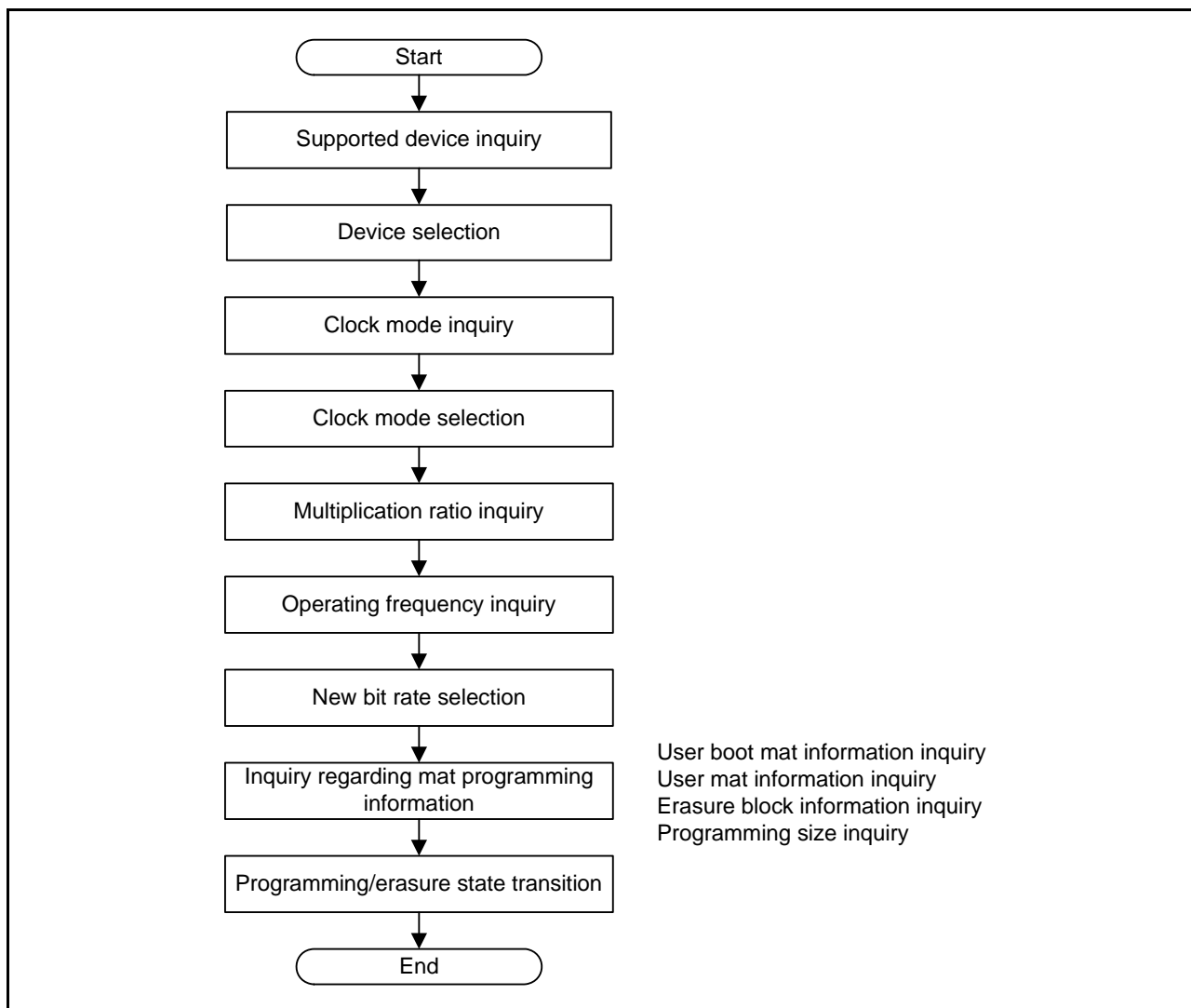


Figure 38.30 Example of Procedure to Use Inquiry/Selection Host Commands for User Mat and User Boot Mat

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to the RX62N/RX621 and the "response" indicates a response sent from the RX62N/RX621 to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by the RX62N/RX621 becomes 00h.

(1) Supported Device Inquiry

In response to a supported device inquiry command sent from the host, the RX62N/RX621 returns the information concerning the devices supported by the embedded program for boot mode. If the supported device inquiry command comes after the host has selected a device, the RX62N/RX621 only returns the information concerning the selected device.

Command	20h			
Response	30h	Size	Device count	
	Character count	Device code		Product code
	Character count	Device code		Product code

	Character count	Device code		Product code
	SUM			

[Legend]

- Size (1 byte): Total number of bytes in the device count, character count, device code, and product code fields
- Device count (1 byte): Number of device types supported by the embedded program for boot mode
- Character count (1 byte): Number of characters included in the device code and product code fields
- Device code (4 bytes): ASCII code for the product name of the chip
- Product code (n bytes): ASCII code for the supported device
- SUM (1 byte): Checksum (in response)

(2) Device Selection

In response to a device selection command sent from the host, the R RX62N/RX621 checks if the selected device is supported. When the selected device is supported, the RX62N/RX621 specifies this device as the device for use and returns a response (06h). If the selected device is not supported or the sent command is illegal, the RX62N/RX621 returns an error response (90h).

Even when 01h has been returned as the number of supported devices in response to a supported device inquiry command, issue a device selection command to specify the device code that has been returned as the result of the inquiry.

Command	10h	Size	Device code	SUM
---------	-----	------	-------------	-----

Response	06h
----------	-----

Error response	90h	Error
----------------	-----	-------

[Legend]

- Size (1 byte): Number of characters in the device code field (fixed at four)
- Device code (4 bytes): ASCII code for the product name of the chip (one of the device codes returned in response to the supported device inquiry command)
- SUM (1 byte): Checksum
- Error (1 byte): Error code
 - 11h: Checksum error (illegal command)
 - 21h: Incorrect device code error

(3) Clock Mode Inquiry

In response to a clock mode inquiry command sent from the host, the RX62N/RX621 returns the supported clock modes. If the clock mode inquiry command comes after the host has selected a clock mode, the RX62N/RX621 only returns the information concerning the selected clock mode.

Command	21h
---------	-----

Response	31h	Size		
	Mode	Mode	...	Mode
	SUM			

[Legend]

- Size (1 byte): Total number of bytes in the mode count and mode fields
- Mode (1 byte): Supported clock mode (for example, 01h indicates clock mode 1)
- SUM (1 byte): Checksum

(4) Clock Mode Selection

In response to a clock mode selection command sent from the host, the RX62N/RX621 checks if the selected clock mode is supported. When the selected mode is supported, the RX62N/RX621 specifies this clock mode for use and returns a response (06h). If the selected mode is not supported or the sent command is illegal, the RX62N/RX621 returns an error response (91h).

Be sure to issue a clock mode selection command only after issuing a device selection command. Even when 00h or 01h has been returned as the number of supported clock modes in response to a clock mode inquiry command, issue a clock mode selection command to specify the clock mode that has been returned as the result of the inquiry.

Command	11h	Size	Mode	SUM
---------	-----	------	------	-----

Response	06h
----------	-----

Error response	91h	Error
----------------	-----	-------

[Legend]

- Size (1 byte): Number of characters in the mode field (fixed at 1)
- Mode (1 byte): Clock mode (same mode as the response to the clock mode inquiry command)
- SUM (1 byte): Checksum
- Error (1 byte): Error code
 - 11h: Checksum error (illegal command)
 - 22h: Incorrect clock mode error

(5) Multiplication Ratio Inquiry

In response to a multiplication ratio inquiry command sent from the host, the RX62N/RX621 returns the clock types, the number of multiplication/division ratios, and the multiplication division ratios supported.

Command	22h
---------	-----

Response	32h	Size	Clock type count		
	Multiplication ratio count	Multiplication ratio	Multiplication ratio	...	Multiplication ratio
	Multiplication ratio count	Multiplication ratio	Multiplication ratio	...	Multiplication ratio
	:	:	:	...	:
	Multiplication ratio count	Multiplication ratio	Multiplication ratio	...	Multiplication ratio
	SUM				

[Legend]

- Size (1 byte): Total number of bytes in the clock type count, multiplication ratio type, and multiplication ratio fields
- Clock type count (1 byte): Number of clock types (for example, 02h indicates two clock types; that is, a system clock and a peripheral clock)
- Multiplication ratio count (1 byte): Number of supported multiplication/division ratios (for example, 04h indicates that four multiplication ratios are supported for the system clock (x1, x2, x4, and x8))
- Multiplication ratio (1 byte): A positive value indicates a multiplication ratio (for example, 04h = 4 = multiplication by 4)
A negative value indicates a division ratio (for example, FEh = -2 = division by 2)
- SUM (1 byte): Checksum

(6) Operating Clock Frequency Inquiry

In response to an operating clock frequency inquiry command sent from the host, the RX62N/RX621 returns the minimum and maximum frequencies for each clock.

Command

23h

Response	33h	Size	Clock type count
	Minimum frequency		Maximum frequency
	Minimum frequency		Maximum frequency
	:		:
	Minimum frequency		Maximum frequency
	SUM		

[Legend]

- Size (1 byte): Total number of bytes in the clock type count, minimum frequency, and maximum frequency fields
- Clock type count (1 byte): Number of clock types (for example, 02h indicates two clock types; that is, a system clock and a peripheral clock)
- Minimum frequency (2 bytes): Minimum value of the operating frequency (for example, 07D0h indicates 20.00 MHz). This value should be calculated by multiplying the frequency value (MHz) to two decimal places by 100.
- Maximum frequency (2 bytes): Maximum value of the operating frequency represented in the same format as the minimum frequency
- SUM (1 byte): Checksum

(7) User Boot Mat Information Inquiry

In response to a user boot mat information inquiry command sent from the host, the RX62N/RX621 returns the number of user boot mat areas and their addresses.

Command

24h

Response	34h	Size	Area count
	Area start address		
	Area end address		
	Area start address		
	Area end address		
	:		
	Area start address		
	Area end address		
	SUM		

[Legend]

- Size (1 byte): Total number of bytes in the area count, area start address, and area end address fields
- Area count (1 byte): Number of user mat areas (consecutive areas are counted as one area)
- Area start address (4 bytes): Start address of a user mat area
- Area end address (4 bytes): End address of a user mat area
- SUM (1 byte): Checksum

(8) User Mat Information Inquiry

In response to a user mat information inquiry command sent from the host, the RX62N/RX621 returns the number of user mat areas and their addresses.

Command	25h
---------	-----

Response	35h	Size	Area count
Area start address			
Area end address			
Area start address			
Area end address			
:			
Area start address			
Area end address			
SUM			

[Legend]

- Size (1 byte): Total number of bytes in the area count, area start address, and area end address fields
- Area count (1 byte): Number of user mat areas (consecutive areas are counted as one area)
- Area start address (4 bytes): Start address of a user mat area
- Area end address (4 bytes): End address of a user mat area
- SUM (1 byte): Checksum

(9) Erasure Block Information Inquiry

In response to an erasure block information inquiry command sent from the host, the RX62N/RX621 returns the number of erasure blocks in the user mat and their addresses.

Command	26h
---------	-----

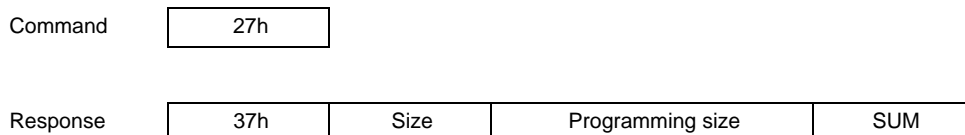
Response	36h	Size	Block count
Block start address			
Block end address			
Block start address			
Block end address			
:			
Block start address			
Block end address			
SUM			

[Legend]

- Size (2 bytes): Total number of bytes in the block count, block start address, and block end address fields
- Block count (1 byte): Number of erasure blocks in the user mat
- Block start address (4 bytes): Start address of an erasure block
- Block end address (4 bytes): End address of an erasure block
- SUM (1 byte): Checksum

(10) Programming Size Inquiry

In response to a programming size inquiry command sent from the host, the RX62N/RX621 returns the programming size.



[Legend]

Size (1 byte): Number of characters included in the programming size field (fixed at two)
 Programming size (2 bytes): Programming unit (bytes)
 SUM (1 byte): Checksum

(11) New Bit Rate Selection

In response to a new bit rate selection command sent from the host, the RX62N/RX621 checks if the on-chip SCI can be set to the selected new bit rate. When the SCI can be set to the new bit rate, the RX62N/RX621 returns a response (06h) and sets the SCI to the new bit rate. If the SCI cannot be set to the new bit rate or the sent command is illegal, the RX62N/RX621 returns an error response (BFh). Upon reception of response 06h, the host waits for a one-bit period in the previous bit rate with which the new bit rate selection command has been sent, and then sets the host's bit rate to the new one. After that, the host sends confirmation data (06h) in the new bit rate, and the RX62N/RX621 returns a response (06h) to the confirmation data.

Be sure to issue a new bit rate selection command only after a clock mode selection command.

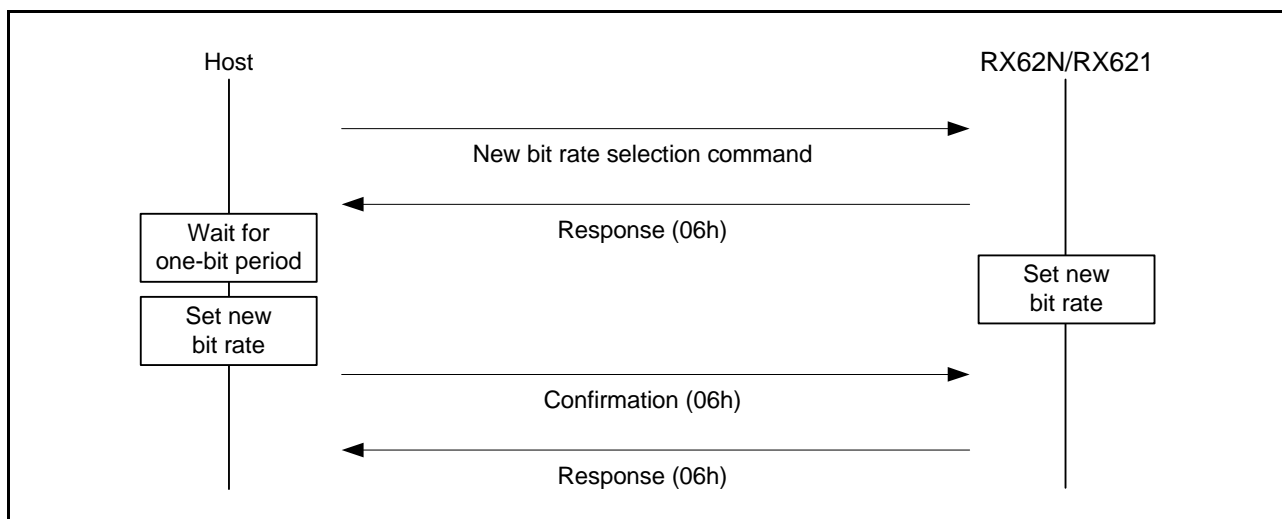


Figure 38.31 New Bit Rate Selection Sequence

Command	3Fh	Size	Bit rate		Input frequency
	Clock type count	Multiplication ratio 1	Multiplication ratio 2		
	SUM				
Response	06h				
Error response	BFh	Error			
Confirmation	06h				
Response	06h				

[Legend]

- Size (1 byte): Total number of bytes in the bit rate, input frequency, clock type count, and multiplication ratio fields
- Bit rate (2 bytes): New bit rate (for example, 00C0h indicates 19200 bps)
1/100 of the new bit rate value should be specified.
- Input frequency (2 bytes): Clock frequency input to the RX62N/RX621 (for example, 04E2h indicates 12.50 MHz)
This value should be calculated by multiplying the input frequency value to two decimal places by 100.
- Clock type count (1 byte): Number of clock types (for example, 02h indicates two clock types; that is, a system clock and a peripheral clock)
- Multiplication ratio 1 (1 byte): Multiplication/division ratio of the input frequency to obtain the system clock (ICLK)
A positive value indicates a multiplication ratio (for example, 04h = 4 = multiplication by 4)
A negative value indicates a division ratio (for example, FEh = -2 = division by 2)
- Multiplication ratio 2 (1 byte): Multiplication/division ratio of the input frequency to obtain the peripheral clock (PCLK)
This value is represented in the same format as multiplication ratio 1
- SUM (1 byte): Checksum
- Error: Error code
 - 11h: Checksum error
 - 24h: Bit rate selection error
 - 25h: Input frequency error
 - 26h: Multiplication ratio error
 - 27h: Operating frequency error

- Bit rate selection error

A bit rate selection error occurs when the bit rate selected through a new bit rate selection command cannot be set for the SCI of the RX62N/RX621 within an error of 4%. The bit rate error can be obtained by the following equation from the bit rate (B) selected through a new bit rate selection command, the input frequency (f_{EX}), multiplication ratio 2 (M_{Pφ}), the bit rate register (BRR) setting (N) in the SCI, and the CKS[1:0] bit value (n) in the serial mode register (SMR).

$$\text{Error (\%)} = \frac{f_{EX} \times M_{P\phi} \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1$$

- Input frequency error

An input frequency error occurs when the input frequency specified through a new bit rate selection command is outside the range from the minimum to maximum input frequencies for the clock mode selected through a clock

mode selection command.

- Multiplication ratio error

A multiplication ratio error occurs when the multiplication ratio specified through a new bit rate selection command does not match the clock mode selected through a clock mode selection command. To check the selectable multiplication ratios, issue a multiplication ratio inquiry command.

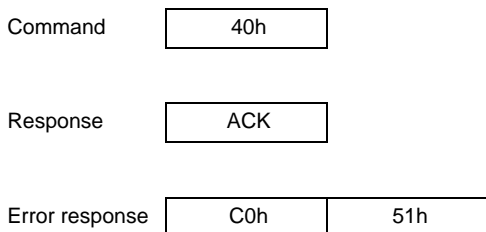
- Operating frequency error

An operating frequency error occurs when the RX62N/RX621 cannot operate at the operating frequencies selected through a new bit rate selection command. The RX62N/RX621 calculates the operating frequencies from the input frequency and multiplication ratios specified through a new bit rate selection command and checks if each calculated frequency is within the range from the minimum to maximum frequencies for the respective clock. To check the minimum and maximum operating frequencies for each clock, issue an operating clock frequency inquiry command.

(12) Programming/Erase State Transition

In response to a programming/erase state transition command sent from the host, the RX62N/RX621 determines whether ID code protection is enabled or disabled using the control code and ID code written in the ROM. When ID code protection is enabled, the RX62N/RX621 returns a response (16h) and waits for the ID code. When ID code protection is disabled, the RX62N/RX621 erases the entire area of each of the user mat, user boot mat, and data mat. After completing erasure, the RX62N/RX621 returns a response (26h) and waits for a programming/erase host command. If the RX62N/RX621 has failed to complete erasure due to an error, it returns an error response (sends C0h and 51h in that order).

Do not issue a programming/erase state transition command before device selection, clock mode selection, and new bit rate selection commands.



[Legend]

ACK (1 byte): ACK code
 26h: ID code protection is disabled
 16h: ID code protection is enabled

(13) Embedded Program Status Inquiry

In response to an embedded program status inquiry command sent from the host, the RX62N/RX621 returns its current status. The embedded program status inquiry command can be issued in both the inquiry/selection host command wait state and programming/erasure host command wait state.

Command

4Fh

Response

5Fh	Size	Status	Error
-----	------	--------	-------

[Legend]

Size (1 byte): Total number of bytes in the status and error fields (fixed at two)

Status (1 byte): Current status in the RX62N/RX621 (see Table 38.14)

Error (1 byte): Error status in the RX62N/RX621 (see Table 38.15)

Table 38.14 Status Code

Code	Description
11h	Waiting for device selection
12h	Waiting for clock mode selection
13h	Waiting for bit rate selection
1Fh	Waiting for transition to programming/erasure host command wait state (bit rate has been selected)
31h	Erasing the user mat and user boot mat
3Fh	Waiting for a programming/erasure host command
4Fh	Waiting for reception of programming data
5Fh	Waiting for erasure block selection

Table 38.15 Error Code

Code	Description
00h	No error
11h	Checksum error
21h	Incorrect device code error
22h	Incorrect clock mode error
24h	Bit rate selection error
25h	Input frequency error
26h	Multiplication ratio error
27h	Operating frequency error
29h	Block number error
2Ah	Address error
2Bh	Data size error
51h	Erasure error
52h	Incomplete erasure error
53h	Programming error
54h	Selection error
80h	Command error
FFh	Bit rate adjustment verification error

38.9.7 ID Code Wait State

Table 38.16 shows the host command available in the ID code wait state.

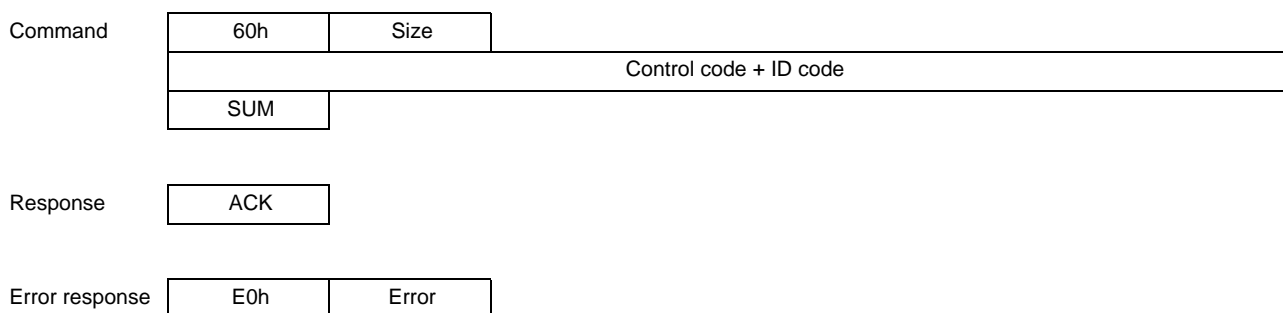
Table 38.16 ID Code Check Host Command

Host Command Name	Function
ID code check	Performs the ID code check

If the host has sent an undefined command, the RX62N/RX621 returns a response indicating a command error. For the format of this response, see section 38.9.6, Inquiry/Selection Host Command Wait State.

(1) ID Code Check

In response to an ID code check command sent from the host, the RX62N/RX621 compares the code sent from the host with the control code and ID code in the ROM and returns the result.



[Legend]

- Size (1 byte): Number of bytes in the ID code field (fixed at 16)
 - ID code (16 bytes): Control code (1 byte) + ID code (15 bytes)
 - SUM (1 byte): Checksum
 - ACK (1 byte): ACK code
 - Error (1 byte): Error code
 - 26h: Returns the response for a programming/erasure state transition command
 - 11h: Checksum error
 - 61h: ID code mismatch
 - 63h: ID code mismatch (erasure error)
- An error has occurred during erasure triggered by an ID code mismatch.

38.9.8 Programming/Erase Host Command Wait State

Table 38.17 shows the host commands available in the programming/erase host command wait state.

Table 38.17 Programming/Erase Host Commands

Host Command Name	Function
User boot mat programming selection	Selects the program for user boot mat programming
256-byte programming	Programs 256 bytes of data
Erase selection	Selects the erase program
Block erase	Erases block data
Memory read	Reads data from memory
User boot mat checksum	Performs checksum verification for the user boot mat
User mat checksum	Performs checksum verification for the user mat
User boot mat blank check	Checks whether the user boot mat is blank
User mat blank check	Checks whether the user mat is blank
Read lock bit status	Reads from the lock bit
Lock bit program	Writes to the lock bit
Lock bit enable	Enables the lock bit protection
Lock bit disable	Disables the lock bit protection
Embedded program status inquiry	Inquires regarding the state of the RX62N/RX621

If the host has sent an undefined command, the RX62N/RX621 returns a response indicating a command error. For the format of this response, see section 38.9.6, Inquiry/Selection Host Command Wait State.

To program the ROM, issue a programming selection command (user mat programming selection/user boot mat programming selection) and then a 256-byte programming command from the host. Upon reception of a programming selection command, the RX62N/RX621 enters the programming data wait state (see section 38.9.4, State Transitions in Boot Mode). In response to a 256-byte programming command sent from the host in this state, the RX62N/RX621 starts programming the ROM. When the host sends a 256-byte programming command specifying FFFF FFFFh as the programming start address, the RX62N/RX621 detects it as the end of programming and enters the programming/erase host command wait state.

To erase the ROM, issue an erase selection command and then a block erase command from the host. Upon reception of an erase selection command, the RX62N/RX621 enters the erase block selection wait state (see section 38.9.4, State Transitions in Boot Mode). In response to a block erase command sent from the host in this state, the RX62N/RX621 erases the specified block in the ROM. When the host sends a block erase command specifying FFh as the block number, the RX62N/RX621 detects it as the end of erase and enters the programming/erase host command wait state.

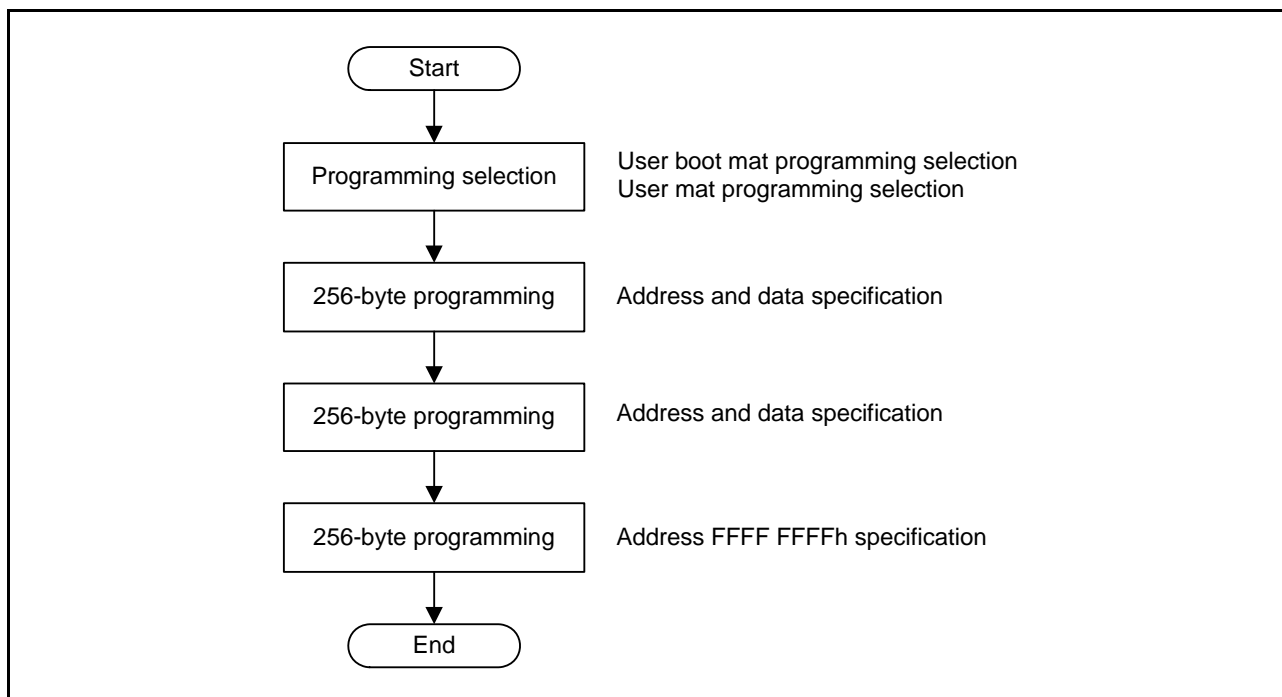


Figure 38.32 Procedure for ROM Programming in Boot Mode

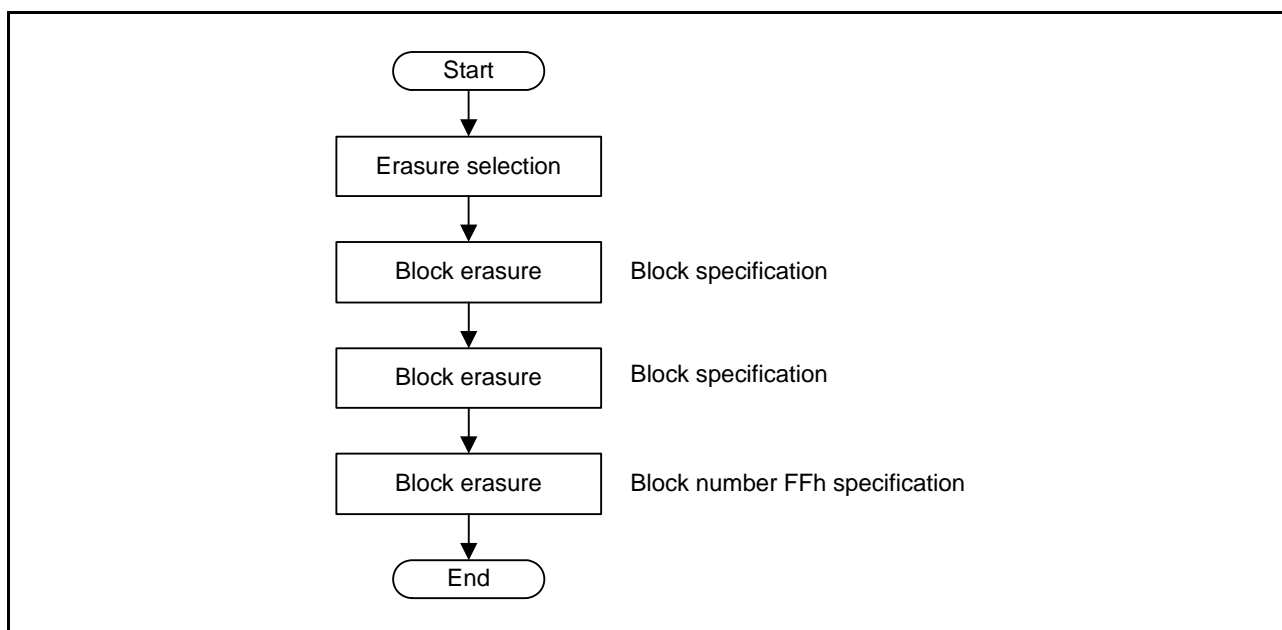


Figure 38.33 Procedure for ROM Erasure in Boot Mode

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to the RX62N/RX621 and the "response" indicates a response sent from the RX62N/RX621 to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by the RX62N/RX621 becomes 00h.

(1) User Boot Mat Programming Selection

In response to a user boot mat programming selection command sent from the host, the RX62N/RX621 selects the program for user boot mat programming and waits for programming data.

Command

42h

Response

06h

(2) User Mat Programming Selection

In response to a user mat programming selection command sent from the host, the RX62N/RX621 selects the program for user mat programming and waits for programming data.

Command

43h

Response

06h

(3) 256-Byte Programming

In response to a 256-byte programming command sent from the host, the RX62N/RX621 programs the ROM. After completing ROM programming successfully, the RX62N/RX621 returns a response (06h). If an error has occurred during ROM programming, the RX62N/RX621 returns an error response (D0h).

Command

50h	Programming address		
Data	Data	...	Data
SUM			

Response

06h

Error response

D0h	Error
-----	-------

[Legend]

Programming address (4 bytes): Target address of programming
 To program the ROM, a 256-byte boundary address should be specified.
 To terminate programming, FFFF FFFFh should be specified.

Data (256 bytes): Programming data
 FFh should be specified for the bytes that do not need to be programmed.
 When terminating programming, no data needs to be sent (only the programming address and SUM should be sent in that order).

SUM (1 byte): Checksum

Error (1 byte): Error code
 11h: Checksum error
 2Ah: Address error (the specified address is not in the target mat)
 53h: Programming cannot be done due to a programming error

(4) Erasure Selection

In response to an erasure selection command sent from the host, the RX62N/RX621 selects the erasure program and waits for erasure block specification.

Command	48h
---------	-----

Response	06h
----------	-----

(5) Block Erasure

In response to a block erasure command sent from the host, the RX62N/RX621 erases the ROM. After completing ROM erasure successfully, the RX62N/RX621 returns a response (06h). If an error has occurred during ROM erasure, the RX62N/RX621 returns an error response (D8h).

Command	58h	Size	Block	SUM
---------	-----	------	-------	-----

Response	06h
----------	-----

Error response	D8h	Error
----------------	-----	-------

[Legend]

Size (1 byte): Number of bytes in the block specification field (fixed at 1)

Block (1 byte): Block number whose data is to be erased
To terminate erasure, FFh should be specified.

SUM (1 byte): Checksum

Error (1 byte): Error code

11h: Checksum error

29h: Block number error (an incorrect block number is specified)

51h: Erasure cannot be done due to an erasure error

(6) Memory Read

In response to a memory read command sent from the host, the RX62N/RX621 reads data from the ROM. After completing ROM reading successfully, the RX62N/RX621 returns the data stored in the address specified by the memory read command. If the RX62N/RX621 has failed to read the ROM, the RX62N/RX621 returns an error response (D2h).

Command	52h	Size	Area	Read start address	
	Reading size			SUM	

Response	52h	Reading size			
	Data	Data	...	Data	
	SUM				

Error response	D2h	Error
----------------	-----	-------

[Legend]

- Size (1 byte): Total number of bytes in the area, read start address, and reading size fields
- Area (1 byte): Target mat to be read
 - 00h: User boot mat
 - 01h: User mat
- Read start address (4 bytes): Start address of the area to be read
- Reading size (4 bytes): Size of data to be read (bytes)
- SUM (1 byte): Checksum
- Data (1 byte): Data read from the ROM
- Error (1 byte): Error code
 - 11h: Checksum error
 - 2Ah: Address error
 - The value specified for area selection is neither 00h nor 01h.
 - The specified read start address is outside the selected mat.
 - 2Bh: Data size error
 - 00h is specified for the reading size.
 - The reading size is larger than the mat.
 - The end address calculated from the read start address and the reading size is outside the selected mat.

(7) User Boot Mat Checksum

In response to a user boot mat checksum command sent from the host, the RX62N/RX621 sums the user boot mat data in byte units and returns the result (checksum).

Command	4Ah
---------	-----

Response	5Ah	Size	Mat checksum	SUM
----------	-----	------	--------------	-----

[Legend]

- Size (1 byte): Number of bytes in the mat checksum field (fixed at 4)
- Mat checksum (4 bytes): Checksum of the user boot mat data
- SUM (1 byte): Checksum (for the response data)

(8) User Mat Checksum

In response to a user mat checksum command sent from the host, the RX62N/RX621 sums the user mat data in byte units and returns the result (checksum).

Command

4Bh

Response

5Bh	Size	Mat checksum	SUM
-----	------	--------------	-----

[Legend]

Size (1 byte): Number of bytes in the mat checksum field (fixed at 4)

Mat checksum (4 bytes): Checksum of the user mat data

The user mat also stores the key code for debugging function authentication. Note that the checksum includes this key code value.

SUM (1 byte): Checksum (for the response data)

(9) User Boot Mat Blank Check

In response to a user boot mat blank check command sent from the host, the RX62N/RX621 checks whether the user boot mat is completely erased. When the user boot mat is completely erased, the RX62N/RX621 returns a response (06h). If the user boot mat boot has an unerased area, the RX62N/RX621 returns an error response (sends CCh and 52h in that order).

Command

4Ch

Response

06h

Error response

CCh	52h
-----	-----

(10) User Mat Blank Check

In response to a user mat blank check command sent from the host, the RX62N/RX621 checks whether the user mat is completely erased. When the user mat is completely erased, the RX62N/RX621 returns a response (06h). If the user mat has an unerased area, the RX62N/RX621 returns an error response (sends CDh and 52h in that order).

Command

4Dh

Response

06h

Error response

CDh	52h
-----	-----

(11) Read Lock Bit Status

In response to a read lock bit status command sent from the host, the RX62N/RX621 reads data from the lock bit. After completing the lock bit reading successfully, the RX62N/RX621 returns the data stored in the address specified by the read lock bit status command. If the RX62N/RX621 has failed to read the lock bit, the RX62N/RX621 returns an error response (F1h).

Command	71h	Size	Area	Third highest order address	Second highest order address	Highest order address	SUM
---------	-----	------	------	-----------------------------	------------------------------	-----------------------	-----

Response	Status
----------	--------

Error response	F1h	Error
----------------	-----	-------

[Legend]

Size (1 byte): Total number of bytes in the area, third highest order address, second highest order address, and highest order address fields (fixed at 4 in the RX62N/RX621)

Area (1 byte): Target mat to be read
01h: User mat

Third highest order address (1 byte): Third highest order address at the specified block's end address (8 to 15 bits)

Second highest order address (1 byte): Second highest order address at the specified block's end address (16 to 23 bits)

Highest order address (1 byte): Highest order address at the specified block's end address (24 to 31 bits)

SUM (1 byte): Checksum

Status (1 byte): Bit 6 locked at "0"
Bit 6 unlocked at "1"

Error (1 byte): Error code
11h: Checksum error
2Ah: Address error (the specified address is not in the target mat)

(12) Lock Bit Program

In response to a lock bit program command sent from the host, the RX62N/RX621 writes to a lock bit and locks the specified block. After completing the lock bit blocking successfully, the RX62N/RX621 returns a response (06h). If the RX62N/RX621 has failed to lock, the RX62N/RX621 returns an error response (F7h).

Command	77h	Size	Area	Third highest order address	Second highest order address	Highest order address	SUM
---------	-----	------	------	-----------------------------	------------------------------	-----------------------	-----

Response	06h
----------	-----

Error response	F7h	Error
----------------	-----	-------

[Legend]

Size (1 byte):	Total number of bytes in the area, third highest order address, second highest order address, and highest order address fields (fixed at 4 in the RX62N/RX621)
Area (1 byte):	Target mat to be locked 01h: User mat
Third highest order address (1 byte):	Third highest order address at the specified block's end address (8 to 15 bits)
Second highest order address (1 byte):	Second highest order address at the specified block's end address (16 to 23 bits)
Highest order address (1 byte):	Highest order address at the specified block's end address (24 to 31 bits)
SUM (1 byte):	Checksum
Error (1 byte):	Error code 11h: Checksum error 2Ah: Address error (the specified address is not in the target mat) 53h: Locking cannot be done due to a programming error

(13) Lock Bit Enable

In response to a lock bit enable command sent from the host, the RX62N/RX621 enables a lock bit.

Command	7Ah
---------	-----

Response	06h
----------	-----

(14) Lock Bit Disable

In response to a lock bit disable command sent from the host, the RX62N/RX621 disables a lock bit.

Command	75h
---------	-----

Response	06h
----------	-----

(15) Embedded Program Status Inquiry

For details, refer to section 38.9.6, Inquiry/Selection Host Command Wait State.

38.10 USB (User) Boot Mode

When the USB (user) boot mode is specified through the MD1 and MD0 pins and a reset is canceled, the LSI is activated from the user boot mat. The reset vector here is FF7F FFFCh of the user boot mat address. Other vector tables refer to the normal vector table (see section 11, Interrupt Control Unit (ICUa)).

At shipment, the USB boot program is stored in the user boot mat. This program allows programming of user mat via the USB.

The user boot mat can be programmed in boot mode. Programming the USB boot program allows programming via an arbitrary interface.

The USB boot mode using the USB boot program is described below.

In USB boot mode, the user mat is programmed or erased by control commands and data for programming transmitted from an externally connected host via the USB.

Using USB boot mode requires preparation on the host side of tools for transmitting the control commands and data for programming, and of the data. Figure 38.34 shows the configuration of a system for use with USB boot mode. Interrupt requests generated in USB boot mode are ignored. Ensure that interrupt requests are not generated on the system side.

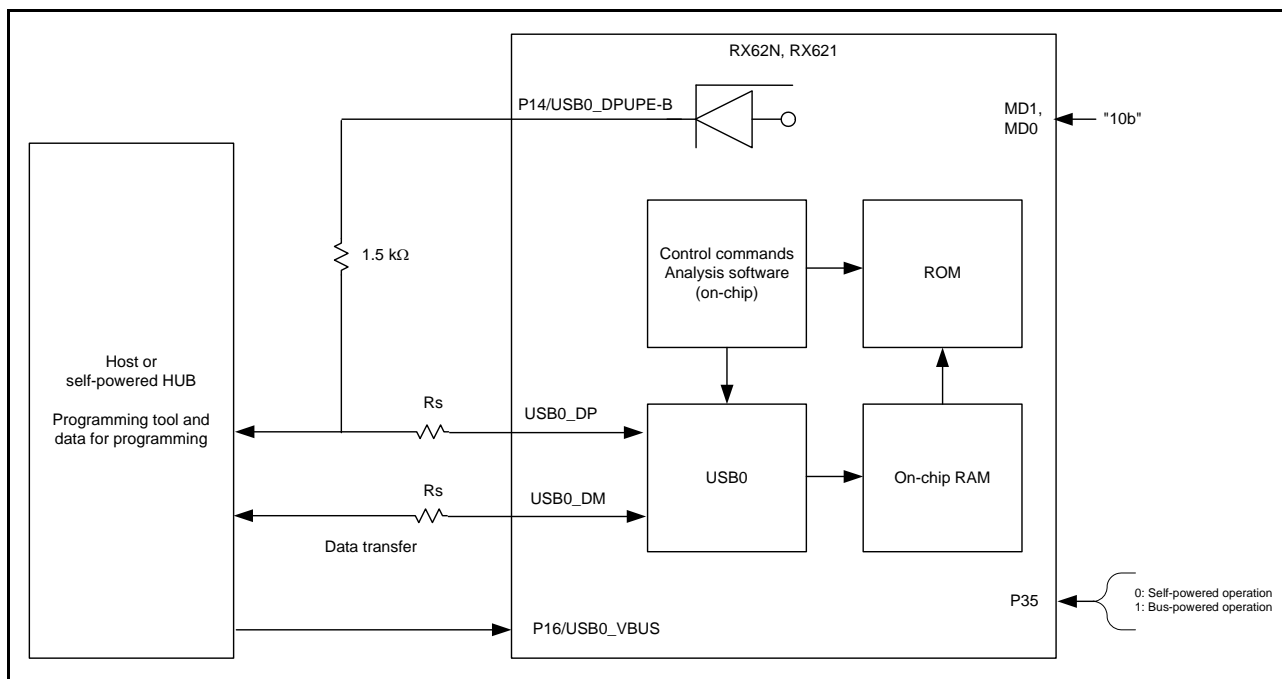


Figure 38.34 Configuration of a System for Use with USB Boot Mode

38.10.1 Features

- Selection of bus-powered or self-powered mode
- Only the USB0-DPUPE-B pin need be used for control of D+ pull-up connection
- See Table 38.18 for the enumeration information

Note that the USB boot mode is only available for USB0, and only when USB0 is operating in USB-function mode.

Table 38.18 Enumeration Information

USB specification	Ver.2.0 (Full-speed)	
Transfer modes	Transfer mode control (in, out) Bulk (in, out)	
Maximum current	Self-powered mode (Pin P35 = 0)	100 mA
	Bus-powered mod (Pin P35 = 1)	500 mA
End-point structure	<pre> EP0 Control (in out) 8 bytes Configuration1 ├── InterfaceNumber0 │ └── AlternateSetting0 │ ├── EP1 Bulk (out) 64 bytes │ └── EP2 Bulk (in) 64 bytes </pre>	

38.10.2 State Transitions

State transitions after a system is booted-up in USB boot mode are shown in Figure 38.35.

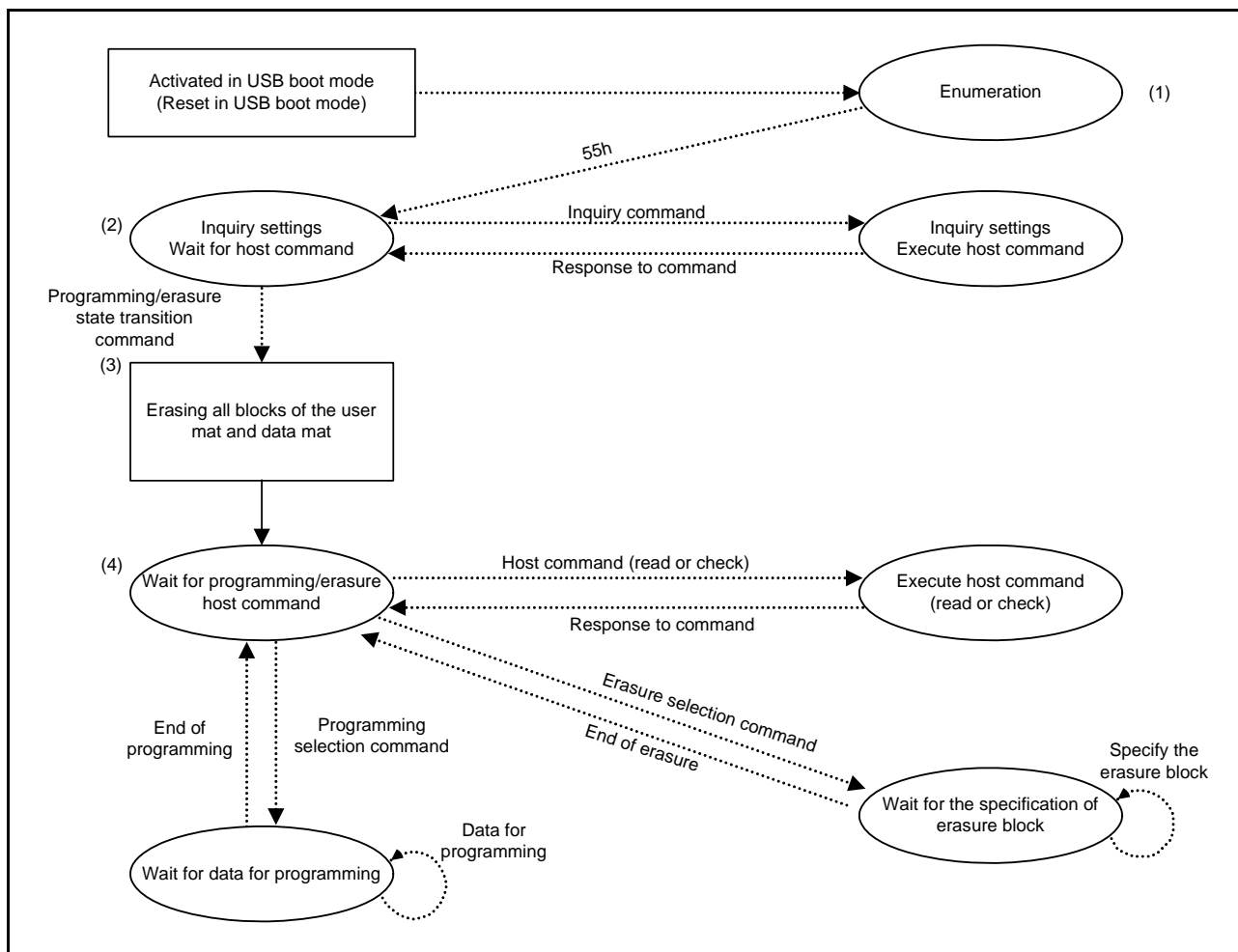


Figure 38.35 State Transitions in USB Boot Mode

1. On the transition to USB boot mode, the boot program embedded within this chip is initiated. When the boot program is initiated, the process of enumeration is conducted with the host. Once enumeration is completed, the host sends a single byte (55h) to this chip. If this byte is not received correctly, booting-up in USB boot mode is restarted.
2. Inquiry data on the size, structure, first address, state of support, etc. of the user mat are sent to the host.
3. On completion of the inquiries, all blocks of the user mat and data mat are automatically erased.
4. After automatic erasure of the user mat and data mat, the state shifts to waiting for programming and erasure commands. When a programming command is received, the state shifts to waiting for data to be programmed. When an erasure-selection command is received, the state shifts to waiting for specification of blocks to be erased. Other than the programming and erasure commands, commands for the following processes on the user mat are also possible: sum checking, blank checking (checking erasure), reading memory, and acquiring current status information.

38.10.3 Points to Note When Running Programs in USB Boot Mode

- A 48-MHz clock signal must be provided to the USB module. Make settings for the clock oscillator and frequency of the external clock to ensure that the dedicated clock for the USB (UCLK) is at 48 MHz. Confirm the details of this by consulting section 8, Clock Generation Circuit.
- Use the USB0-DPUPE-B pin for control of D+ pull-up connection.
- To ensure that the power supply is stable during programming and erasure of flash memory, do not connect a cable that is in turn connected via a bus-powered hub.
- In the worst case, disconnection of the USB cable during programming and erasure of flash memory can permanently damage the chip, so take particular care to ensure that the cable remains in place.
- Even if the USB enters the suspended mode while the bus is supplying it with power, the chip does not be placed in software standby mode of the low power-consumption states.

38.11 ID Code Protection on Connection of the On-Chip Debugger

This function is used to prohibit connection with the on-chip debugger. When connecting an on-chip debugger, the control code and ID code that have been written to the ROM are used to determine whether ID code protection on connection of the on-chip debugger is enabled or disabled and to judge ID code protection on connection of the on-chip debugger. When the ID code protection is enabled, the code sent from the on-chip debugger is compared with the control code and ID code in the ROM to determine whether they match. If they match, connection with the on-chip debugger is allowed. If they do not match, the on-chip debugger cannot be connected. However, if the control code is 52h and the ID code is 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh (from the ID code 1 field), there is no determination of matching and the ID code is always considered to be non-matching. Furthermore, if all bytes of the control code and ID code have the value FFh, there is no determination of matching, the ID code is always considered to match, and connection of the on-chip debugger is allowed. See Figure 38.25 for the configuration of ID codes in flash memory.

Table 38.19 Specifications for ID Code Protection on Connection of the On-Chip Debugger

Control Code	ID Code	State of Protection	Operations at the Time of Connection with the On-Chip Debugger
FFh	FFh, ..., FFh (all bytes FFh)	Protection disabled	The control code and ID code are not judged, the ID code always matches, and connection to the on-chip debugger is permitted.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh	Protection enabled	The control code and ID code are not judged, the ID code is always non-matching, and connection to the on-chip debugger is prohibited.
Other than the above	Other than the above	Protection enabled	Matching ID code: Authentication of the on-chip debugger is ended and connection with the on-chip debugger is permitted. Non-matching ID code: Further transition to the ID code protection waiting state

38.12 ROM Code Protection

ROM code protection is a facility for prohibiting a PROM programmer from reading from or writing to flash memory. The ROM code in flash memory is a 32-bit code. Figure 38.35 shows the configuration of ROM codes. Set ROM codes as 32-bit units.

For release from ROM code protection, erase block EB00 (eraser block 00) in boot mode or by user programming.

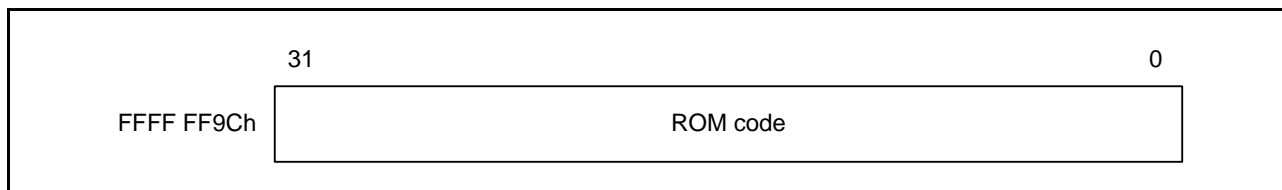


Figure 38.36 Configuration of a ROM Code

Table 38.20 Specifications for ROM Code Protection

ROM Code	State of Protection	Operations at the Time of Connection with the PROM Programmer
0000 0000h	Protection enabled (ROM code protection 1)	Access (both reading and writing) to the user mat and user boot mat is prohibited.
0000 0001h	Protection enabled (ROM code protection 2)	Reading from the user mat and user boot mat is prohibited.
Other than the above	Protection disabled	Access (both reading and writing) to the user mat and user boot mat is permitted.

38.13 Usage Notes

(1) Areas where Programming or Erasure is Suspended

Data in areas where programming or erasure is suspended are undefined. To avoid malfunctions due to the reading of undefined data, prevent the reading of data and execution of code from areas where programming or erasure is currently suspended.

(2) Suspending Programming or Erasure

If you use the programming/erasure suspension command to suspend the processing of programming or erasure, be sure to use the resumption command so that the processing is completed. Within 20 μ s (PCLK = 50 MHz) after the resume command has been issued, do not issue the programming/erasure suspension command again.

(3) Prohibition of Reprogramming

Two or more programming operations cannot be performed for the same address range. If an address range that has already been programmed is to be programmed again, be sure to erase the area in advance of the programming.

(4) Reset during Programming or Erasure

When applying a reset on the RES# pin during programming or erasure, release the chip from the reset state after at least 100 μ s have elapsed while the operating voltage is within the range specified in the electrical characteristics.

When using the FRESETR.FRESET bit to reset the FCU during programming or erasure, make sure that the reset state is maintained over tRESW2 (see section 41, Electrical Characteristics). Do not allow programming or erasure of the ROM while the FCU is being reset.

WDT and IWDT resets during programming or erasure can be used without securing the above intervals.

(5) Prohibition of Non-Maskable Interrupts during Programming or Erasure

The generation of a non-maskable interrupt (interrupt on the NMI pin, oscillation-stoppage detected interrupt, or a voltage-monitoring interrupt) during programming or erasure leads to fetching of the corresponding vector from the ROM, but the data read out are undefined. Therefore, make sure that non-maskable interrupts are not generated during programming or erasure. (This prohibition only applies to the ROM.)

(6) Interrupt Vector Assignment During Programming or Erasure

The generation of interrupts during programming or erasure may lead to the fetching of vectors from the ROM. To prevent access to the ROM area due to the generation of interrupts, set the interrupt table register (INTB) of the CPU so that the destination for the fetching of interrupt vector is an area outside the ROM.

(7) Abnormal Termination of Programming and Erasure

When programming or erasure is not completed normally due to a fluctuation of the operating voltage above the allowed range, a reset, a reset of the FCU by using the FRESETR.FRESET bit, entry to the command-locked state due to the detection of an error, or any of the prohibited items listed under point (8) below, the value of the lock bit becomes zero (corresponding to the protected state). In such cases, issue a block-erasure command to erase the lock bit while the FPROTR.FPROTCN bit is 1. After that, repeat the programming that was not completed normally.

(8) Actions Prohibited during Programming and Erasure

To prevent damage to the flash memory during programming or erasure, observe the prohibitions described below.

- In an RX62N or RX621 product, the voltage from the power supply being beyond the allowed range of operating voltage
- A change to the value of the FWEPROR.FLWE[1:0] bits
- A change to the operating mode due to the setting of the SYSCR0.ROME bit
- A change to the PCLK multiplication ratio due to the setting in the SCKCR register
- Setting the PCKAR register for a frequency other than that of PCLK
- A transition to all-module clock stop mode, software standby, or deep software standby

39. Data Flash Memory (Flash Memory for Data Storage)

The RX62N/RX621 Group has a maximum 512-Kbyte flash memory for storing program code (ROM) and a 32-Kbyte flash memory for storing data (data flash).

This section covers the flash memory for data storage. For the ROM, see section 38, ROM (Flash Memory for Code Storage).

39.1 Overview

Table 39.1 lists the specifications of the data flash memory, and Figure 39.1 is a block diagram of the ROM, data flash memory, and related modules.

Table 39.1 Specifications of Data Flash Memory

Item	Specifications	
Memory capacity	· Data mat: 32 Kbytes	
Reading via the peripheral bus	A read operation takes three cycles of PCLK in words or bytes	
Programming/erasing method	<ul style="list-style-type: none"> · The chip incorporates a dedicated sequencer (FCU) for programming of the data flash. · Programming and erasing the data flash are handled by issuing commands to the FCU. 	
BGO (background operation)	<ul style="list-style-type: none"> · Execution of program code from the ROM is possible while the data flash memory is being programmed or erased. · The CPU is able to execute program code from areas other than the ROM or data flash while the ROM is being programmed or erased. 	
Suspension and resumption	<ul style="list-style-type: none"> · The CPU is able to read the data flash area by suspending the programming/erasing operation to the data flash (suspended). · Programming and erasure of the ROM can be restarted (resumed) after suspension. 	
Units of programming and erasure	<ul style="list-style-type: none"> · Unit of programming for the data mat: 8 or 128 bytes · Unit of erasure for the data mat: 2 Kbytes (16 blocks) 	
Blank checking function	<ul style="list-style-type: none"> · The blank checking command can be executed to check the erasure state of data flash. · The size of the area to be blank-checked is 8 bytes or 2 Kbytes. 	
On-board programming (two types)	USB (user) boot mode	<ul style="list-style-type: none"> · Booting up from the user boot mat and programming of the user mat. · The USB boot program is stored in the user boot mat at shipment; the user mat is programmable via the USB. · Programming the user boot mat allows programming of the user mat via an arbitrary interface.
	USB boot mode	· The user mat is programmable via the USB.
	User program	Programming of the data mat under program control
Protection	Software-controlled protection	The FENTRYR.FENTRYD bit, FWEPROR.FLWE[1:0] bits, and DFLREk and DFLWEk registers (k = 0, 1), can be used to prevent unintentional programming.
	Error protection	Prevention of further programming or erasure after the detection of abnormal operations during programming or erasure
Times for programming and erasure, durability (number of times reprogramming is possible)	See section 41, Electrical Characteristics.	

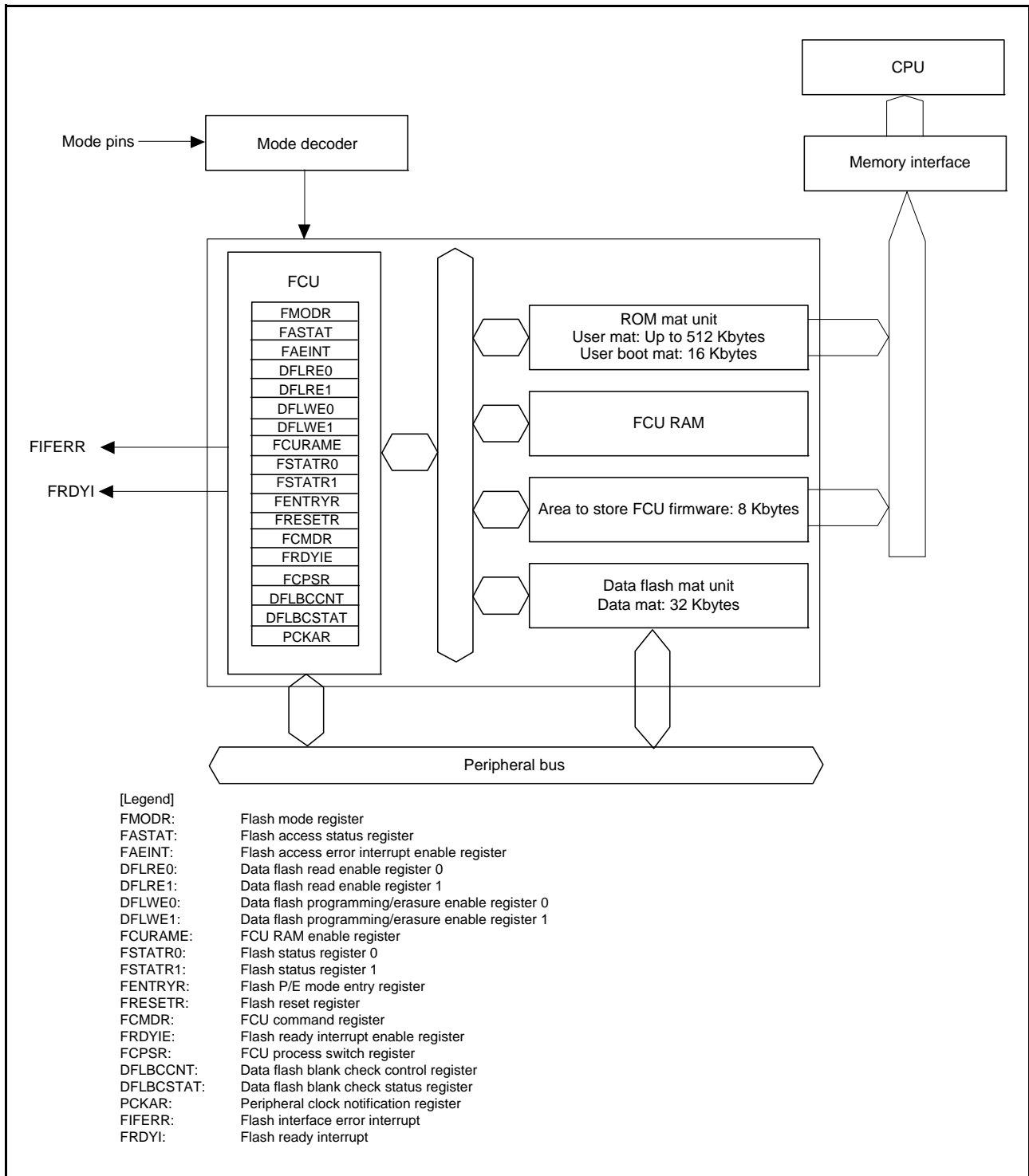


Figure 39.1 Block Diagram of Data Flash Memory

Input and output pins associated with the data flash are listed in Table 39.2.

Table 39.2 Input and Output Pins Associated with the Data Flash

Pin Name	I/O	Description
PF2/RxD1-B (176-pin version) P30/RxD1 (145-, 144-,100-, or 85-pin version)	Input	Used in boot mode to receive data via SCI (for host communications)
PF0/TxD1-B (176-pin version) P26/TxD1 (145-, 144-,100-, or 85-pin version)	Output	Used in boot mode to transmit data from SCI (for host communications)
MD1, MD0	Input	Operating mode settings for products of the RX62N/RX621 Groups
USB0_DP, USB0_DM	I/O	Input/output pins for USB data (for use in USB boot mode)
P16/USB0_VBUS	Input	Detection of connection and disconnection of USB cables (for use in USB boot mode)
P35	Input	Selection of USB bus-power mode or self-power mode (for use in USB boot mode)

39.2 Register Descriptions

Table 39.3 lists the registers related to the data flash memory. Some registers also have bits related to the ROM, but this section deals only with the bits that are relevant to the data flash. For registers containing bits with common functions for the ROM and data flash (FRDYIE, FCURAME, FSTATR0, FSTATR1, FRESETR, FCMDR, FCPSR, PCKAR, and FWEPROR) and details on the functions of bits dedicated to the ROM, see section 38.2, Register Descriptions.

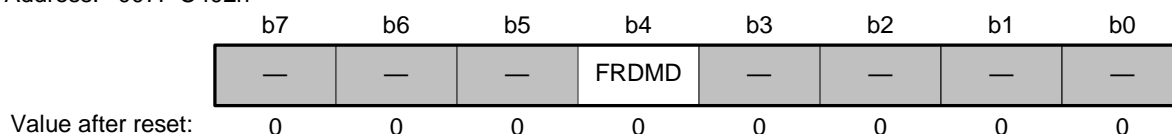
The registers related to the data flash are initialized by a reset.

Table 39.3 Registers Related to the Data Flash

Register Name	Symbol	Value after Reset	Address	Access Size
Flash mode register	FMODR	00h	007F C402h	8
Flash access status register	FASTAT	00h	007F C410h	8
Flash access error interrupt enable register	FAEINT	9Bh	007F C411h	8
Flash ready interrupt enable register	FRDYIE	00h	007F C412h	8
Data flash read enable register 0	DFLRE0	0000h	007F C440h	16
Data flash read enable register 1	DFLRE1	0000h	007F C442h	16
Data flash programming/erasure enable register 0	DFLWE0	0000h	007F C450h	16
Data flash programming/erasure enable register 1	DFLWE1	0000h	007F C452h	16
FCU RAM enable register	FCURAME	0000h	007F C454h	16
Flash status register 0	FSTATR0	80h	007F FFB0h	8
Flash status register 1	FSTATR1	0xh	007F FFB1h	8
Flash P/E mode entry register	FENTRYR	0000h	007F FFB2h	16
Flash reset register	FRESETR	0000h	007F FFB6h	16
FCU command register	FCMDR	FFFFh	007F FFBAh	16
FCU processing switching register	FCPSR	0000h	007F FFC8h	16
Data flash blank check control register	DFLBCCNT	0000h	007F FFCAh	16
Data flash blank check status register	DFLBCSTAT	0000h	007F FFCEh	16
Peripheral clock notification register	PCKAR	0000h	007F FFE8h	16
Flash write erase protection register	FWEPROR	02h	0008 C289h	8

39.2.1 Flash Mode Register (FMODR)

Address: 007F C402h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b4	FRDMD	FCU Read Mode Select	0:Memory Area Reading Method The method of lock bit reading is set. Since there are no lock bits for the data flash, undefined data are read from the data flash area after the FCU has been placed in lock bit read mode. 1:Register Reading Method This is the setting when the blank checking command is to be used.	R/W
b7 to b5	—	Reserved	These bits are always read as 0.The write value should always be 0.	R/W

FMODR is used to specify the method for the reading of lock bits. Set the FRDMD bit to 1 if blank checking is to be used.

In modes in which the on-chip ROM is disabled, the value read from FMODR is 00h and writing is disabled.

FMODR is initialized by a reset.

FRDMD Bit (FCU Read Mode Select)

Processing for a transition of the lock bit reading mode is used in processing for blank checking of the data flash.

The FRDMD bit is used to select the method of reading when lock bit values for the ROM are read out (see section 38, ROM (Flash Memory for Code Storage)).

39.2.2 Flash Access Status Register (FASTAT)

Address: 007F C410h

b7	b6	b5	b4	b3	b2	b1	b0
ROMAE	—	—	CMDLK	DFLAE	—	DFLRPE	DFLWPE

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

Bit	Symbol	Bit Name	Description	R/W
b0	DFLWPE	Data Flash Programming/Erase Protection Violation	0: No data flash programming/erasure command is not issued which conflicts with the DFLWEk settings 1: A data flash programming/erasure command is issued which conflicts with the DFLWEk settings (k = 0, 1)	R/(W)*
b1	DFLRPE	Data Flash Read Protection Violation	0: There is no such data flash read that conflicts with the DFLREk settings 1: There is such a data flash read that conflicts with the DFLREk settings (k = 0, 1)	R/(W)*
b2	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b3	DFLAE	Data Flash Access Violation	0: No data flash access violation 1: Data flash access violation	R/(W)*
b4	CMDLK	FCU Command Lock	0: FCU is not in the command-locked state 1: FCU is in the command-locked state	R
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	ROMAE	ROM Access Violation	See section 38, ROM (Flash Memory for Code Storage)	R/(W)*

Note : * Only 0 can be written after reading 1 to clear the flag.

FASTAT is a register to check if the access to the ROM/data flash is allowed.

When on-chip ROM is disabled, the data read from FASTAT is 00h and writing is disabled. When one of the bits in FASTAT is set to 1, the FCU is placed in the command-locked state (see section 39.7.2, Error Protection). To clear the command-locked state, a status clear command must be issued to the FCU after setting FASTAT to 10h.

FASTAT is initialized by a reset.

DFLWPE Bit (Data Flash Programming/Erase Protection Violation)

This bit is used to indicate whether or not the programming/erasure protection set by DFLWEy (y = 0, 1) is violated.

[Setting condition]

- A programming/erasure command is issued for a data flash area for which programming or erasure is disabled by DFLWEy.

[Clearing condition]

- When 0 is written after reading 1

DFLRPE Bit (Data Flash Read Protection Violation)

This bit is used to indicate whether or not the reading protection set by DFLREy (y = 0, 1) is violated.

[Setting condition]

- A read command is issued for a data flash area for which reading is disabled by DFLREy.

[Clearing condition]

- When 0 is written after reading 1

DFLAE Bit (Data Flash Read Protection Violation)

This bit indicates whether a data flash access violation occurred.

When the DFLAE bit is set to 1, the ILGLERR bit in FSTATR0 is set to 1, placing the FCU in the command-locked state.

For FSTATR0, see section 38.2.5, Flash Status Register 0 (FSTATR0).

[Setting conditions]

- A read command is issued for a data flash area in data flash P/E normal mode and when the FENTRYD bit in FENTRYR is set to 1.
- A write command is issued for a data flash area when the FENTRYD bit is set to 0.
- A command is issued for a data flash area when the FENTRYR.FENTRY0 bit is set to 1.

[Clearing condition]

- When 0 is written after reading 1

CMDLK Bit (FCU Command-Locked)

This command indicates that the FCU is in the command-locked state (see section 39.7.2, Error Protection).

[Setting condition]

- After the FCU detects an error and enters the command-locked state

[Clearing condition]

- After the FCU has processed a status clear command

39.2.3 Flash Access Error Interrupt Enable Register (FAEINT)

Address: 007F C411h

	b7	b6	b5	b4	b3	b2	b1	b0
	ROMAEIE	—	—	CMDLKIE	DFLAEIE	—	DFLRPEIE	DFLWPEIE
Value after reset:	1	0	0	1	1	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	DFLWPEIE	Data Flash Programming/ Erasure Protection Violation Interrupt Enable	0: FIFERR interrupt requests disabled when the DFLWPE bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the DFLWPE bit in FASTAT is set to 1	R/W
b1	DFLRPEIE	Data Flash Read Protection Violation Interrupt Enable	0: FIFERR interrupt requests disabled when the DFLRPE bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the DFLRPE bit in FASTAT is set to 1	R/W
b2	—	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
b3	DFLAEIE	Data Flash Read Access Violation Interrupt Enable	0: FIFERR interrupt requests disabled when the DFLAE bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the DFLAE bit in FASTAT is set to 1	R/W
b4	CMDLKIE	FCU Command Lock Interrupt Enable	0: FIFERR interrupt requests disabled when the CMDLK bit in FASTAT is set to 1 1: FIFERR interrupt requests enabled when the CMDLK bit in FASTAT is set to 1	R/W
b6, b5	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	ROMAEIE	ROM Access Violation Interrupt Enable	See section 38, ROM (Flash Memory for Code Storage).	R/W

FAEINT is a register to enable and disable a flash interface error interrupt (FIFERR).

When on-chip ROM is disabled, the data read from FAEINT is 00h and writing is disabled.

FAEINT is initialized by a reset.

DFLWPEIE Bit (Data Flash Programming/Erasure Protection Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when a data flash programming/erasure protection violation occurs and the DFLWPE bit in FASTAT is set to 1.

DFLRPEIE Bit (Data Flash Read Protection Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when a data flash read protection violation occurs and the DFLRPE bit in FASTAT is set to 1.

DFLAEIE Bit (Data Flash Read Protection Violation Interrupt Enable)

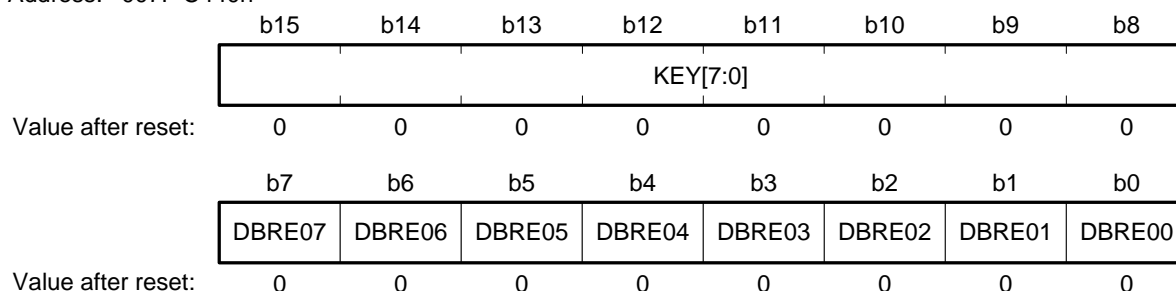
This bit is used to enable or disable FIFERR interrupt requests when a data flash access violation occurs and the DFLAE bit in FASTAT is set to 1.

CMDLKIE Bit (FCU Command Lock Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when a FCU command-locked state occurs and the CMDLK bit in FASTAT is set to 1.

39.2.4 Data Flash Read Enable Register 0 (DFLRE0)

Address: 007F C440h



Bit	Symbol	Bit Name	Description	R/W
b0	DBRE00	DB00 Block Read Enable	0: Read disabled 1: Read enabled	R/W
b1	DBRE01	DB01 Block Read Enable		R/W
b2	DBRE02	DB02 Block Read Enable		R/W
b3	DBRE03	DB03 Block Read Enable		R/W
b4	DBRE04	DB04 Block Read Enable		R/W
b5	DBRE05	DB05 Block Read Enable		R/W
b6	DBRE06	DB06 Block Read Enable		R/W
b7	DBRE07	DB07 Block Read Enable		R/W
b15 to b8	KEY[7:0]	Key Code	Enable or disable rewriting of the DBREi bit (i = 07 to 00).	R/(W)*

Note : * Write data is not retained.

DFLRE0 is a register to enable or disable the DB07 to DB00 blocks of the data mat (see Figure 39.3) to be read.

Only specific values written to the upper byte in word access are valid. Data written to the upper byte is not retained.

When on-chip ROM is disabled, the data read from DFLRE0 is 0000h and writing is disabled.

DFLRE0 is initialized by a reset.

DBREi Bit (DBi Block Read Enable) (i = 07 to 00)

This bit is used to enable or disable the DB07 to DB00 blocks of the data mat to be read.

The DBREi bit is used to control reading of the DBi blocks.

Writing to the DBREi is enabled only in word access when KEY[7:0] is 2Dh.

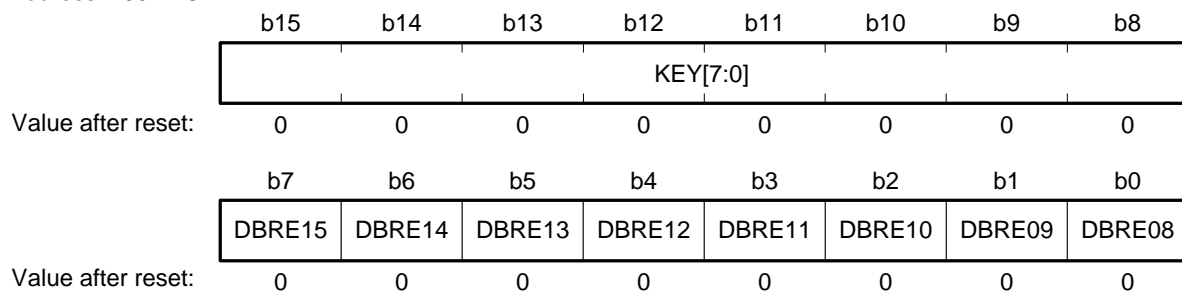
KEY[7:0] Bits (Key Code)

These bits are used to enable or disable rewriting of the DBREi bit.

Data written to the KEY[7:0] bits is not retained.

39.2.5 Data Flash Read Enable Register 1 (DFLRE1)

Address: 007F C442h



Bit	Symbol	Bit Name	Description	R/W
b0	DBRE08	DB08 Block Read Enable	0: Read disabled	R/W
b1	DBRE09	DB09 Block Read Enable	1: Read enabled	R/W
b2	DBRE10	DB10 Block Read Enable		R/W
b3	DBRE11	DB11 Block Read Enable		R/W
b4	DBRE12	DB12 Block Read Enable		R/W
b5	DBRE13	DB13 Block Read Enable		R/W
b6	DBRE14	DB14 Block Read Enable		R/W
b7	DBRE15	DB15 Block Read Enable		R/W
b15 to b8	KEY[7:0]	Key Code	Enable or disable rewriting of the DBREi bit (i = 15 to 08).	R/(W)*

Note : * Write data is not retained.

DFLRE1 is a register to enable or disable the DB15 to DB08 blocks of the data mat (see Figure 39.3) to be read.

Only specific values written to the upper byte in word access are valid. Data written to the upper byte is not retained.

When on-chip ROM is disabled, the data read from DFLRE1 is 0000h and writing is disabled.

DFLRE1 is initialized by a reset.

DBREi Bit (DBi Block Read Enable) (i = 15 to 08)

This bit is used to enable or disable the DB15 to DB08 blocks of the data mat to be read.

The DBREi bit is used to control reading of the DBi blocks.

Writing to the DBREi is enabled only in word access when KEY[7:0] is D2h.

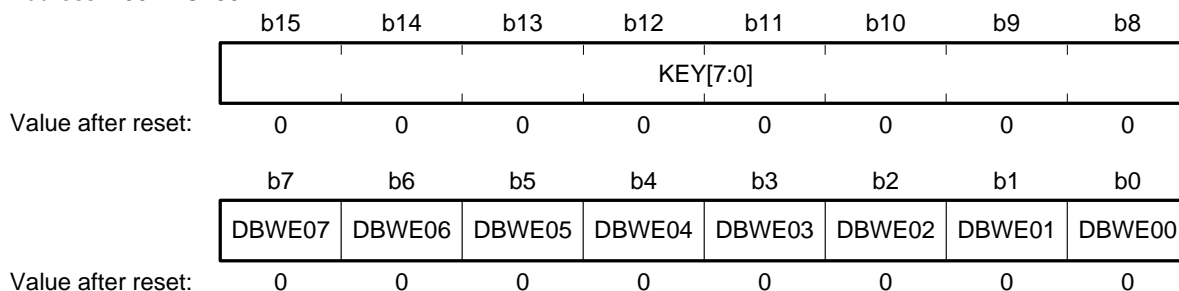
KEY[7:0] Bits (Key Code)

These bits are used to enable or disable rewriting of the DBREi bit.

Data written to the KEY[7:0] bits is not retained.

39.2.6 Data Flash Programming/Erase Enable Register 0 (DFLWE0)

Address: 007F C450h



Bit	Symbol	Bit Name	Description	R/W
b0	DBWE00	DB00 Block Programming/Erase Enable	0: Programming/erase disabled	R/W
b1	DBWE01	DB01 Block Programming/Erase Enable	1: Programming/erase enabled	R/W
b2	DBWE02	DB02 Block Programming/Erase Enable		R/W
b3	DBWE03	DB03 Block Programming/Erase Enable		R/W
b4	DBWE04	DB04 Block Programming/Erase Enable		R/W
b5	DBWE05	DB05 Block Programming/Erase Enable		R/W
b6	DBWE06	DB06 Block Programming/Erase Enable		R/W
b7	DBWE07	DB07 Block Programming/Erase Enable		R/W
b15 to b8	KEY[7:0]	Key Code	Enable or disable rewriting of the DBWEi bit (i = 07 to 00).	R/(W)*

Note : * Write data is not retained.

DFLWE0 is a register to enable or disable the DB07 to DB00 blocks of the data mat (see Figure 39.3) to be programmed or erased.

Only specific values written to the upper byte in word access are valid. Data written to the upper byte is not retained.

When on-chip ROM is disabled, the data read from DFLWE0 is 0000h and writing is disabled.

DFLWE0 is initialized by a reset.

DBWEi Bit (DBi Block Programming/Erase Enable) (i = 07 to 00)

This bit is used to enable or disable the DB07 to DB00 blocks of the data mat to be programmed or erased.

The DBWEi bit is used to control programming/erasure of the DBi blocks.

Programming of the DBWEi bit is enabled only in word access when KEY[7:0] is 1Eh.

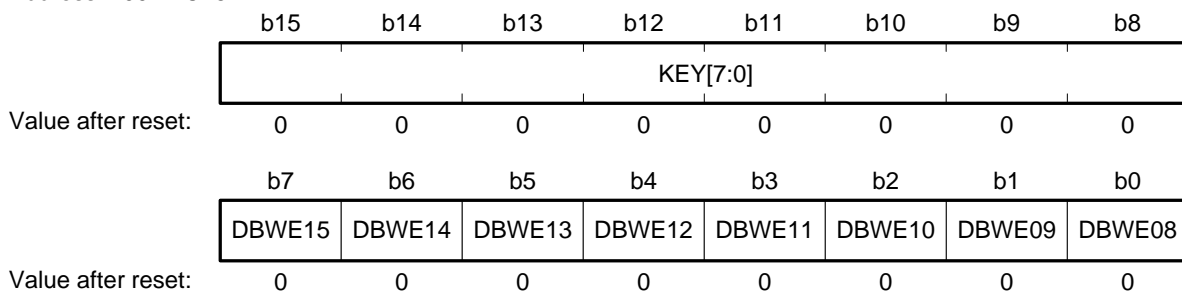
KEY[7:0] Bits (Key Code)

These bits are used to enable or disable rewriting of the DBWEi bit.

Data written to the KEY[7:0] bits is not retained.

39.2.7 Data Flash Programming/Erase Enable Register 1 (DFLWE1)

Address: 007F C452h



Bit	Symbol	Bit Name	Description	R/W
b0	DBWE08	DB08 Block Programming/Erase Enable	0: Programming/erase disabled	R/W
b1	DBWE09	DB09 Block Programming/Erase Enable	1: Programming/erase enabled	R/W
b2	DBWE10	DB10 Block Programming/Erase Enable		R/W
b3	DBWE11	DB11 Block Programming/Erase Enable		R/W
b4	DBWE12	DB12 Block Programming/Erase Enable		R/W
b5	DBWE13	DB13 Block Programming/Erase Enable		R/W
b6	DBWE14	DB14 Block Programming/Erase Enable		R/W
b7	DBWE15	DB15 Block Programming/Erase Enable		R/W
b15 to b8	KEY[7:0]	Key Code	Enable or disable rewriting of the DBWE _i bit (i = 15 to 08).	R/(W)*

Note : * Write data is not retained.

DFLWE1 is a register to enable or disable the DB15 to DB08 blocks of the data mat (see Figure 39.3) to be programmed or erased.

Only specific values written to the upper byte in word access are valid. Data written to the upper byte is not retained.

When on-chip ROM is disabled, the data read from DFLWE1 is 0000h and writing is disabled.

DFLWE1 is initialized by a reset.

DBWE_i Bit (DB_i Block Programming/Erase Enable) (i = 15 to 08)

This bit is used to enable or disable the DB15 to DB08 blocks of the data mat to be programmed or erased.

The DBWE_i bit is used to control programming/erasure of the DB_i blocks.

Programming of the DBWE_i bit is enabled only in word access when KEY[7:0] is E1h.

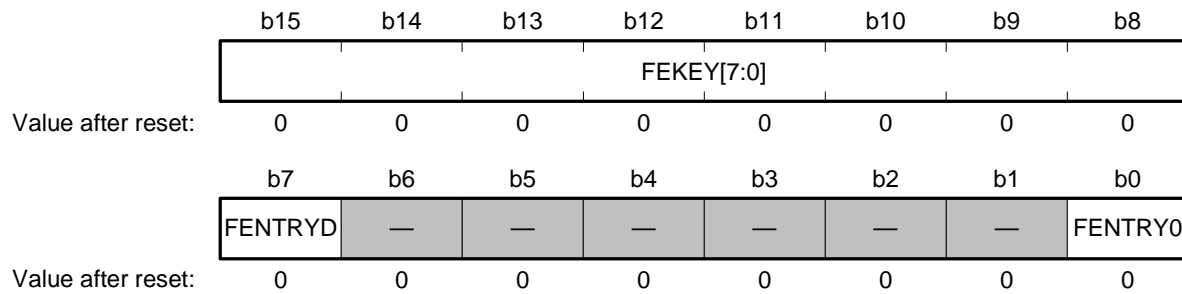
KEY[7:0] Bits (Key Code)

These bits are used to enable or disable rewriting of the DBWE_i bit.

Data written to the KEY[7:0] bits is not retained.

39.2.8 Flash P/E Mode Entry Register (FENTRYR)

Address: 007F FFB2h



Bit	Symbol	Bit Name	Description	R/W
b0	FENTRY0	ROM P/E Mode Entry 0	See section 38, ROM (Flash Memory for Code Storage).	R/W
b6 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b7	FENTRYD	Data Flash P/E Mode Entry	0: Data flash is in read mode 1: Data flash is in P/E mode	R/W
b15 to b8	FEKEY[7:0]	Key Code	Enable or disable rewriting of the FENTRYD and FENTRY0 bits.	R/(W)*

Note : * Write data is not retained.

FENTRYR is a register to place the ROM/data flash in P/E mode.

To place ROM/data flash in P/E mode and accept commands from the FCU, either the FENTRYD or FENTRY0 bit must be set to 1. If more than one bit is set to 1, the ILGLERR bit is set in FSTATR0 and the FCU enters the command-locked state.

Only specific values written to the upper byte in word access are valid. Any other writing causes the register to be initialized. Data written to the upper byte is not retained.

When on-chip ROM is disabled, the data read from FENTRYR is 0000h and writing is disabled.

FENTRYR is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

For FSTATR0, see section 38.2.5, Flash Status Register 0 (FSTATR0).

For FRESETR, see section 38.2.10, Flash Reset Register (FRESETR).

FENTRYD Bit (Data Flash P/E Mode Entry)

The FENTRYD bit is used to place the data flash in P/E mode.

[Writing-enable conditions (when all of the following conditions are met)]

- On-chip ROM is enabled.
- The FRDY bit in FSTATR0 is set to 1
- AAh is written to the FEKEY bit in word access.

[Setting condition]

- When the writing-enable conditions are met, FENTRYR is set to 0000h, and 1 is written to the FENTRYD bit

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FEKEY[7:0] bit is other than AAh.
- When the writing-enable conditions are met, 0 is written to the FENTRYD bit.
- When the writing-enable conditions are met and FENTRYR is other than 0000h, data is written to FENTRYR.

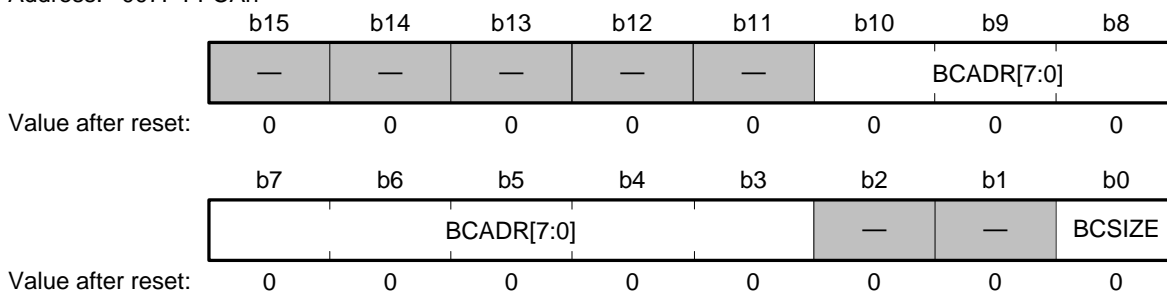
FEKEY[7:0] Bits (Key Code)

These bits enable or disable rewriting of the FENTRYD and FENTRY0 bits.

Data written to the FEKEY[7:0] bits is not retained.

39.2.9 Data Flash Blank Check Control Register (DFLBCCNT)

Address: 007F FFCAh



Bit	Symbol	Bit Name	Description	R/W
b0	BCSIZE	Blank Check Size Setting	0: The size of the area to be blank-checked is 8 bytes. 1: The size of the area to be blank-checked is 2 Kbytes.	R/W
b2, b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
b10 to b3	BCADR[7:0]	Blank Check Address Setting	Set the address of the area to be checked	R/W
b15 to b11	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

DFLBCCNT is a register for specifying the address and size of the area to be checked by a blank check command.

When on-chip ROM is disabled, the data read from DFLBCCNT is 0000h and writing is disabled.

DFLBCCNT is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

For FRESETR, see section 38.2.10, Flash Reset Register (FRESETR).

BCSIZE Bit (Blank Check Size Setting)

The BCSIZE bit is used to set the size of the area to be checked by a blank check command.

BCADR[7:0] Bits (Blank Check Address Setting)

When the size of the area to be checked by a blank check command is 8 bytes (the BCSIZE bit is set to 0), this bit is used to set the address of the area to be checked.

When the BCSIZE bit is set to 0, the setting of DFLBCCNT (the setting of the BCADR[7:0] bits shifted three bits in the MSB direction) added with the erased block start address specified when issuing a blank check command is the start address of the area to be checked.

39.2.10 Data Flash Blank Check Status Register (DFLBCSTAT)

Address: 007F FFCEh

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

Bit	Symbol	Bit Name	Description	R/W
b0	BCST	Blank Check Status	0: The area to be blank-checked is erased (blank) 1: 0 or 1 is written in the area to be blank-checked	R
b15 to b1	—	Reserved	These bits are always read as 0. The write value should always be 0.	R/W

DFLBCSTAT is a register which stores the results of a blank check command.

When on-chip ROM is disabled, the data read from DFLBCSTAT is 0000h and writing is disabled.

DFLBCSTAT is initialized by a reset, or when the FRESET bit in FRESETR is set to 1.

For FRESETR, see section 38.2.10, Flash Reset Register (FRESETR).

BCST Bit (Blank Check Status)

This bit is used to indicate the results of blank checking.

39.3 Configuration of Memory Mat for the Data Flash Memory

The data flash memory of products in the RX62N/RX621 Group is configured as a 32-Kbyte data mat. The address range occupied by this mat is shown in Figure 39.2.

Note that for the data mat, the address range for reading is the same as the address range for programming and erasure.

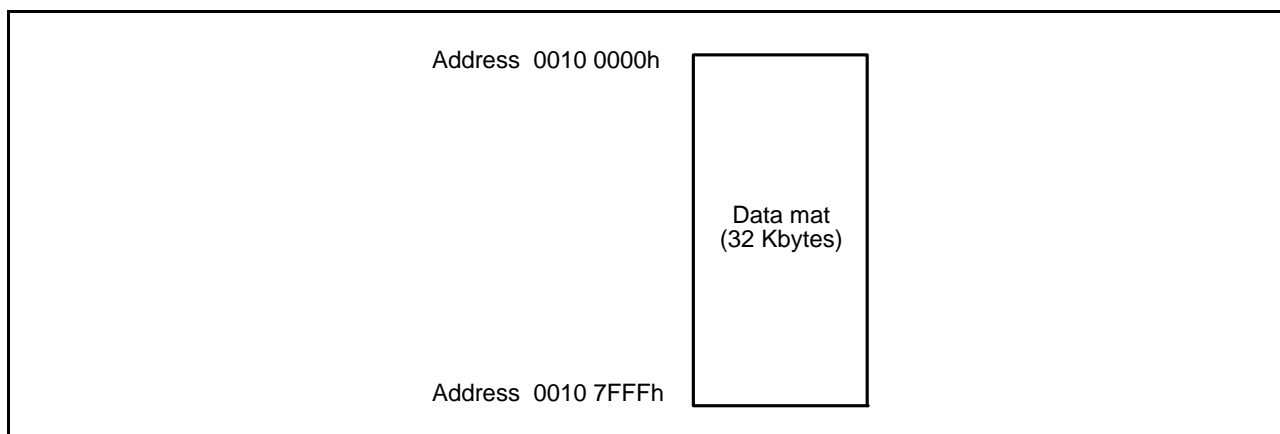


Figure 39.2 Configuration of the Data Mat

39.4 Block Configuration

Figure 39.3 shows how the erasure blocks of the data mat are configured. The data mat is divided into 16 2-Kbyte blocks, and erasure proceeds in these block units. Programming proceeds in 8- or 128-byte units. For programming in 8-byte units, each unit starts at an address where the value of the three lower-order bits is 0. For programming in 128-byte units, each unit starts at an address where the value of the lower-order bits is 00h or 80h.

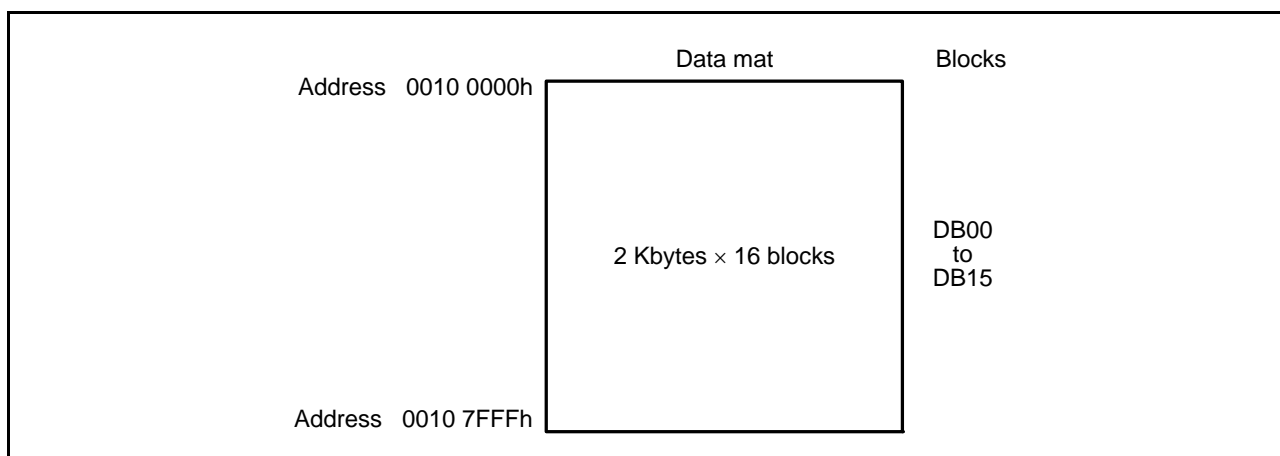


Figure 39.3 Division of the Data Mat into Blocks

39.5 Operating Modes Associated with the Data Flash

For the transitions between operating modes, see section 38.5, Operating Modes Associated with the ROM.

Reading, programming, and erasing of the data flash memory in an on-board device can proceed if the device is in boot mode, USB (user) boot mode, or single-chip mode (with on-chip ROM enabled), or in on-chip-ROM-enabled expansion mode.

The differences between modes are indicated in Table 39.4.

Table 39.4 Differences between Modes

Item	Boot Mode	USB (User) Boot Mode	Single-Chip Mode (with On-chip ROM Enabled) or On-chip-ROM-Enabled Expansion Mode
Environment for programming and erasure	On-board programming	On-board programming	On-board programming
Programmable and erasable mat	Data mat	Data mat	Data mat
Division into erasure blocks	Possible*1	Possible*1	Possible
Target mat for booting after a reset	Mat containing the embedded program*2	User boot mat	User mat

Note 1. All flash memory areas may be erased at the time of booting up. Specified blocks can subsequently be erased. For details, refer to section 38.9.2, ID Code Protection, section 38.9.4, State Transitions in Boot Mode, and section 38.10.2, State Transitions

Note 2. Not available to users.

- In boot mode, a host is able to program or read out the data mat via an SCI.
- USB (user) boot mode is activated from the user boot mat. At shipment, the USB boot program is stored in the user boot mat, allowing programming or reading the user mat or data mat via the USB. Programming the user boot mat in boot mode allows programming or reading the user mat or data mat via an arbitrary interface.
- In boot mode, on-chip RAM is employed for the embedded program for use in boot mode. For this reason, preserving the contents of on-chip RAM is not possible in this case.

39.6 Programming and Erasing the Data Flash Memory

The data flash memory is programmed and erased by issuing commands to a dedicated sequencer (FCU) for programming and erasure. The FCU has five modes. For programming and erasure, the mode is changed and then commands for programming and erasure are issued.

The mode transitions required to program or erase the data flash and the system of commands are described below. The descriptions apply in common to boot mode, USB (user) boot mode and single-chip mode (with on-chip ROM enabled), and to on-chip-ROM-enabled expansion mode.

39.6.1 FCU Modes

The FCU has five modes or sets of modes. Transitions between modes are caused by writing to the FENTRYR register or issuing FCU commands. Figure 39.4 is a diagram of the FCU mode transitions.

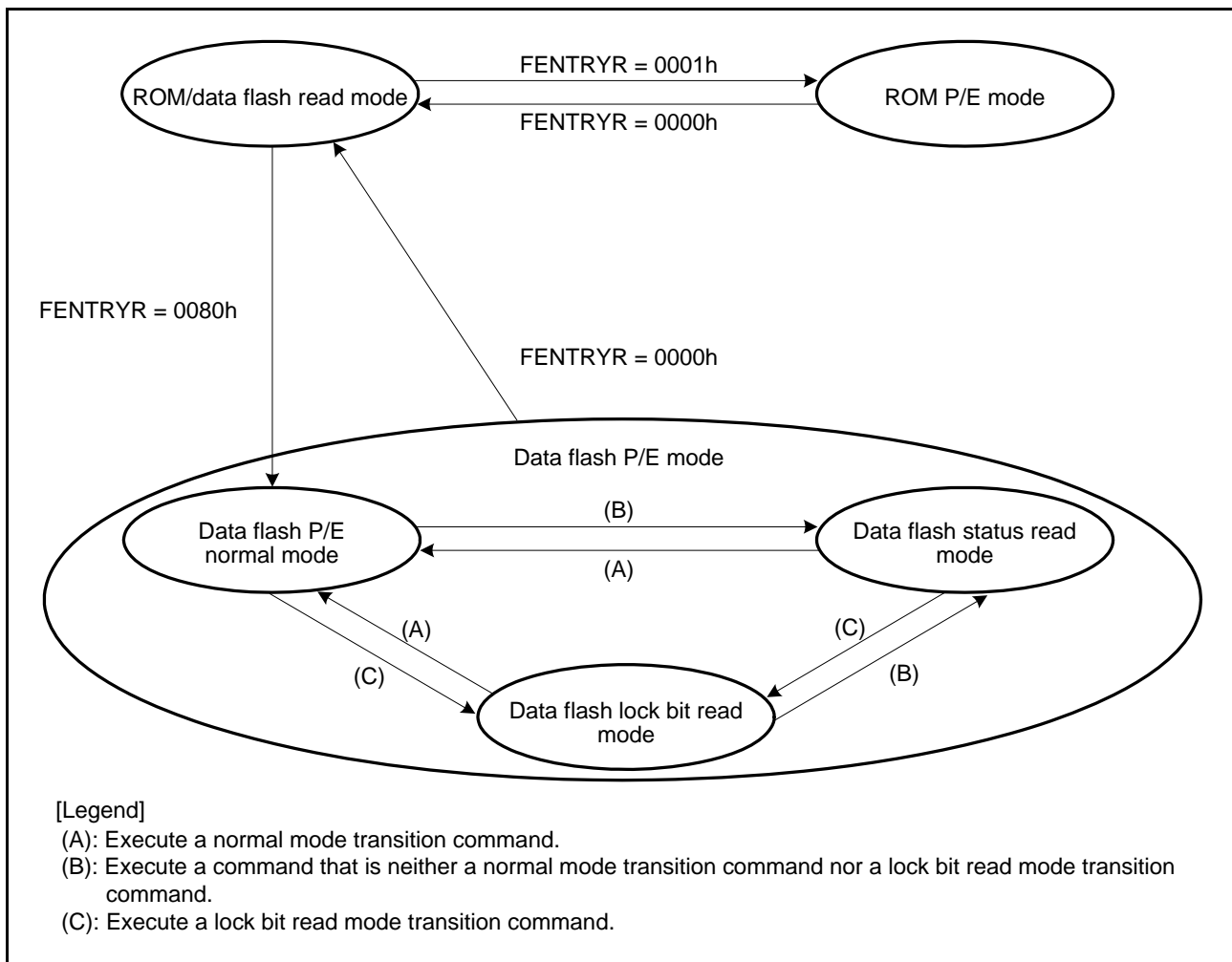


Figure 39.4 Mode Transitions of the FCU (Associated with the Data Flash)

39.6.1.1 ROM P/E Modes

The ROM P/E modes are for programming and erasure of the ROM.

For details on the ROM P/E modes, see section 38.6.1.2, ROM P/E Modes.

39.6.1.2 ROM/Data Flash Read Mode

This mode is for reading the ROM or data flash memory. The FCU does not accept commands.

The FCU enters this mode when both the FENTRYD bit and FENTRY0 bit in FENTRYR are set to 0.

39.6.1.3 Data Flash P/E Modes

These modes are for programming and erasure of the data flash memory. Reading out the data flash is not possible.

Data flash P/E normal mode, data flash status read mode, and data flash lock-bit read mode are the three data flash P/E modes.

(1) Data Flash P/E Normal Mode

The transition to data flash P/E normal mode is the first transition in the process of programming or erasing the data flash. The FCU enters this mode when the FENTRYD bit in FENTRYR is set to 1 and the FENTRY0 bit in FENTRYR is set to 0 in ROM/data flash read mode, or when the normal mode transition command is received in data flash P/E modes. Table 39.7 lists the acceptable commands in this mode.

Read access to an address within the data flash area causes a data-flash-access violation, and the FCU enters the command-locked state. High-speed reading of the ROM is possible.

(2) Data Flash Status Read Mode

The data flash status read mode is for reading information on the state of the data flash.

The FCU enters this mode when a command other than the normal mode transition and lock bit read mode transition command is received in data flash P/E modes. Data flash status read mode encompasses the states where the FRDY bit in FSTATR0 is 0 and the command-locked state after an error has occurred. Table 39.7 lists the acceptable commands in this mode.

Read access to an address within the data flash area will actually read out the value of the FSTATR0 register. High-speed reading of the ROM is possible.

(3) Data Flash Lock-Bit Read Mode

The data flash lock-bit read mode is for reading the values of the lock bits of the data flash. However, this is not possible because the data flash does not have lock bits.

The FCU enters this mode when a lock-bit read mode transition command is received in data flash P/E modes. Table 39.7 lists the acceptable commands in this mode.

Since the data flash does not have lock bits, data read out in read access to addresses within the data flash area are undefined. However, the access does not lead to a data-flash-access violation. High-speed reading of the ROM is possible.

39.6.2 FCU Commands

FCU commands consist of commands for mode transitions of the FCU and of commands for programming and erasure.

Table 39.5 lists the FCU commands for use with the data flash.

Table 39.5 FCU Commands for Use with Data Flash Memory

Command	Description
P/E Normal mode transition	Changes the mode to normal mode (see section 39.6.3, Connections between FCU Modes and Commands)
Status read mode transition	Changes the mode to status read mode (see section 39.6.3, Connections between FCU Modes and Commands)
Lock bit read mode transition (lock bit read 1)	Changes the mode to lock bit read mode (see section 39.6.3, Connections between FCU Modes and Commands)
Peripheral clock notification	Sets the frequency of the peripheral clock
Programming	Data flash programming (in 8-byte or 128-byte units)
Block erasure	Data flash erasure (in block units, with the lock bit being erased simultaneously)
P/E suspension	Suspends programming/erasure
P/E resumption	Resumes programming/erasure
Status register clearing	Clears the ILGLERR, ERSERR and PRGERR bits in FSTATR0 and releases the FCU from the command-locked state
Lock bit read 2/blank checking	Checks whether the specified block of data flash memory has been erased (is blank)

Commands other than the blank-checking command are also for use with the ROM.

The blank-checking command for the data flash memory is also used as the lock bit read 2 command for the ROM. That is, when the same command is issued for the ROM, a lock bit of the ROM is read out.

Commands for the FCU are issued by write access to addresses within the data flash area.

Table 39.6 shows the formats of the programming commands and the blank checking command. For the formats of FCU commands other than programming and blank checking commands, see 37.6.2, FCU Commands.

Write access as listed in Table 39.6 and in accord with certain conditions causes the FCU to execute processing for the corresponding command. For details on the conditions for acceptance of the individual FCU commands, see section 39.6.3, Connections between FCU Modes and Commands. For how to use the FCU commands, see section 39.6.4, FCU Command Usage.

Table 39.6 FCU Command Formats

Command	Number of bus cycles	First Cycle		Second Cycle		Third Cycle		4th to (N+2)th Cycles		(N+3)th Cycles	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Programming (8-byte programming; n = 4)	7	EA	E8h	EA	04h	WA	WDn	EA	WDn	EA	D0h
Programming (128-byte programming; n = 64)	67	EA	E8h	EA	40h	WA	WDn	EA	WDn	EA	D0h
Blank checking	2	EA	71h	BA	D0h	—	—	—	—	—	—

[Legend] Address column EA: Address within the data flash area.
 Any address in the range from 0010 0000h to 0010 7FFFh.
 WA: Correctly aligned address in 8-byte or 128-byte.
 BA: Address in an erasure block of the data flash. Any address within the target erasure block.
 Data columns WDn: nth word of data for programming (n = 1 to N)

39.6.3 Connections between FCU Modes and Commands

The sets of FCU commands that can be accepted in each of the FCU modes are fixed. Furthermore, which commands are acceptable in a given FCU mode varies according to the state of the FCU.

Issuing of an FCU command must follow checking of the FCU's state after transitions of the FCU mode.

Commands that are acceptable in the various FCU modes and states are listed in Table 39.7. Issuing a command that is not currently acceptable leads to the FCU being placed in the command-locked state (see section 39.7.2, Error Protection).

Issuing of an FCU command must follow checking of the values of the FRDY, ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and of the FCUERR bit in FSTATR1 after transitions of the FCU mode. Furthermore, the CMDLK bit in FASTAT can be checked to see if an error has occurred. The value of the CMDLK bit in FASTAT is the logical OR of the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FCUERR bit in FSTATR1.

Table 39.7 Acceptable Commands and the State and Mode (Data Flash P/E Mode) of the FCU

	P/E Normal Mode			Status read mode							Lock-bit read mode		
	Programming suspended	Erasure suspended	Other state	Programming or erasure	Processing to suspend programming or erasure	Blank checking	Programming suspended	Erasure suspended	Command-locked state	Other state	Programming suspended	Erasure suspended	Other state
FSTATR0.FRDY bit	1	1	1	0	0	0	1	1	0/1	1	1	1	1
FSTATR0.SUSRDY bit	0	0	0	1	0	0	0	0	0	0	0	0	0
FSTATR0.ERSSPD bit	0	1	0	0	0/1	0	0	1	0	0	0	1	0
FSTATR0.PRGSPD bit	1	0	0	0	0/1	0	1	0	0	0	1	0	0
FASTAT.CMDLK bit	0	0	0	0	0	0	0	0	1	0	0	0	0
P/E Normal mode transition	A	A	A	X	X	X	A	A	X	A	A	A	A
Status read transition	A	A	A	X	X	X	A	A	X	A	A	A	A
Lock-bit read transition (lock bit read 1)	A	A	A	X	X	X	A	A	X	A	A	A	A
Peripheral clock setting	X	X	A	X	X	X	X	X	X	A	X	X	A
Programming	X	*	A	X	X	X	X	*	X	A	X	*	A
Block erasure	X	X	A	X	X	X	X	X	X	A	X	X	A
P/E suspension	X	X	X	A	X	X	X	X	X	X	X	X	X
P/E resumption	A	A	X	X	X	X	A	A	X	X	A	A	X
Status register clearing	A	A	A	X	X	X	A	A	A	A	A	A	A
Blank checking	A	A	A	X	X	X	A	A	X	A	A	A	A

[Legend]

A: Acceptable

*: Only programming is acceptable for blocks other than the block where erasure was suspended

X: Not acceptable

39.6.4 FCU Command Usage

This section shows how to program and erase the data flash memory by using programming and block erasure commands, respectively, and how to check the state of erasure of the data flash by using the blank check command. For the method for transferring the firmware to the FCU RAM and the ways to use other FCU commands, see section 38.6.4, FCU Command Usage.

(1) Using the Peripheral Clock Notification Command

This command handles notification of the frequency of the peripheral clock. For details, see section 38.6.4, FCU Command Usage. Set the FENTRYD bit in FENTRYR to 1 and make settings to indicate an address within the data flash area.

(2) Programming

To program the data flash, use one of the programming commands.

Use byte access to write E8h to an address within the data flash area in the first cycle of the programming command, and the number of words (N)* to be programmed in the second cycle. Access the peripheral bus in words from the third cycle to cycle N + 2 of the command. In the third cycle, write the first word of data for programming to the address where the target area for programming starts. This address must be on an 8-byte boundary for 8-byte programming or on a 128-byte boundary address for 128-byte programming. After writing words to addresses in the data flash area N times, write byte D0h to an address within the data flash area in cycle N + 3; the FCU will then start actual programming of the data flash. Read the FRDY bit in FSTATR0 to confirm the completion of data flash programming.

If the area accessed in the third cycle to cycle N + 2 includes addresses that do not require programming, write FFFFh as the programming data for those addresses. To ignore the programming and erasure protection provided by the DFLWEK (k = 0, 1) setting, set the program/erase enable bit for the target block to 1 before programming starts.

Figure 39.5 shows the procedure for data flash programming.

Note: * N = 04h for 8-byte programming or N = 40h for 128-byte programming.

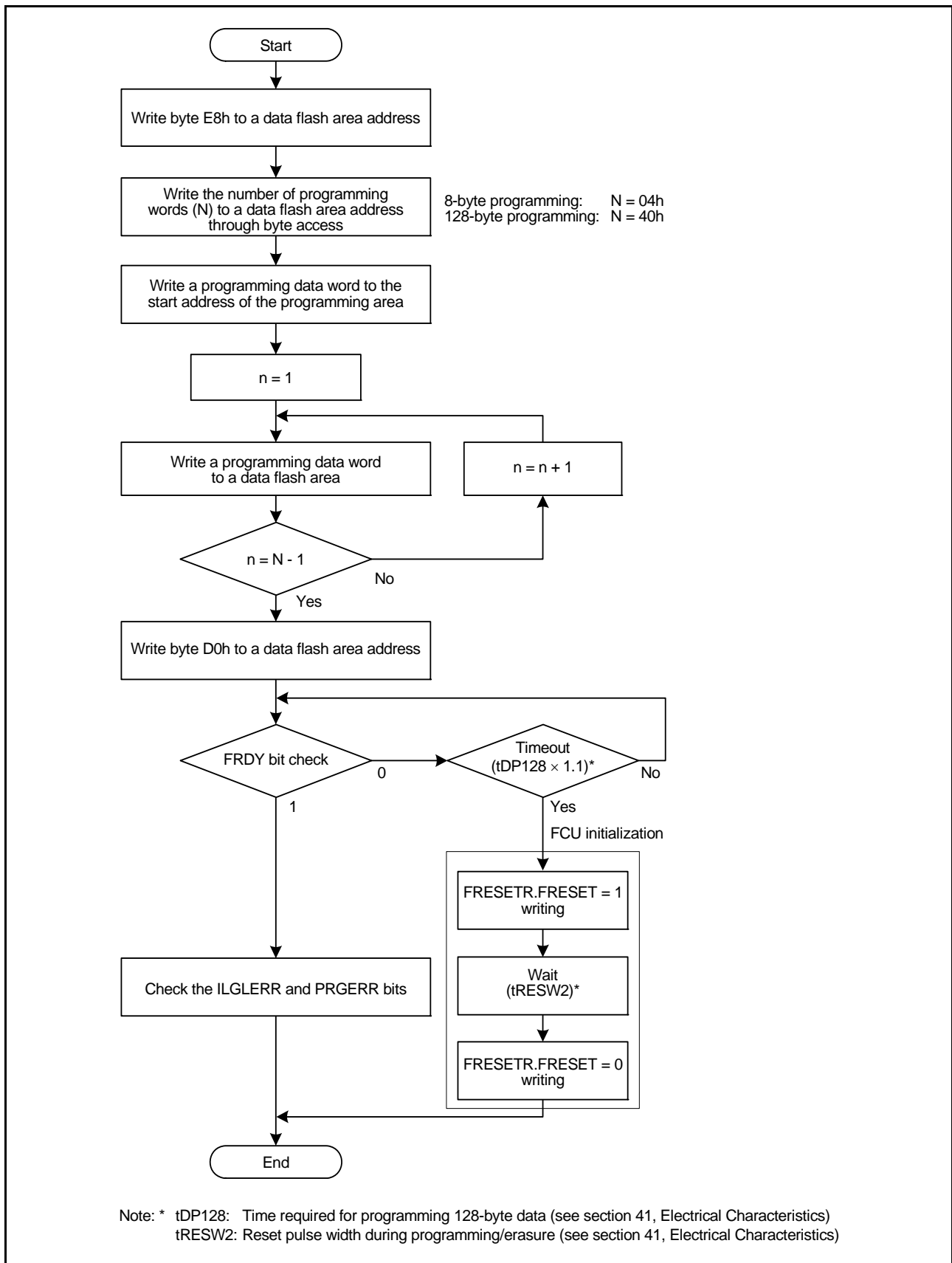


Figure 39.5 Procedure for Data Flash Programming

(3) Erasure

To erase the data flash, use the block erasure command. The data flash is erased in the same way as the ROM (see section 38, ROM (Flash Memory for Code Storage)).

Note that the data flash has a programming and erasure protection function that is controlled by DFLWE_k (k = 0, 1). Erasure can only be performed with protection provided by the DFLWE_k setting disabled, so set the programming/erasure enable bit for the target erasure block to 1 before issuing the erasure command.

(4) Blank Checking

Since using the CPU to read erased areas of the data flash produces undefined values, the blank checking command should be used to check whether the data flash has actually been erased. To make the blank checking command available for use, start by setting the FRDMD bit in FMODR to 1 to enable the command, and then specify the size and start address of the target area in DFLBCCNT. If the BCSIZE bit of DFLBCCNT is set to 1, checking will be performed for the entire erasure block (2 Kbytes) specified in the second cycle of the command. If the BCSIZE bit is set to 0, checking will be performed on the 8-byte range starting from the address obtained by adding the start address of the erased area as specified in the second cycle of the command and the value held by DFLBCCNT. In the first cycle of the command sequence, the value 71h is written as a byte unit to an address in the data flash. In the second cycle, when the value D0h is written to an address within the target area, the FCU starts blank checking of the data flash. Test the FRDY bit in the FSTATR0 register to check whether or not the check is complete. On completion of blank checking, check the BCST bit of DFLBCSTAT to see whether the target area has been erased or is filled with 0s and/or 1s via.

Figure 39.6 shows the procedure for blank checking of the data flash.

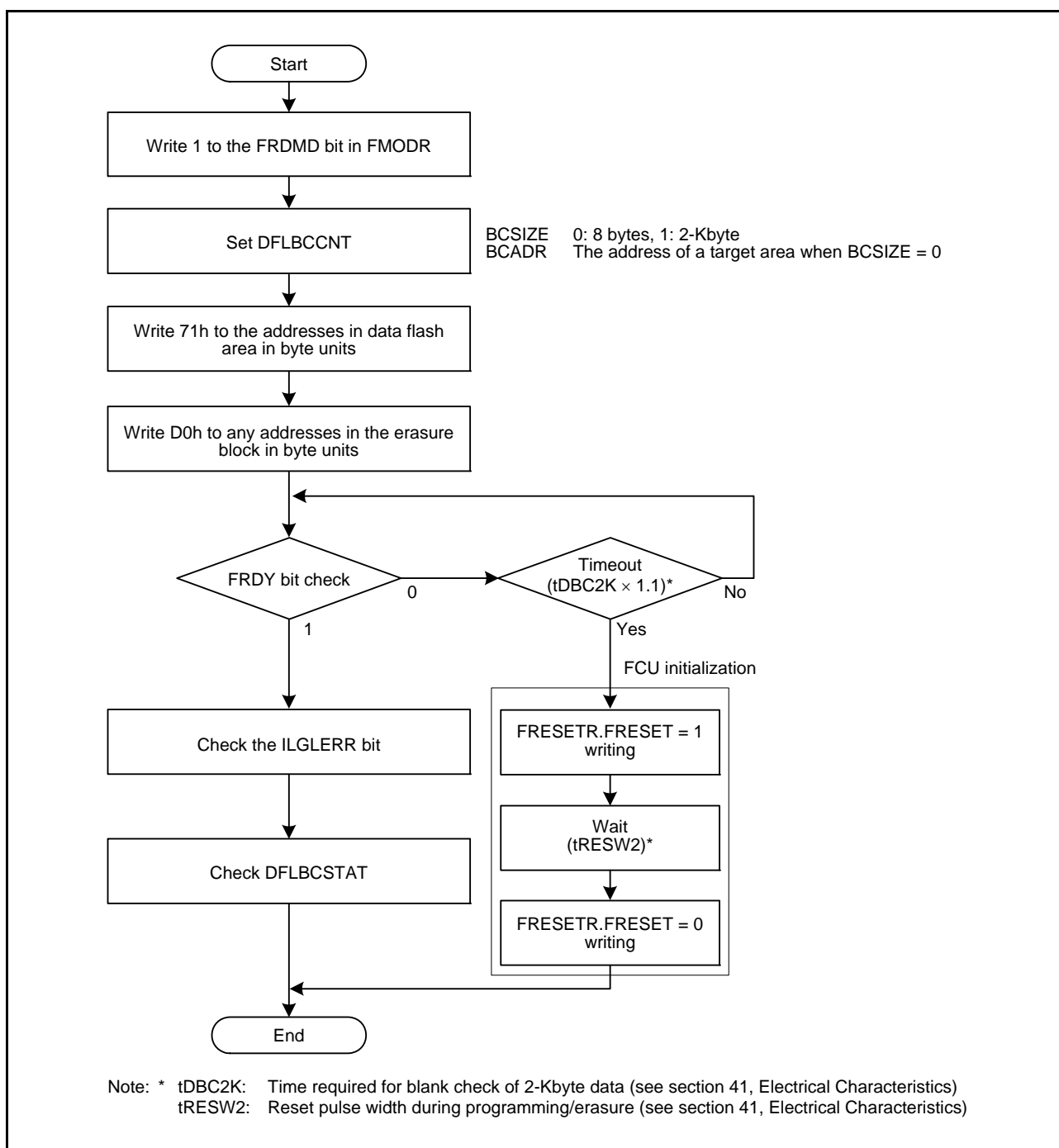


Figure 39.6 Procedure for Blank Checking of the Data Flash

39.7 Protection

There are two types of data flash programming/erasure protection: software protection and error protection.

39.7.1 Software Protection

In the software protection function, control register settings are used to disable data flash programming and erasure. If an attempt is made to issue a programming or erasure command for the data flash and the command violates current software protection, the FCU detects the error and enters the command-locked state.

(1) Protection through FWEPROR

If the FLWE[1:0] bits in FWEPROR are not set to 01b, programming cannot be performed in any mode.

(2) Protection through FENTRYR

When the FENTRYD bit in FENTRYR is 0, the ROM/data flash read mode is selected. Since the FCU does not accept commands in ROM/data flash read mode, data flash programming and erasure are disabled. If an attempt is made to issue an FCU command for the data flash in ROM/data flash read mode, the FCU detects an illegal command error and enters the command-locked state (see section 39.7.2, Error Protection).

(3) Protection through DFLWEk

When the DBWE_i (i = 15 to 00) bit in DFLWE_k (k = 0, 1) is 0, programming and erasure of block DB_i in the data mat is disabled. If an attempt is made to program or erase block DB_i while the DBWE_i bit is 0, the FCU detects a programming/erasure protection error and enters the command-locked state (see section 39.7.2, Error Protection).

(4) Protection through DFLREk

When the DBRE_i (i = 15 to 00) bit in DFLRE_k (k = 0, 1) is 0, reading of block DB_i in the data mat is disabled. If an attempt is made to read block DB_i while the DBRE_i bit is 0, the FCU detects a read protection error and enters the command-locked state (see section 39.7.2, Error Protection).

39.7.2 Error Protection

Error protection is the detection of errors in the issuing of FCU commands and of prohibited access, and response in the form of notification of the FCU malfunction and prohibition of the reception of further commands by the FCU (the FCU enters the command-locked state). When the FCU enters the command-locked state (FASTAT.CMDLK bit is 1), one or several of the status bits (FSTATR0.ILGLERR, ERSERR, and PRGERR bits, FSTATR1.FCUERR bit, and FASTAT.DFLAE, DFLRPE, and DFLWPE bits) are set to 1 and programming and erasure of the data-flash are prohibited by placing the FCU in the command-locked state. To release the FCU from the command-locked state, a status register clearing command must be issued with FASTAT set to 10h.

While the CMDLKIE bit in FAEINT is 1, a flash interface error (FIFERR) interrupt will be generated if the FCU enters the command-locked state (the CMDLK bit in FASTAT becomes 1). While a data flash-related interrupt enable bit (DFLAEIE, DFLRPEIE, or DFLWPEIE) in FAEINT is 1, an FIFERR interrupt will also be generated if the corresponding status bit (DFLAE, DFLRPE, or DFLWPE) in FASTAT becomes 1.

Table 39.8 shows the error protection types for the data flash and the values of the status bits (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the DFLAE, DFLRPE, and DFLWPE bits in FASTAT) after the detection of each type of error. For the error protection types used in common by the ROM and data flash (FENTRYR setting error, most illegal command errors, erasing errors, programming errors, and FCU errors, see section 39.7.2, Error Protection.

If the FCU enters the command-locked state due to a command other than a suspension command issued during programming or erasure processing, the FCU continues programming or erasing the data flash. In this state, the P/E suspension command cannot suspend programming or erasure. If a command is issued in the command-locked state, the ILGLERR bit becomes 1.

Table 39.8 Error Protection Types (for Data Flash Only)

Error	Description	ILGLERR	ERSERR	PRGERR	DFLAE	DFLRPE	DFLWPE	CMDLK
Illegal command error	The value specified in the second cycle of a programming command was neither 04h nor 40h.	1	0	0	0	0	0	1
	A lock bit programming command was issued for an area in the data flash while the FENTRYD bit of FENTRYR register was set to 1.	1	0	0	0	0	0	1
Data flash access error	A read access command was issued for the data flash area while FENTRYD = 1 in FENTRYR in data flash P/E normal mode.	1	0	0	1	0	0	1
	A write access command was issued for the data flash area while FENTRYD = 0.	1	0	0	1	0	0	1
	An access command was issued for the data flash area while the FENTRY0 bit in FENTRYR was 1.	1	0	0	1	0	0	1
Data flash read protect error	A read access command was issued for the data flash area while it was protected against programming and erasure by the DFLREk (k = 0, 1) setting.	1	0	0	0	1	0	1
Data flash programming protect error	A program/block erase command was issued for the data flash area while it was protected against reading by the DFLWEk (k = 0, 1) setting.	1	0	0	0	0	1	1

39.8 Boot Mode

To program or erase the data mat in boot mode, send control commands and programming data from the host. For the system configuration and settings in boot mode, see section 38.9, Boot Mode. This section describes only the commands dedicated for the data flash.

39.8.1 Inquiry/Selection Host Commands

Table 39.9 shows the inquiry/selection host commands dedicated to the data flash. The data mat inquiry and data mat information inquiry commands are used in the step of "Inquiry regarding mat programming information" in the flowchart shown in figure 38.30 (Example of Procedure to Use Inquiry/Selection Host Commands for User Mat and User Boot Mat) in section 38.9.6, Inquiry/Selection Host Command Wait State.

Table 39.9 Inquiry/Selection Host Commands (only for Data Flash)

Host Command Name	Function
Data mat inquiry	Inquires regarding the availability of data mat
Data mat information inquiry	Inquires regarding the number of data mats and the start and end addresses

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to the RX62N/RX621 and the "response" indicates a response sent from the RX62N/RX621 to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by the RX62N/RX621 becomes 00h.

(1) Data Mat Inquiry

In response to a data mat inquiry command sent from the host, the RX62N/RX621 returns the information concerning the availability of data mats.

Command

2Ah

Response

3Ah	Size	Mat availability	SUM
-----	------	------------------	-----

[Legend]

Size (1 byte): Number of characters in the mat availability field (fixed at 1)

Mat availability (1 byte): Availability of data mats (fixed at 21h)

21h: Data mat is available

SUM (1 byte): Checksum

(2) Data Mat Information Inquiry

In response to a data mat information inquiry command sent from the host, the RX62N/RX621 returns the number of data mat areas and their addresses.

Command

2Bh

Response	3Bh	Size	Area count	
	Area start address			
	Area end address			
	Area start address			
	Area end address			
	:			
	Area start address			
	Area end address			
	SUM			

[Legend]

- Size (1 byte): Total number of bytes in the area count, area start address, and area end address fields
- Area count (1 byte): Number of data mat areas (consecutive areas are counted as one area)
- Area start address (4 bytes): Start address of a data mat area
- Area end address (4 bytes): End address of a data mat area
- SUM (1 byte): Checksum

The information concerning the block configuration in the data mat is included in the response to the erasure block information inquiry command (see section 38.9.6, Inquiry/Selection Host Command Wait State).

39.8.2 Programming/Erasing Host Commands

Table 39.10 shows the programming/erasing host commands dedicated to the data flash. Data flash-dedicated host commands are provided only for checksum and blank check of the data flash; the programming, erasing, and reading commands are used in common for the ROM and data flash.

To program the data mat, issue from the host a user mat programming selection command and then a 256-byte programming command specifying a data mat address as the programming address. To erase the data mat, issue an erasure selection command and then a block erasure command specifying an erasure block in the data mat. The information concerning the erasure block in the data mat is included in the response to the erasure block information inquiry command. To read data from the data mat, select the user mat through a memory read command specifying a data mat address as the read address.

For the user mat programming selection, user boot mat programming selection, 256-byte programming, erasure selection, block erasure, and memory read commands, refer to section 38.9.8, Programming/Erasure Host Command Wait State. For the erasure block information inquiry command, refer to section 38.9.6, Inquiry/Selection Host Command Wait State.

Table 39.10 Programming/Erasure Host Commands (only for Data Flash)

Host Command Name	Function
Data mat checksum	Performs checksum verification for the data mat
Data mat blank check	Checks whether the data mat is blank

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to the RX62N/RX621 and the "response" indicates a response sent from the RX62N/RX621 to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by the RX62N/RX621 becomes 00h.

(1) Data Mat Checksum

In response to a data mat checksum command sent from the host, the RX62N/RX621 sums the data mat data in byte units and returns the result (checksum).

Command

61h

Response

71h	Size	Mat checksum	SUM
-----	------	--------------	-----

[Legend]

Size (1 byte): Number of bytes in the mat checksum field (fixed at 4)

Mat checksum (4 bytes): Checksum of the data mat data

SUM (4 bytes): Checksum (for the response data)

(2) Data Mat Blank Check

In response to a data mat blank check command sent from the host, the RX62N/RX621 checks whether the data mat is completely erased. When the data mat is completely erased, the RX62N/RX621 returns a response (06h). If the data mat has an unerased area, the RX62N/RX621 returns an error response (sends E2h and 52h in that order).

Command

62h

Response

06h

Error response

E2h	52h
-----	-----

39.9 Usage Notes

(1) Protection of Data Mat Immediately after a Reset

As the initial values of DFLREk and DFLWEk ($k = 0, 1$) are 0000h, programming, erasure, and reading of the data mat are disabled immediately after a reset. To read data from the data mat, set DFLREk appropriately before accessing the data mat. To program or erase the data mat, set DFLWEk appropriately before issuing an FCU command for programming or erasure. If an attempt is made to read, program, or erase the data mat without setting the registers, the FCU detects the error and enters the command-locked state.

(2) Other Points to Note

The other points to note are the same as for the ROM. See section 38.13, Usage Notes. However, for the data flash, notes on programming/erasing also apply to the blank checking. However, data-flash memory also has a blank-checking facility. Accordingly, for "programming and erasure", please read "programming, erasure, and blank-checking".

40. Boundary Scan

The RX62N, RX621 Group has boundary scan function, and this function is supported in the 176-pin LFBGA, 145-pin TFLGA, 144-pin LQFP, and 85-pin TFLGA version. This function is not supported only in the 100-pin LQFP version.

The boundary scan is a serial I/O interface based on the JTAG (Joint Test Action Group, IEEE Std.1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture).

40.1 Features

Table 40.1 lists the specifications of boundary scan.

Figure 40.1 shows the block diagram of the boundary scan function.

Table 40.1 Specifications of Boundary Scan

Item	Description
Boundary scan enabled/disabled	Boundary scan is enabled when the EMLE pin is driven low and the BSCANP pin is driven high.
Dedicated boundary scan pins	The following pins are dedicated for JTAG, when boundary scan function is enabled (TDO/TCK/TDI/TMS/TRST#). 176-pin LFBGA: PF0/PF1/PF2/PF3/PF4 145-pin TFLGA/144-pin LQFP: P26/P27/P30/P31/P34 85-pin TFLGA: P26/P27/P30/P31/P34
Six test modes	BYPASS mode EXTEST mode SAMPLE/PRELOAD mode CLAMP mode HIGHZ mode IDCODE mode

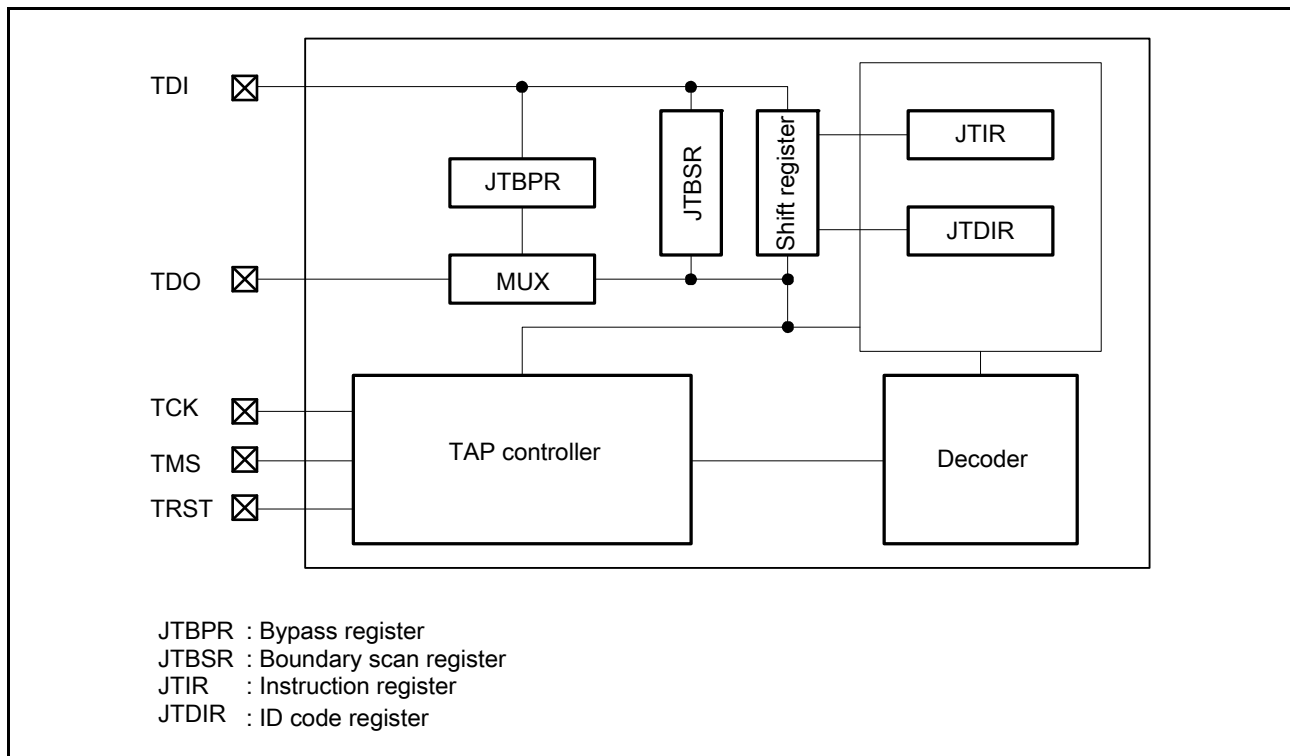


Figure 40.1 Block Diagram of Boundary Scan Function

Table 40.2 shows the I/O pins used in the boundary scan function.

Table 40.2 Pin Configuration

Pin Name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. Input the clock the duty cycle of which is 50 percent when boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin
TRST#	Input	Test reset input pin

40.2 Register Descriptions

Table 40.3 shows the lists the boundary scan registers.

Table 40.3 lists the boundary scan registers

Register Name	Symbol	Value after Reset	Address	Access Size
Instruction register	JTIR	55h	—	—
ID code register	JTIDR	080B B447h	—	—
Bypass register	JTBPR	Undefined	—	—
Boundary scan register	JTBSR	Undefined	—	—

Instructions can be input to the instruction register (JTIR) via the test data input pin (TDI) by serial transfer.

The bypass register (JTBPR), which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.

The boundary scan register (JTBSR), which is a JTBSR-bit register (see Table 40.6), is connected between the TDI and TDO pins when test data are being shifted in.

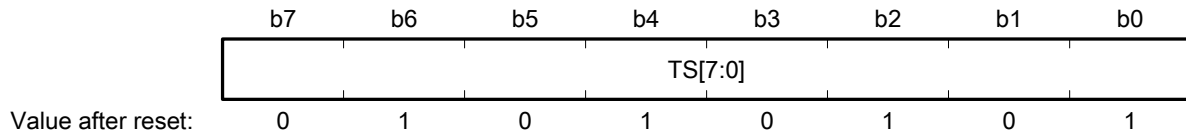
None of the registers is accessible from the CPU.

Table 40.4 shows the availability of serial transfer for the registers.

Table 40.4 Serial Transfers for Registers

Register Abbreviation	Serial Input	Serial Output
Instruction register (JTIR)	Available	Not available
ID code register (JTIDR)	Not available	Available
Bypass register (JTBPR)	Available	Available
Boundary scan register (JTBSR)	Available	Available

40.2.1 Instruction Register (JTIR)



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TS[7:0]	Test Bit Set	The command configuration is as shown in Table 40.5.	—

Table 40.5 Command Configuration

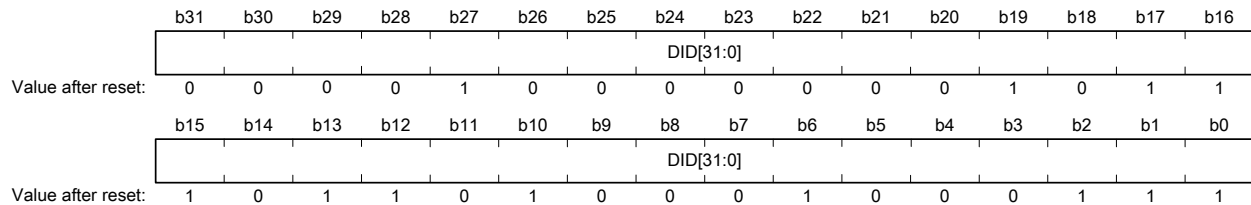
TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0	Instruction
0	0	0	0	0	0	0	0	EXTEST
0	1	0	0	0	0	0	0	SAMPLE/PRELOAD
0	1	0	1	0	1	0	1	IDCODE (initial value)
1	1	0	1	0	0	0	0	CLAMP
1	0	0	0	0	0	0	0	HIGHZ
1	1	1	1	1	1	1	1	BYPASS
Other than above								Reserved

JTIR is an 8-bit register.

JTAG instructions can be transferred to JTIR by serial input from the TDI pin.

JTIR is initialized when the TRST# signal is low level, when the TAP controller is in the Test-Logic-Reset state.

40.2.2 ID Code Register (JTID)



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DID[31:0]	Reserved	JTIDR is a register with the fixed value that indicates the device IDCODE.	—

JTID is a 32-bit register.

JTID data is output from the TDO pin when the IDCODE instruction has been executed.

40.2.3 Bypass Register (JTBPR)

JTBPR is a 1-bit register and is connected between the TDI and TDO pins when JTIR is set to BYPASS mode.

JTBPR cannot be read from or written to by the CPU.

40.2.4 Boundary Scan Register (JTBSR)

JTBSR is a shift register to control the external input and output pins of this LSI and is distributed across the pads.

The EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ instructions are issued to apply JTBSR in boundary-scan testing conformant to the JTAG standard.

Table 40.6 shows the correspondence between the JTBSR bits and the pins of this LSI.

The value after reset is undefined.

Table 40.6 Boundary Scan Register (LFBGA176)
(1 / 9)

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
C3	P05	Output	356
		Output enable	355
		Input	354
C2	P03	Output	353
		Output enable	352
		Input	351
B1	P02	Output	350
		Output enable	349
		Input	348
D2	P01	Output	347
		Output enable	346
		Input	345
C1	P00	Output	344
		Output enable	343
		Input	342
F2	WDTOVF#	Output	341
		Output enable	340
G4	MDE	Input	339
G2	MD1	Input	338
G3	MD0	Input	337
H2	P35	Input	336
K1	P33	Output	335
		Output enable	334
		Input	333
J2	P32	Output	332
		Output enable	331
		Input	330
M2	P25	Output	329
		Output enable	328
		Input	327
P1	P24	Output	326
		Output enable	325
		Input	324
N2	P23	Output	323
		Output enable	322
		Input	321
M3	P22	Output	320
		Output enable	319
		Input	318
R1	P21	Output	317
		Output enable	316
		Input	315

Table 40.6 Boundary Scan Register (LFBGA176)
(2 / 9)

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
N3	P20	Output	314
		Output enable	313
		Input	312
N4	P17	Output	311
		Output enable	310
		Input	309
P3	P16	Output	308
		Output enable	307
		Input	306
N5	P15	Output	305
		Output enable	304
		Input	303
P4	P14	Output	302
		Output enable	301
		Input	300
P5	P13	Output	299
		Output enable	298
		Input	297
R3	P12	Output	296
		Output enable	295
		Input	294
M5	P11	Output	293
		Output enable	292
		Input	291
N6	P57	Output	290
		Output enable	289
		Input	288
P7	P56	Output	287
		Output enable	286
		Input	285
M6	P55	Output	284
		Output enable	283
		Input	282
N7	P10	Output	281
		Output enable	280
		Input	279
M7	P54	Output	278
		Output enable	277
		Input	276
R9	P85	Output	275
		Output enable	274
		Input	273

Table 40.6 Boundary Scan Register (LFBGA176)
(3 / 9)

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
R10	P53	Output	272
		Output enable	271
		Input	270
P9	P84	Output	269
		Output enable	268
		Input	267
N8	P52	Output	266
		Output enable	265
		Input	264
M8	P51	Output	263
		Output enable	262
		Input	261
P10	P50	Output	260
		Output enable	259
		Input	258
R11	P83	Output	257
		Output enable	256
		Input	255
R12	PC7	Output	254
		Output enable	253
		Input	252
M10	PC6	Output	251
		Output enable	250
		Input	249
N10	PC5	Output	248
		Output enable	247
		Input	246
P11	P82	Output	245
		Output enable	244
		Input	243
M11	P81	Output	242
		Output enable	241
		Input	240
R13	P80	Output	239
		Output enable	238
		Input	237
P12	PC4	Output	236
		Output enable	235
		Input	234
N11	PC3	Output	233
		Output enable	232
		Input	231

Table 40.6 Boundary Scan Register (LFBGA176)
(4 / 9)

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
R14	P77	Output	230
		Output enable	229
		Input	228
P13	P76	Output	227
		Output enable	226
		Input	225
N12	PC2	Output	224
		Output enable	223
		Input	222
R15	P75	Output	221
		Output enable	220
		Input	219
N13	P74	Output	218
		Output enable	217
		Input	216
P14	PC1	Output	215
		Output enable	214
		Input	213
M12	PC0	Output	212
		Output enable	211
		Input	210
N14	P73	Output	209
		Output enable	208
		Input	207
P15	PB7	Output	206
		Output enable	205
		Input	204
M14	PB6	Output	203
		Output enable	202
		Input	201
N15	PB5	Output	200
		Output enable	199
		Input	198
L13	PB4	Output	197
		Output enable	196
		Input	195
L14	PB3	Output	194
		Output enable	193
		Input	192
M15	PB2	Output	191
		Output enable	190
		Input	189

**Table 40.6 Boundary Scan Register (LFBGA176)
(5 / 9)**

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
K12	PB1	Output	188
		Output enable	187
		Input	186
K14	P72	Output	185
		Output enable	184
		Input	183
K13	P71	Output	182
		Output enable	181
		Input	180
K15	PB0	Output	179
		Output enable	178
		Input	177
J14	PA7	Output	176
		Output enable	175
		Input	174
J15	PA6	Output	173
		Output enable	172
		Input	171
J13	PA5	Output	170
		Output enable	169
		Input	168
H14	PA4	Output	167
		Output enable	166
		Input	165
H15	PA3	Output	164
		Output enable	163
		Input	162
H12	PG7	Output	161
		Output enable	160
		Input	159
H13	PA2	Output	158
		Output enable	157
		Input	156
G14	PG6	Output	155
		Output enable	154
		Input	153
G15	PA1	Output	152
		Output enable	151
		Input	150
F15	PG5	Output	149
		Output enable	148
		Input	147

**Table 40.6 Boundary Scan Register (LFBGA176)
(6 / 9)**

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
F14	PA0	Output	146
		Output enable	145
		Input	144
F12	PG4	Output	143
		Output enable	142
		Input	141
E15	P67	Output	140
		Output enable	139
		Input	138
F13	PG3	Output	137
		Output enable	136
		Input	135
E14	P66	Output	134
		Output enable	133
		Input	132
E12	PG2	Output	131
		Output enable	130
		Input	129
D15	P65	Output	128
		Output enable	127
		Input	126
D14	PE7	Output	125
		Output enable	124
		Input	123
C15	PE6	Output	122
		Output enable	121
		Input	120
B15	P70	Output	119
		Output enable	118
		Input	117
C14	PE5	Output	116
		Output enable	115
		Input	114
D13	PE4	Output	113
		Output enable	112
		Input	111
C13	PE3	Output	110
		Output enable	109
		Input	108
B14	PE2	Output	107
		Output enable	106
		Input	105

**Table 40.6 Boundary Scan Register (LFBGA176)
(7 / 9)**

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
A15	PE1	Output	104
		Output enable	103
		Input	102
C12	PE0	Output	101
		Output enable	100
		Input	99
B13	P64	Output	98
		Output enable	97
		Input	96
A14	P63	Output	95
		Output enable	94
		Input	93
B12	P62	Output	92
		Output enable	91
		Input	90
A13	P61	Output	89
		Output enable	88
		Input	87
B11	P60	Output	86
		Output enable	85
		Input	84
A12	PD7	Output	83
		Output enable	82
		Input	81
D10	PG1	Output	80
		Output enable	79
		Input	78
B10	PD6	Output	77
		Output enable	76
		Input	75
A11	PG0	Output	74
		Output enable	73
		Input	72
C10	PD5	Output	71
		Output enable	70
		Input	69
A10	PD4	Output	68
		Output enable	67
		Input	66
B9	P97	Output	65
		Output enable	64
		Input	63

**Table 40.6 Boundary Scan Register (LFBGA176)
(8 / 9)**

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
A9	PD3	Output	62
		Output enable	61
		Input	60
B8	P96	Output	59
		Output enable	58
		Input	57
A8	PD2	Output	56
		Output enable	55
		Input	54
D8	P95	Output	53
		Output enable	52
		Input	51
B7	PD1	Output	50
		Output enable	49
		Input	48
C8	P94	Output	47
		Output enable	46
		Input	45
A7	PD0	Output	44
		Output enable	43
		Input	42
D7	P93	Output	41
		Output enable	40
		Input	39
C7	P92	Output	38
		Output enable	37
		Input	36
B6	P91	Output	35
		Output enable	34
		Input	33
A6	P90	Output	32
		Output enable	31
		Input	30
B5	P47	Output	29
		Output enable	28
		Input	27
A5	P46	Output	26
		Output enable	25
		Input	24
A4	P45	Output	23
		Output enable	22
		Input	21

Table 40.6 Boundary Scan Register (LFBGA176)
(9 / 9)

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
B4	P44	Output	20
		Output enable	19
		Input	18
D5	P43	Output	17
		Output enable	16
		Input	15
A3	P42	Output	14
		Output enable	13
		Input	12
D4	P41	Output	11
		Output enable	10
		Input	9
C5	P40	Output	8
		Output enable	7
		Input	6
C4	P07	Output	5
		Output enable	4
		Input	3
		To TDO	

**Table 40.7 Boundary Scan Register (LGA145/
 LQFP144) (1 / 8)**

From TDI				
Pin No		Pin Name	Input/Output	Bit Name
LGA145	LQFP144			
C2	2	P05	Output	356
			Output enable	355
			Input	354
B1	4	P03	Output	353
			Output enable	352
			Input	351
D3	6	P02	Output	350
			Output enable	349
			Input	348
C1	7	P01	Output	347
			Output enable	346
			Input	345
E3	8	P00	Output	344
			Output enable	343
			Input	342
F3	11	WDTOVF#	Output	341
			Output enable	340
F4	13	MDE	Input	339
G3	15	MD1	Input	338
G4	16	MD0	Input	337
J3	24	P35	Input	336
J1	26	P33	Output	335
			Output enable	334
			Input	333
J4	27	P32	Output	332
			Output enable	331
			Input	330
L1	32	P25	Output	329
			Output enable	328
			Input	327
K2	33	P24	Output	326
			Output enable	325
			Input	324
M1	34	P23	Output	323
			Output enable	322
			Input	321
L2	35	P22	Output	320
			Output enable	319
			Input	318
N1	36	P21	Output	317
			Output enable	316
			Input	315

**Table 40.7 Boundary Scan Register (LGA145/
 LQFP144) (2 / 8)**

From TDI				
Pin No		Pin Name	Input/Output	Bit Name
LGA145	LQFP144			
M2	37	P20	Output	314
			Output enable	313
			Input	312
L3	38	P17	Output	311
			Output enable	310
			Input	309
N2	40	P16	Output	308
			Output enable	307
			Input	306
M4	42	P15	Output	305
			Output enable	304
			Input	303
M5	43	P14	Output	302
			Output enable	301
			Input	300
N4	44	P13	Output	299
			Output enable	298
			Input	297
L4	45	P12	Output	296
			Output enable	295
			Input	294
L6	50	P56	Output	287
			Output enable	286
			Input	285
M7	51	P55	Output	284
			Output enable	283
			Input	282
N7	52	P54	Output	278
			Output enable	277
			Input	276
K5	53	P53	Output	272
			Output enable	271
			Input	270
L7	54	P52	Output	266
			Output enable	265
			Input	264
N8	55	P51	Output	263
			Output enable	262
			Input	261
M8	56	P50	Output	260
			Output enable	259
			Input	258

**Table 40.7 Boundary Scan Register (LGA145/
LQFP144) (3 / 8)**

Pin No		Pin Name	Input/Output	Bit Name
LGA145	LQFP144			
L8	58	P83	Output	257
			Output enable	256
			Input	255
K7	60	PC7	Output	254
			Output enable	253
			Input	252
M9	61	PC6	Output	251
			Output enable	250
			Input	249
N10	62	PC5	Output	248
			Output enable	247
			Input	246
K8	63	P82	Output	245
			Output enable	244
			Input	243
L9	64	P81	Output	242
			Output enable	241
			Input	240
M10	65	P80	Output	239
			Output enable	238
			Input	237
N11	66	PC4	Output	236
			Output enable	235
			Input	234
K9	67	PC3	Output	233
			Output enable	232
			Input	231
L10	68	P77	Output	230
			Output enable	229
			Input	228
N12	69	P76	Output	227
			Output enable	226
			Input	225
M11	70	PC2	Output	224
			Output enable	223
			Input	222
L11	71	P75	Output	221
			Output enable	220
			Input	219
N13	72	P74	Output	218
			Output enable	217
			Input	216

**Table 40.7 Boundary Scan Register (LGA145/
LQFP144) (4 / 8)**

Pin No		Pin Name	Input/Output	Bit Name
LGA145	LQFP144			
M12	73	PC1	Output	215
			Output enable	214
			Input	213
K12	75	PC0	Output	212
			Output enable	211
			Input	210
K11	77	P73	Output	209
			Output enable	208
			Input	207
K10	78	PB7	Output	206
			Output enable	205
			Input	204
L13	79	PB6	Output	203
			Output enable	202
			Input	201
J12	80	PB5	Output	200
			Output enable	199
			Input	198
J11	81	PB4	Output	197
			Output enable	196
			Input	195
K13	82	PB3	Output	194
			Output enable	193
			Input	192
J10	83	PB2	Output	191
			Output enable	190
			Input	189
H12	84	PB1	Output	188
			Output enable	187
			Input	186
J13	85	P72	Output	185
			Output enable	184
			Input	183
H11	86	P71	Output	182
			Output enable	181
			Input	180
H10	87	PB0	Output	179
			Output enable	178
			Input	177
H13	88	PA7	Output	176
			Output enable	175
			Input	174

**Table 40.7 Boundary Scan Register (LGA145/
LQFP144) (5 / 8)**

Pin No		Pin Name	Input/Output	Bit Name
LGA145	LQFP144			
G12	89	PA6	Output	173
			Output enable	172
			Input	171
G11	90	PA5	Output	170
			Output enable	169
			Input	168
G13	92	PA4	Output	167
			Output enable	166
			Input	165
F11	94	PA3	Output	164
			Output enable	163
			Input	162
F13	95	PA2	Output	158
			Output enable	157
			Input	156
F10	96	PA1	Output	152
			Output enable	151
			Input	150
E12	97	PA0	Output	146
			Output enable	145
			Input	144
E11	98	P67	Output	140
			Output enable	139
			Input	138
E13	99	P66	Output	134
			Output enable	133
			Input	132
E10	100	P65	Output	128
			Output enable	127
			Input	126
D12	101	PE7	Output	125
			Output enable	124
			Input	123
D13	102	PE6	Output	122
			Output enable	121
			Input	120
C13	104	P70	Output	119
			Output enable	118
			Input	117
C11	106	PE5	Output	116
			Output enable	115
			Input	114

**Table 40.7 Boundary Scan Register (LGA145/
LQFP144) (6 / 8)**

Pin No		Pin Name	Input/Output	Bit Name
LGA145	LQFP144			
B13	107	PE4	Output	113
			Output enable	112
			Input	111
C12	108	PE3	Output	110
			Output enable	109
			Input	108
B11	109	PE2	Output	107
			Output enable	106
			Input	105
B12	110	PE1	Output	104
			Output enable	103
			Input	102
B10	111	PE0	Output	101
			Output enable	100
			Input	99
A13	112	P64	Output	98
			Output enable	97
			Input	96
C10	113	P63	Output	95
			Output enable	94
			Input	93
A12	114	P62	Output	92
			Output enable	91
			Input	90
C9	115	P61	Output	89
			Output enable	88
			Input	87
A11	117	P60	Output	86
			Output enable	85
			Input	84
C8	119	PD7	Output	83
			Output enable	82
			Input	81
A10	120	PD6	Output	77
			Output enable	76
			Input	75
B8	121	PD5	Output	71
			Output enable	70
			Input	69
D8	122	PD4	Output	68
			Output enable	67
			Input	66

**Table 40.7 Boundary Scan Register (LGA145/
LQFP144) (7 / 8)**

Pin No		Pin Name	Input/Output	Bit Name
LGA145	LQFP144			
A9	123	PD3	Output	62
			Output enable	61
			Input	60
C7	124	PD2	Output	56
			Output enable	55
			Input	54
B7	125	PD1	Output	50
			Output enable	49
			Input	48
A8	126	PD0	Output	44
			Output enable	43
			Input	42
D7	127	P93	Output	41
			Output enable	40
			Input	39
C6	128	P92	Output	38
			Output enable	37
			Input	36
A7	129	P91	Output	35
			Output enable	34
			Input	33
B6	131	P90	Output	32
			Output enable	31
			Input	30
A6	133	P47	Output	29
			Output enable	28
			Input	27
C5	134	P46	Output	26
			Output enable	25
			Input	24
B5	135	P45	Output	23
			Output enable	22
			Input	21
A5	136	P44	Output	20
			Output enable	19
			Input	18
D4	137	P43	Output	17
			Output enable	16
			Input	15
A4	138	P42	Output	14
			Output enable	13
			Input	12

**Table 40.7 Boundary Scan Register (LGA145/
LQFP144) (8 / 8)**

Pin No		Pin Name	Input/Output	Bit Name
LGA145	LQFP144			
C4	139	P41	Output	11
			Output enable	10
			Input	9
B4	141	P40	Output	8
			Output enable	7
			Input	6
B2	144	P07	Output	5
			Output enable	4
			Input	3
To TDO				

Table 40.8 Boundary Scan Register (LGA85) (1 / 4)

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
C1	P03	Output	353
		Output enable	352
		Input	351
D1	MDE	Input	339
C7	MD1	Input	338
D3	MD0	Input	337
G1	P35	Input	336
G2	P33	Output	335
		Output enable	334
		Input	333
F3	P32	Output	332
		Output enable	331
		Input	330
J1	P25	Output	329
		Output enable	328
		Input	327
J3	P24	Output	326
		Output enable	325
		Input	324
J2	P23	Output	323
		Output enable	322
		Input	321
K2	P22	Output	320
		Output enable	319
		Input	318
K1	P21	Output	317
		Output enable	316
		Input	315
K3	P20	Output	314
		Output enable	313
		Input	312
J4	P16	Output	308
		Output enable	307
		Input	306
H5	P14	Output	302
		Output enable	301
		Input	300
H6	P13	Output	299
		Output enable	298
		Input	297
K5	P12	Output	296
		Output enable	295
		Input	294

Table 40.8 Boundary Scan Register (LGA85) (2 / 4)

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
H7	P53	Output	272
		Output enable	271
		Input	270
J7	P52	Output	266
		Output enable	265
		Input	264
J8	P51	Output	263
		Output enable	262
		Input	261
K8	P50	Output	260
		Output enable	259
		Input	258
J9	PC3	Output	233
		Output enable	232
		Input	231
K9	PC2	Output	224
		Output enable	223
		Input	222
K10	PC1	Output	215
		Output enable	214
		Input	213
J10	PC0	Output	212
		Output enable	211
		Input	210
H9	PB7	Output	206
		Output enable	205
		Input	204
H8	PB6	Output	203
		Output enable	202
		Input	210
G9	PB5	Output	200
		Output enable	199
		Input	198
H10	PB4	Output	197
		Output enable	196
		Input	195
G8	PB3	Output	194
		Output enable	193
		Input	192
F9	PB2	Output	191
		Output enable	190
		Input	189

Table 40.8 Boundary Scan Register (LGA85) (3 / 4)

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
G10	PB1	Output	188
		Output enable	187
		Input	186
E9	PB0	Output	179
		Output enable	178
		Input	177
F10	PA7	Output	176
		Output enable	175
		Input	174
D9	PA6	Output	173
		Output enable	172
		Input	171
C9	PA5	Output	170
		Output enable	169
		Input	168
E10	PA4	Output	167
		Output enable	166
		Input	165
B9	PA3	Output	164
		Output enable	163
		Input	162
D10	PA2	Output	158
		Output enable	157
		Input	156
B10	PA1	Output	152
		Output enable	151
		Input	150
C10	PA0	Output	146
		Output enable	145
		Input	144
A9	PD7	Output	83
		Output enable	82
		Input	81
A10	PD6	Output	77
		Output enable	76
		Input	75
A8	PD5	Output	71
		Output enable	70
		Input	69
A7	PD4	Output	68
		Output enable	67
		Input	66

Table 40.8 Boundary Scan Register (LGA85) (4 / 4)

From TDI			
Pin No	Pin Name	Input/Output	Bit Name
B8	PD3	Output	62
		Output enable	61
		Input	60
B7	PD2	Output	56
		Output enable	55
		Input	54
A6	PD1	Output	50
		Output enable	49
		Input	48
B6	PD0	Output	44
		Output enable	43
		Input	42
A5	P47	Output	29
		Output enable	28
		Input	27
B5	P46	Output	26
		Output enable	25
		Input	24
C6	P45	Output	23
		Output enable	22
		Input	21
C5	P44	Output	20
		Output enable	19
		Input	18
A4	P43	Output	17
		Output enable	16
		Input	15
B4	P42	Output	14
		Output enable	13
		Input	13
C4	P41	Output	11
		Output enable	10
		Input	9
C3	P40	Output	8
		Output enable	7
		Input	6
A1	P05	Output	2
		Output enable	1
		Input	0
To TDO			

40.3 Operations

The boundary scan functionality is valid when the RES# pin is driven high, EMLE pin is driven low and the BSCANP pin is driven high.

40.3.1 TAP Controller

Figure 40.2 shows the state transition diagram of the TAP controller.

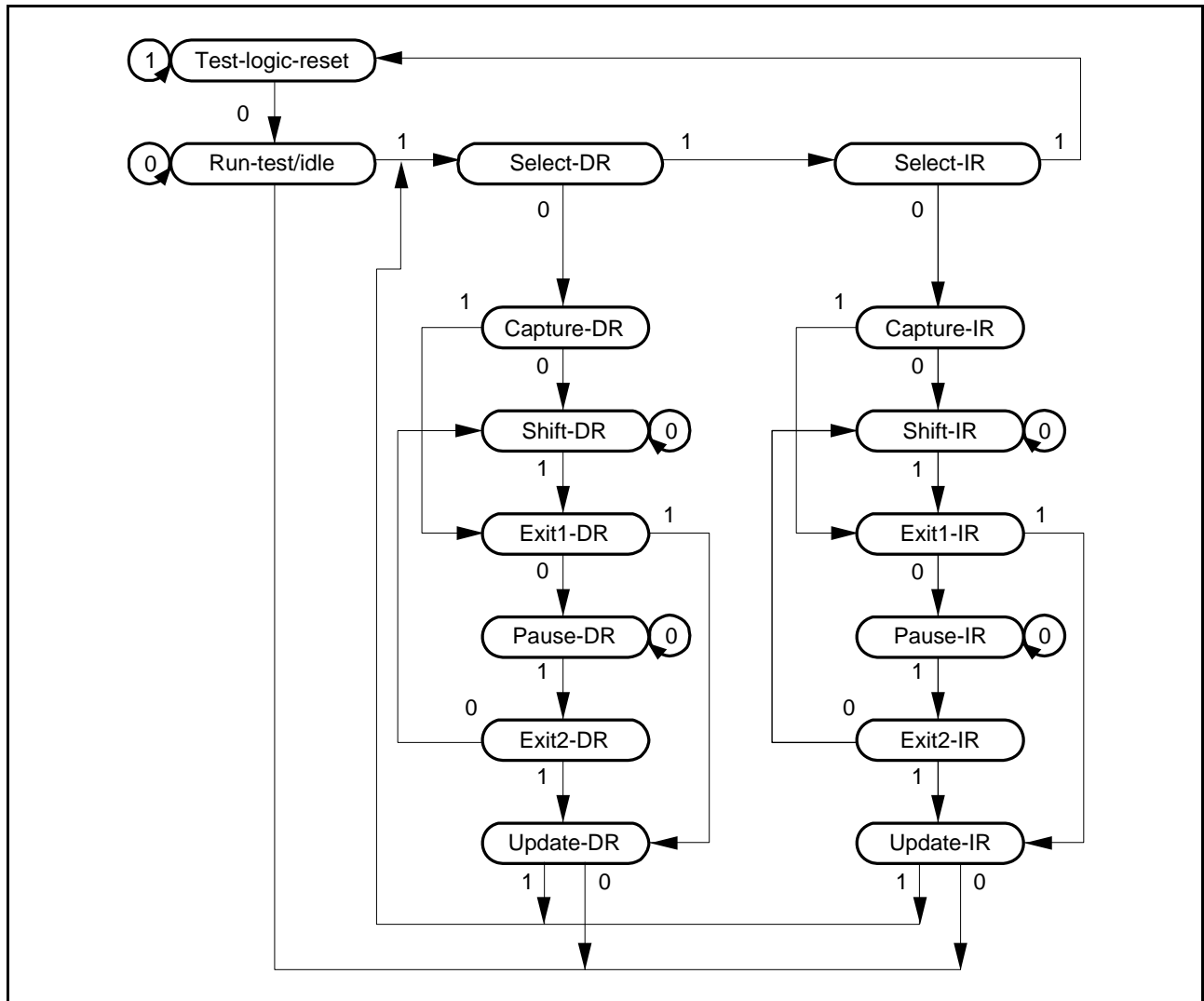


Figure 40.2 State Transition of TAP Controller

40.3.2 List of Commands

(1) BYPASS [Instruction Code: 1111 1111b]

The BYPASS instruction is an instruction that drives the bypass register (JTBPR). This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed-circuit board at higher speeds. While this instruction is being executed, the test circuit has no effect on the system circuits.

The bypass register (JTBPR) is connected between the TDI and TDO pins. Bypass operation is initiated from shift-DR operation. The TDO is low in the first clock cycle in the shift-DR state; in the subsequent clock cycles, the TDI signal is output on the TDO pin.

(2) EXTEST [Instruction Code: 0000 0000b]

The EXTEST instruction is used to test external circuits when this LSI is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified by the SAMPLE/PRELOAD instruction) from the boundary scan register to the print circuit board, and input pins are used to input test result.

(3) SAMPLE/PRELOAD [Instruction Code: 0100 0000b]

The SAMPLE/PRELOAD instruction is used to input data from the LSI internal circuits to the boundary scan register, output data from scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to the LSI and output signals are also directly output to the external circuits. The LSI system circuit is not affected by this instruction.

In SAMPLE operation, the boundary scan register latches the snap shot of data transferred from input pins to internal circuit or data transferred from internal circuit to output pins. The latched data is read from the scan path. The scan register latches the snap data at the rising edge of the TCK in Capture-DR state. The scan register latches snap shot without affecting the LSI normal operation.

In PRELOAD operation, initial value is written from the scan path to the parallel output latch of the boundary scan register prior to the EXTEST instruction execution. If the EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In EXTEST instruction, output parallel latches are always output to the output pins.)

(4) IDCODE [Instruction Code: 0101 0101b]

When the IDCODE instruction is selected, IDCODE register value is output to the TDO in Shift-DR state of the TAP controller. In this case, IDCODE register value is output from the LSB. During this instruction execution, test circuit does not affect the system circuit. INSTR is initialized by the IDCODE instruction in Test-Logic-Reset state of the TAP controller.

(5) CLAMP [Instruction Code: 1101 0000b]

When the CLAMP instruction is selected, output pins output the boundary scan register value which was specified by the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of boundary scan register is maintained regardless of the TAP controller state.

BYPASS is connected between TDI and TDO, the same operation as BYPASS instruction can be achieved.

(6) HIGHZ [Instruction Code: 1000 0000b]

When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the status of boundary scan register is maintained regardless of the state of the TAP controller.

BYPASS is connected between TDI and TDO pins, leading to the same operation as when the BYPASS instruction has been selected.

40.4 Usage Notes

1. In serial transfer, data are input or output in LSB order (see Figure 40.3).

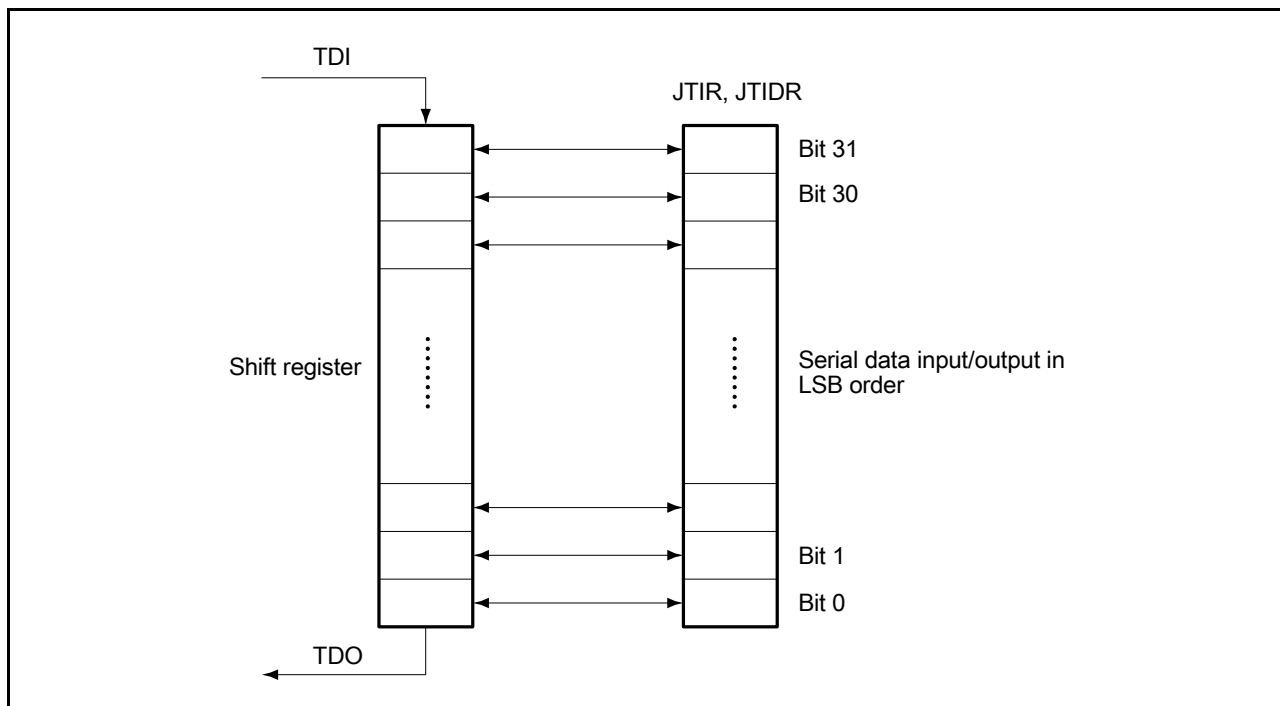


Figure 40.3 Serial Data Input/Output

2. Pins of the boundary scan (TCK, TDI, TMS, and TRST#) have to be pulled up by pull-up resistors. However, handle the #TRST pin in the way described in the manual for the given on-chip emulator if an on-chip emulator is in use.
If the #TRST pin is pulled down but a boundary scan is to proceed, ensure that the #TRST pin is also controllable.
3. Power supply pins (VCC, VCL, VSS, AVCC, AVSS, VREFH, VREFL, PLLVCC, PLLVSS, VCC_USB, and VSS_USB) cannot be boundary-scanned.
4. Clock pins (EXTAL, XTAL, XCIN, and XCOU) cannot be boundary-scanned.
5. Reset signal (RES#) cannot be boundary-scanned.
6. USB dedicated pins (USB0_DP, USB0_DM, USB1_DP, and USB1_DM) cannot be boundary-scanned.
7. The on-chip emulator enable pin (EMLE) cannot be boundary-scanned.
8. The boundary-scan pin (BSCANP) cannot be boundary-scanned.
9. The CNVSS pin cannot be boundary-scanned.
10. The boundary-scan pins (TCK, TMS, TRST#, TDI, and TDO) cannot be boundary-scanned.
11. The boundary-scan facility is not available when the chip is in the states below.
 - Reset state
 - Software standby or deep software standby
12. For a pin that incorporates open-drain functionality and for which the open-drain function is enabled, if the boundary-scan function sets the corresponding bit in the output scan register and output enable register to 1, executing an EXTTEST, CLAMP, or SAMPLE/PRELOAD instruction makes the pin output the high level rather than placing it in the high-impedance state.

13. Figure 40.4 (1) shows the pin configuration of pins P12, P13, P20, and P21. When the boundary scan function is used with pins P12, P13, P20, and P21 to be used as RIIC pins (SDA0, SDA1, SCL0, or SCL1), the conflict with open-drain output or sneak current might be generated.
14. Figure 40.4 (2) shows the pin configuration of pins P40 to P47. When the boundary scan function is used with pins P40 to P47 to be used as AD input pins (AN0 to AN7), the conflict with the AD input or sneak current might be generated.
15. Figure 40.4 (3) shows the pin configuration of pins P03 and P05. When the boundary scan function is used with pins P03 and P05 to be used as DA output pins (DA0 and DA1), the conflict with the DA output or sneak current might be generated.

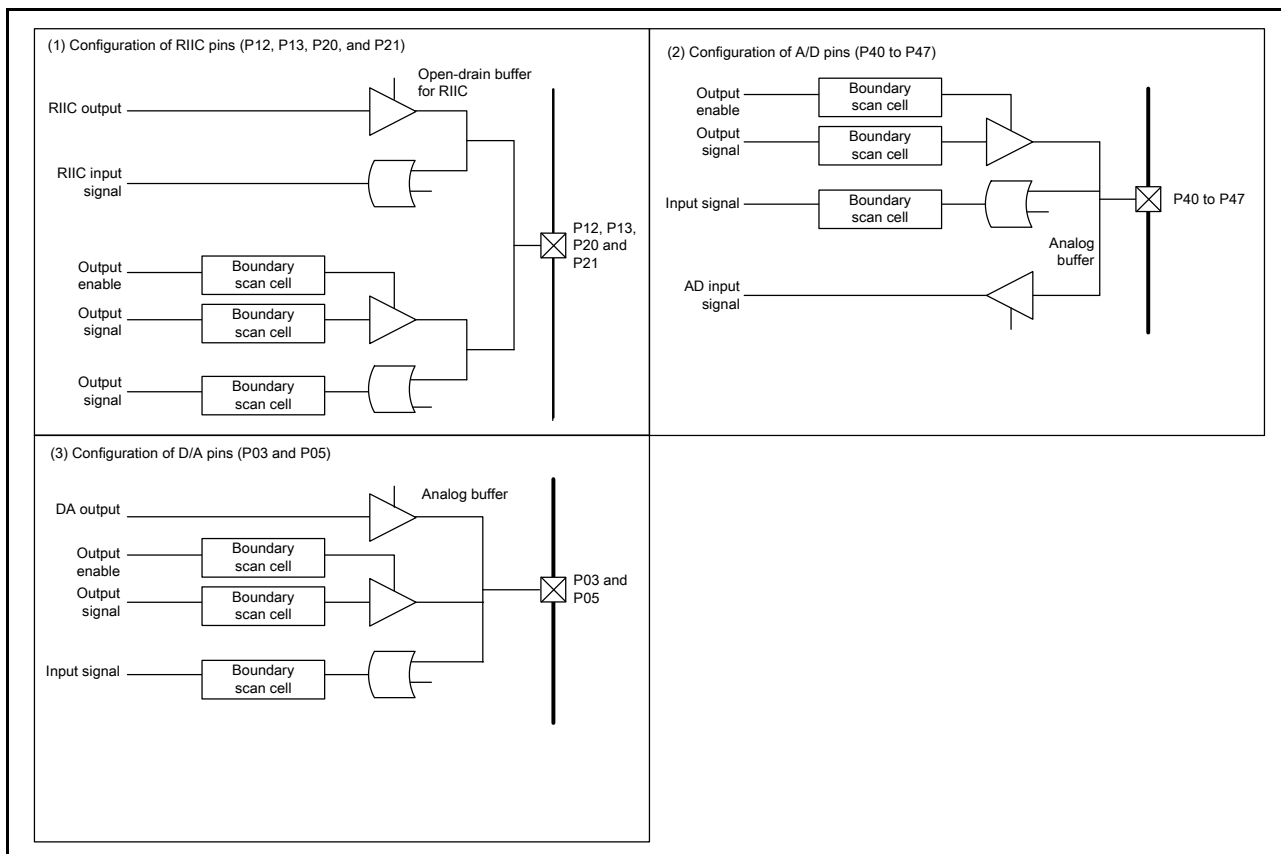


Figure 40.4 Pin Configuration

41. Electrical Characteristics

41.1 Absolute Maximum Ratings

Table 41.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	VCC PLLVCC VCC_USB	-0.3 to +4.6	V
Input voltage (except for ports 00 to 02, 07, ports 12, 13, 16, 17, ports 20, 21, port 33)	V _{IN}	-0.3 to VCC+0.3	V
Input voltage (ports 00 to 02, 07, ports 12, 13, 16, 17, ports 20, 21, port 33 ^{*1})	V _{IN}	-0.3 to +5.8	V
Reference power supply voltage	V _{REF}	-0.3 to VCC+0.3	V
Analog power supply voltage	AVCC ^{*2}	-0.3 to +4.6	V
Analog input voltage	V _{AN}	-0.3 to VCC+0.3	V
Operating temperature	T _{opr}	-40 to +85	°C
Storage temperature	T _{stg}	-55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

Note 1. Ports 00 to 02, 07, ports 12, 13, 16, 17, ports 20, 21, and port 33 are 5 V tolerant.

Note 2. Connect AVCC to VCC. When neither the A/D converter nor the D/A converter is in use, do not leave the AVCC, VREFH, AVSS, and VREFL pins open. Connect the AVCC and VREFH pins to VCC, and the AVSS and VREFL pins to VSS, respectively.

41.2 DC Characteristics

Table 41.2 DC Characteristics (1)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

T_a = -40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Schmitt trigger input voltage	IRQ input pin*1 MTU input pin*1 TMR input pin*1 SCI input pin*1 ADTRG input pin*1 RES#, NMI	V _{IH}	VCC × 0.8	—	VCC+0.3	V	
		V _{IL}	-0.3	—	VCC × 0.2		
		ΔV _T	VCC × 0.06	—	—		
	RIIC input pin (except for SMBus)	V _{IH}	VCC × 0.7	—	5.8		
		V _{IL}	-0.3	—	VCC × 0.3		
		ΔV _T	VCC × 0.05	—	—		
	Ports 00 to 02, 07 ports 12, 13, 16, 17 ports 20, 21 port 33	V _{IH}	VCC × 0.8	—	5.8		
		V _{IL}	-0.3	—	VCC × 0.2		
	Ports 03, 05, 10, 11, 14, 15 ports 22 to 27 ports 30 to 32, 34, 35 ports 4 to G Other input pins	V _{IH}	VCC × 0.8	—	VCC+0.3		
		V _{IL}	-0.3	—	VCC × 0.2		
	Input high voltage (except Schmitt trigger input pin)	MD pin, EMLE EXTAL, RSPI, ETHERC EXDMAC, WAIT#, TCK XCIN D0 to D31 RIIC (SMBus)	V _{IH}	VCC × 0.9	—		VCC+0.3
				VCC × 0.8	—		VCC+0.3
VCC × 0.8				—	VCC+0.3		
VCC × 0.7				—	VCC+0.3		
2.1				—	VCC+0.3		
Input low voltage (except Schmitt trigger input pin)	MD pin, EMLE EXTAL, RSPI, ETHERC EXDMAC, WAIT#, TCK XCIN D0 to D31 RIIC (SMBus)	V _{IL}	-0.3	—	VCC × 0.1		
			-0.3	—	VCC × 0.2		
			-0.3	—	VCC×0.2		
			-0.3	—	VCC×0.3		
			-0.3	—	0.8		

Table 41.3 DC Characteristics (2)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

T_a = -40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Output high voltage	All output pins	V _{OH}	VCC-0.5	—	—	V	I _{OH} = -1mA
Output low voltage	All output pins (except for RIIC pins)	V _{OL}	—	—	0.5	V	I _{OL} = 1.0mA
	RIIC pins		—	—	0.4	V	I _{OL} = 3.0mA
	RIIC pins (only P12 and P13 in channel 0)	V _{OL}	—	—	0.6		I _{OL} = 6.0mA
			—	0.4	—	V	I _{OL} = 15.0mA (ICFER.FMPE = 1)
						I _{OL} = 20.0mA (ICFER.FMPE = 1)	
Input leakage current	RES#, MD pin, EMLE, NMI	I _{in}	—	—	1.0	μA	V _{in} = 0V V _{in} = VCC
Three-state leakage current (off state)	Ports 03, 05, 10, 11, 14, 15 ports 22 to 27 ports 30 to 32, 34, 35 ports 4 to G	I _{TSI}	—	—	1.0	μA	V _{in} = 0V V _{in} = VCC
	Ports 00 to 02, 07, 12, 13 Ports 16, 17, 20, 21, 33		—	—	5.0		
Input pull-up MOS current	Ports 9 to E, G	-I _p	10	—	300	μA	VCC = 2.7 to 3.6V V _{in} = 0V
Input capacitance	All input pins (except for ports 12, 13, 20, 21 ports 40 to 47, and EMLE)	C _{in}	—	—	15	pF	V _{in} = 0V f = 1MHz T _a = 25°C
	Ports 12, 13, 20, 21, Ports 40 to 47, EMLE		—	—	30		

Table 41.4 DC Characteristics (3)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

T_a = -40 to +85°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions			
Supply current ²	In operation	Max. ³	I _{CC} ⁴	—	—	100	mA	ICLK = 100MHz PCLK = 50MHz BCLK = 50MHz			
		Normal operation		Peripheral function: Clocks supplied ⁵	—	48			—		
				Peripheral function: Clocks not supplied ⁵	—	35			—		
		Increased by BGO operation ⁶		—	15	—					
	Sleep			—	20	60					
	All-module-clock-stop mode ⁷			—	14	28					
	Standby mode	Software standby mode		—	0.12	3.0	mA				
		Deep software standby mode		RTC in operation	RAM, USB retained	—			30	206	μA
						RAM, USB power supply halted			—	26	
		RTC halted		RAM, USB retained	—	25			200	μA	
RAM, USB power supply halted	—		21		60	μA					
Analog power supply current	During 12-bit A/D conversion (per unit)		AI _{CC}	—	2.5		3.0	mA			
	During 10-bit A/D conversion (per unit)			—	0.8	1.2	mA				
	During D/A conversion (per unit)			—	0.3	2.0			μA		
	Idle (all units)			—	30	35					
	During A/D or D/A standby (all units)			—	0.1	4.0					
Reference power supply current	During 12-bit A/D conversion (per unit)		AI _{CC}	—	0.5	0.7	mA				
	During 10-bit A/D conversion (per unit)			—	0.06	0.1		mA			
	During D/A conversion (per unit)			—	0.6	1.0			mA		
	Idle (all units)			—	0.4	0.6					
	During A/D or D/A standby (all units)			—	0.1	2.0		μA			
RAM standby voltage		V _{RAM}	2.48	—	—	V					
VCC rising gradient		SVCC	—	—	20	ms/V					

Note 1. This does not include the pins, which are multiplexed as ports 00 to 02, 07, ports 12, 13, 16, 17, ports 20, 21, and port 33 for 5 V tolerant.

Note 2. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Note 3. Measured with clocks supplied to the peripheral functions. This does not include the BGO operation.

Note 4. ICC depends on f (ICLK) as follows. (ICLK: PCLK: BCLK: BCLK pin = 8 : 4: 8: 4)

ICC max. = 0.89 x f + 11 (max.)

ICC typ. = 0.43 x f + 5 (normal operation, peripheral function: clocks supplied)

ICC typ. = 0.30 x f + 5 (normal operation, peripheral function: clocks not supplied)

ICC max. = 0.48 x f + 12 (sleep mode)

Note 5. This does not include the BGO operation.

Note 6. Incremented if data is written to or erased from the ROM or data flash for data storage during the program execution.

Note 7. The values are for reference.

Table 41.5 Permissible Output Currents

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

T_a = -40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (average value per pin)	All output pins except for RIIC pins	I _{OL}	—	—	2.0	mA
	RIIC pins (ICFER.FMPE = 0)	I _{OL}	—	—	6.0	mA
	RIIC pins (ICFER.FMPE = 1)	I _{OL}	—	—	20.0	mA
Permissible output low current (max. value per pin)	All output pins except for RIIC pins	I _{OL}	—	—	4.0	mA
	RIIC pins (ICFER.FMPE = 0)	I _{OL}	—	—	6.0	mA
	RIIC pins (ICFER.FMPE = 1)	I _{OL}	—	—	20.0	mA
Permissible output low current (total)	Total of all output pins	ΣI _{OL}	—	—	80	mA
Permissible output high current (average value per pin)	All output pins (except for USB_DPUPE pin)	-I _{OH}	—	—	2.0	mA
	USB_DPUPE pin	-I _{OH}	—	—	3.0	mA
Permissible output high current (max. value per pin)	All output pins	-I _{OH}	—	—	4.0	mA
Permissible output high current (total)	Total of all output pins	Σ-I _{OH}	—	—	80	mA

Caution: To protect the LSI's reliability, the output current values should not exceed the permissible output current.

41.3 AC Characteristics

Table 41.6 Operation Frequency Value [176-pin LFBGA/145-pin TFLGA/144-pin LQFP]

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

ICLK = 8 to 100MHz, PCLK = 8 to 50MHz, BCLK = 8 to 100MHz, SDCLK = 8 to 50MHz

T_a = -40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit
Operation frequency	System clock (ICLK)	f	8 ^{*1}	—	100	MHz
	Peripheral module clock (PCLK)		8 ^{*2}	—	50	
	External bus clock (BCLK)		8	—	100	
	BCLK pin output		8	—	50	
	SDRAM clock (SDCLK)		8	—	50	
	SDCLK pin output		8	—	50	

Note 1. The ICLK must run at a frequency of at least 12.5 MHz if the Ethernet controller is in use.

Note 2. The PCLK must run at a frequency of at least 24 MHz if the USB is in use.

Table 41.7 Operation Frequency Value [100-pin LQFP/85-pin TFLGA]

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

ICLK = 8 to 100MHz, PCLK = 8 to 50MHz, BCLK = 8 to 50MHz

T_a = -40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit
Operation frequency	System clock (ICLK)	f	8 ^{*1}	—	100	MHz
	Peripheral module clock (PCLK)		8 ^{*2}	—	50	
	External bus clock (BCLK)		8	—	50	
	BCLK pin output		8	—	25	

Note 1. The ICLK must run at a frequency of at least 12.5 MHz if the Ethernet controller is in use.

Note 2. The PCLK must run at a frequency of at least 24 MHz if the USB is in use.

41.3.1 Clock Timing

Table 41.8 Clock Timing

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

T_a = -40 to +85°C

Item	Symbol	Min.	Max.	Unit	Test Conditions
BCLK pin output cycle time [176-pin LFBGA/145-pin TFLGA/144-pin LQFP]	t _{Bcyc}	20	125	ns	Figure 41.1
BCLK pin output cycle time [100-pin LQFP/85-pin TFLGA]	t _{Bcyc}	40	125	ns	
BCLK pin output high pulse width	t _{CH}	5	—	ns	
BCLK pin output low pulse width	t _{CL}	5	—	ns	
BCLK pin output rising time	t _{Cr}	—	5	ns	
BCLK pin output falling time	t _{Cf}	—	5	ns	
SDCLK pin output cycle time	t _{SDcyc}	20	125	ns	
SDCLK pin output high pulse width	t _{CH}	5	—	ns	
SDCLK pin output low pulse width	t _{CL}	5	—	ns	
SDCLK pin output rising time	t _{Cr}	—	5	ns	
SDCLK pin output falling time	t _{Cf}	—	5	ns	
Oscillation settling time after reset (crystal)	t _{OSC1}	10	—	ms	Figure 41.2
Oscillation settling time after leaving software standby mode (crystal)	t _{OSC2}	10	—	ms	Figure 41.3
Oscillation settling time after leaving deep software standby mode (crystal)	t _{OSC3}	10	—	ms	Figure 41.4
EXTAL external clock output delay settling time	t _{DEXT}	1	—	ms	Figure 41.2
EXTAL external clock input low pulse width	t _{EXL}	30.71	—	ns	Figure 41.5
EXTAL external clock input high pulse width	t _{EXH}	30.71	—	ns	
EXTAL external clock rising time	t _{EXr}	—	5	ns	
EXTAL external clock falling time	t _{EXf}	—	5	ns	
XCIN sub-clock oscillation settling time	t _{SUBOSC}	2	—	s	Figure 41.6
XCIN sub-clock oscillation frequency	f _{SUB}	32.768	—	kHz	
On-chip oscillator (IWDTCCLK) oscillation frequency	f _{IWDTCCLK}	62.5	187.5	kHz	

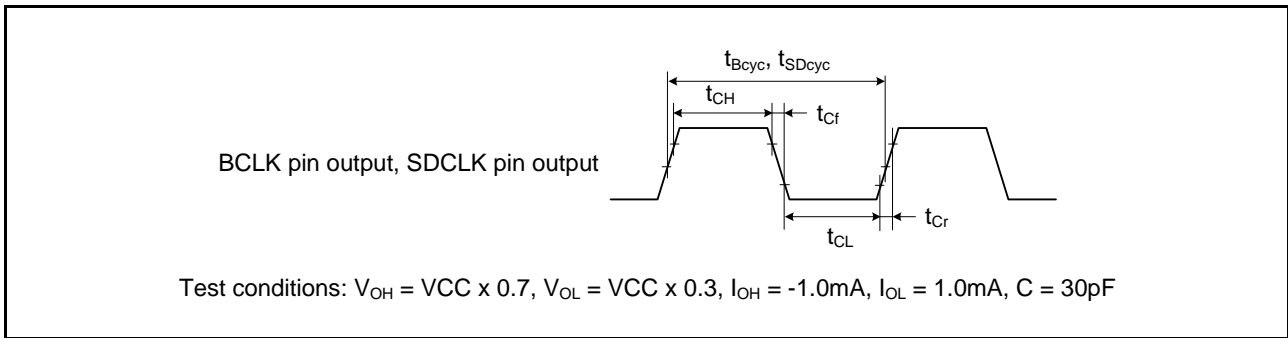


Figure 41.1 BCLK Pin Output, SDCLK Pin Output Timing

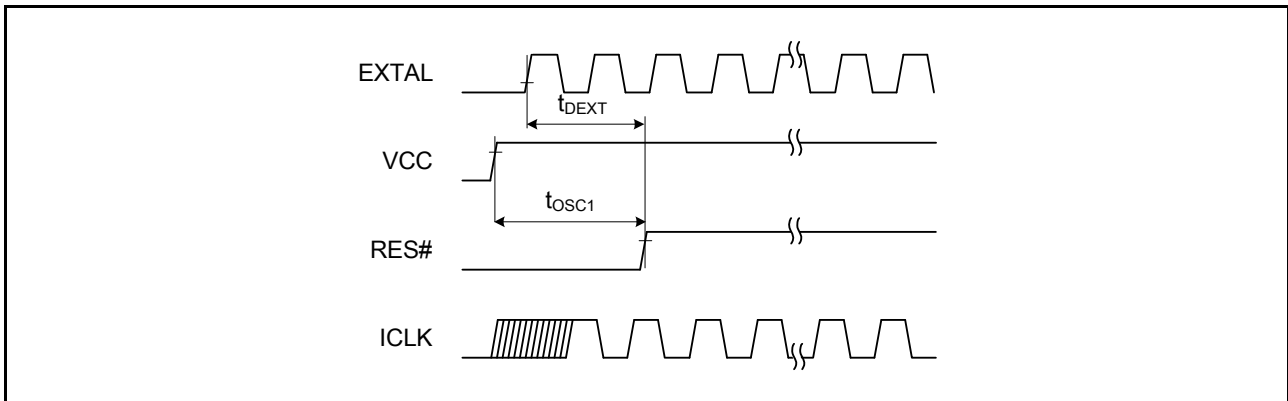


Figure 41.2 Oscillation Settling Timing

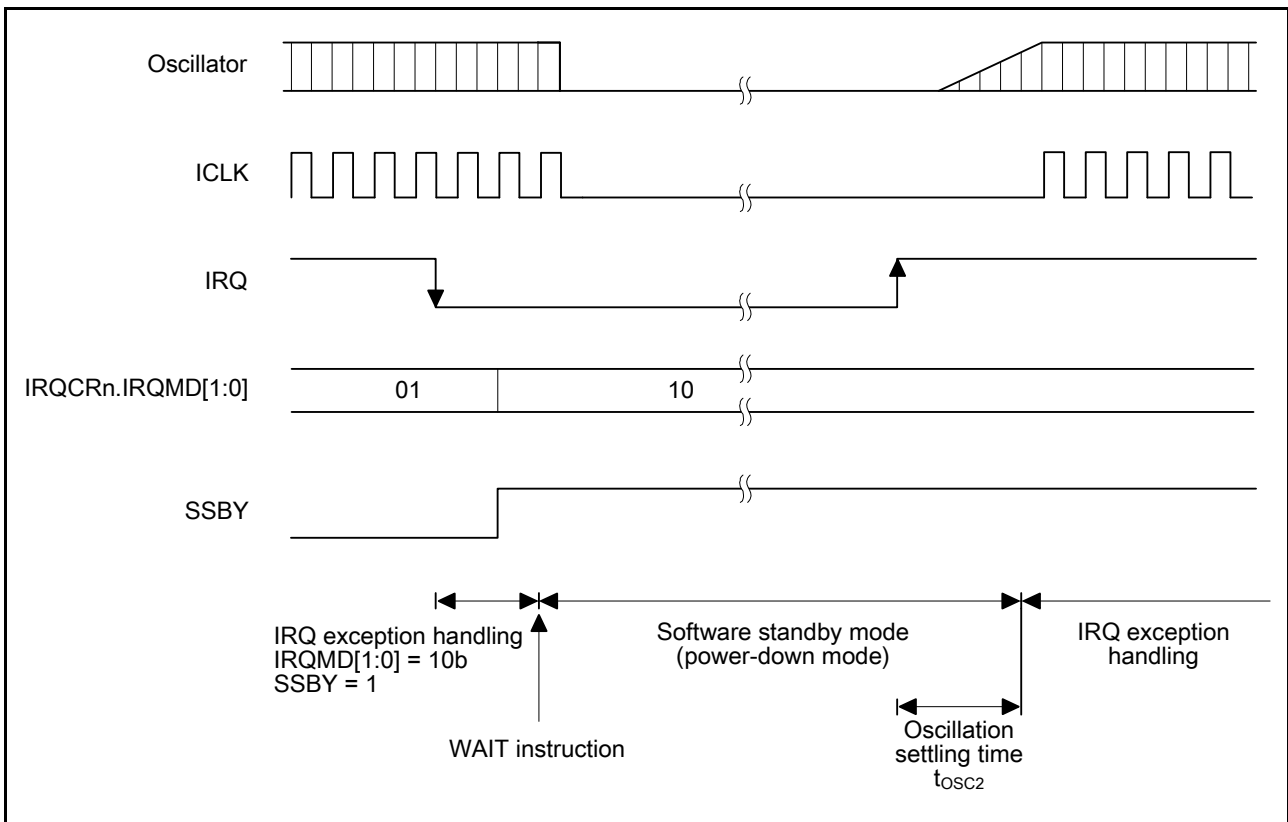


Figure 41.3 Oscillation Settling Timing after Software Standby Mode

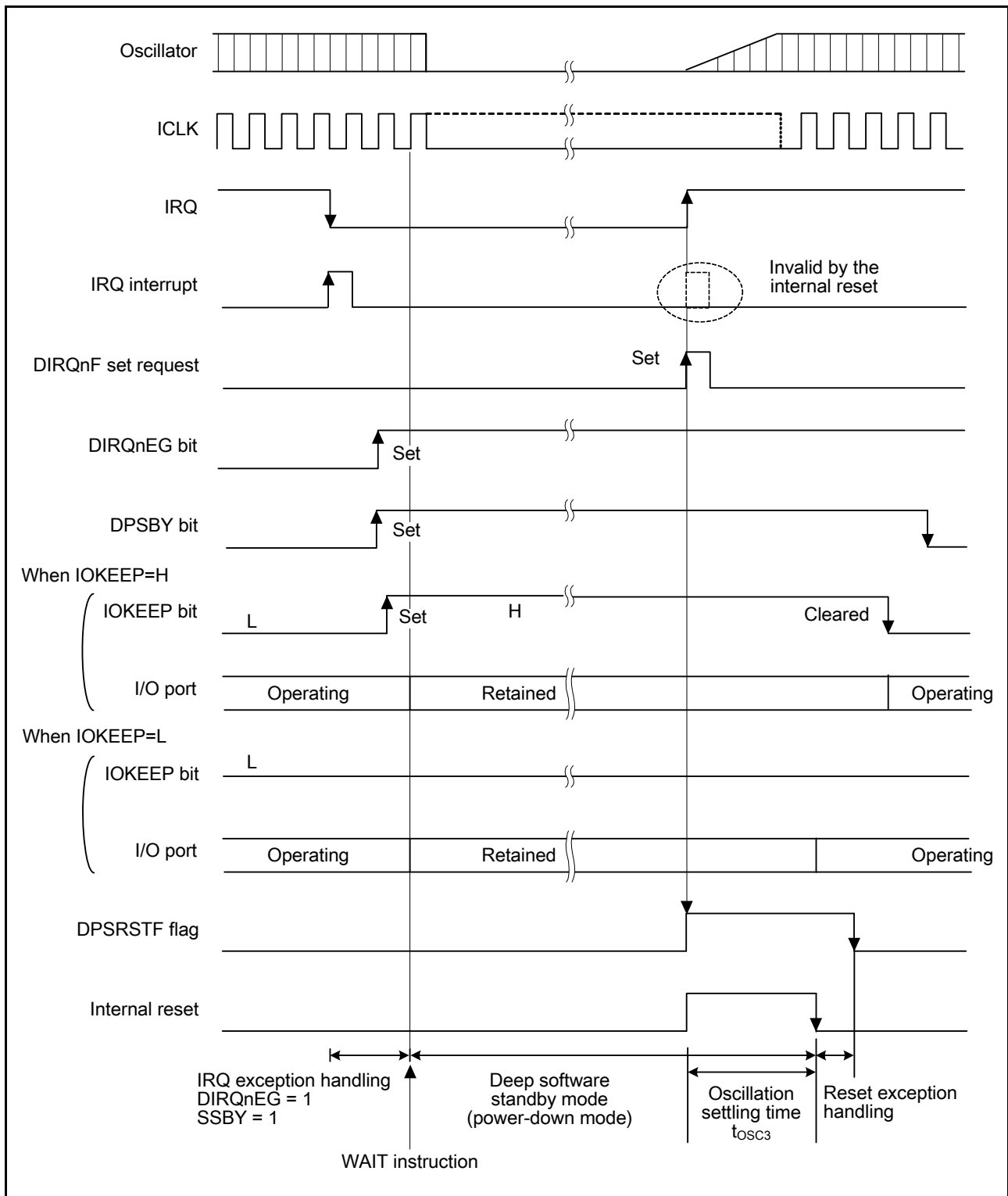


Figure 41.4 Oscillation Settling Timing after Deep Software Standby Mode

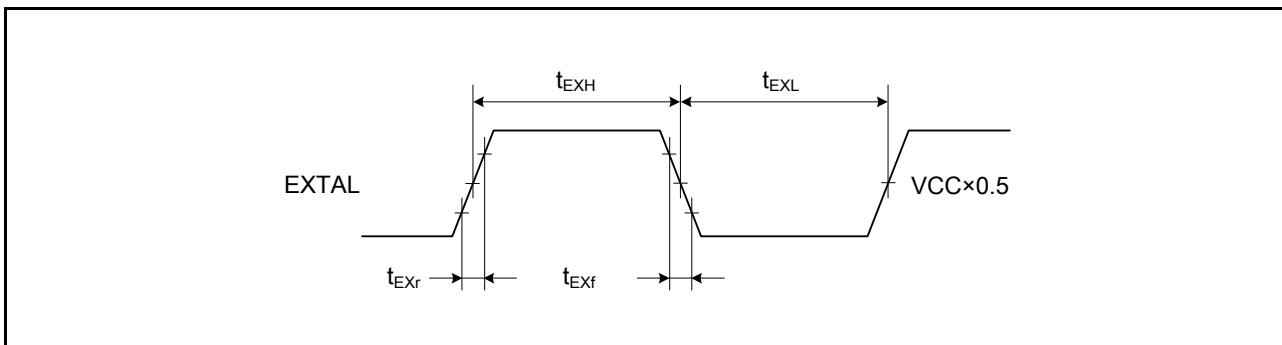


Figure 41.5 EXTAL External Input Clock Timing

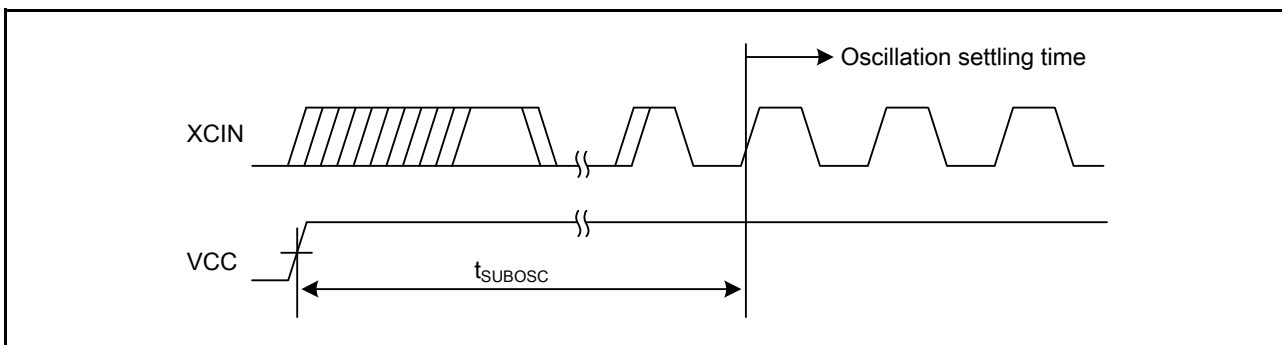


Figure 41.6 XCIN Sub-Clock Oscillation Settling Time

41.3.2 Control Signal Timing

Table 41.9 Control Signal Timing

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC
 VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V
 T_a = -40 to +85°C

Item	Symbol	Min.	Max.	Unit	Test Conditions
RES# pulse width (except for programming or erasure of the ROM or data-flash memory or blank checking of the data-flash memory*1)	t _{RESW} *2	20	—	t _{ICLK} *4	Figure 41.7
		1.5	—	μs	
Internal reset time*3	t _{RESW2}	35	—	μs	
NMI pulse width	t _{NMIW}	200	—	ns	Figure 41.8
IRQ pulse width	t _{IRQW}	200	—	ns	Figure 41.9

Note 1. For a reset by the signal on the RES# pin during programming or erasure of the ROM or data-flash memory or during blank checking of the data-flash memory, see section 38.13, Usage Notes, in section 38, ROM (Flash Memory for Code Storage).

Note 2. Both the time and the number of cycles should satisfy the specifications.

Note 3. This is to specify the FCU reset.

Note 4. t_{ICLK}: ICLK cycles

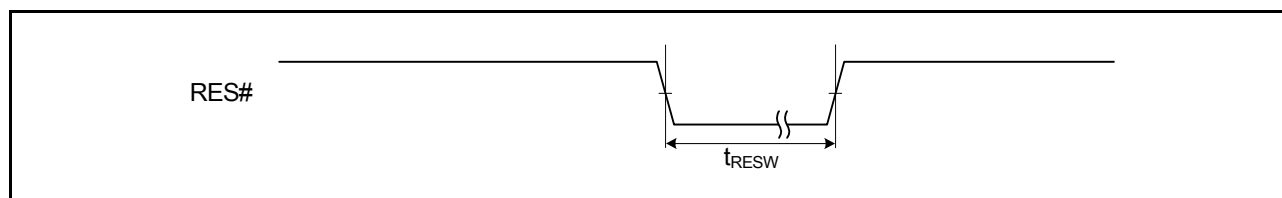


Figure 41.7 Reset Input Timing

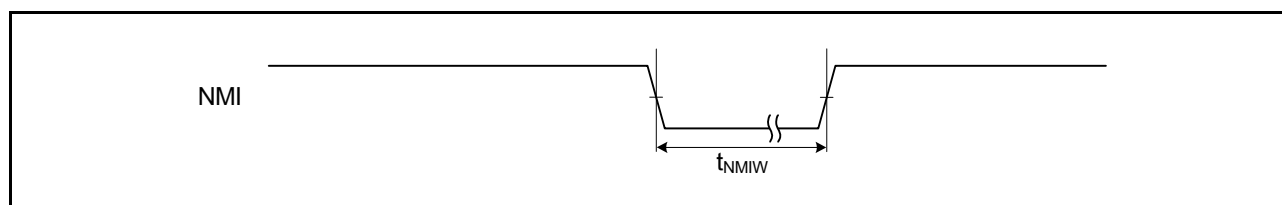


Figure 41.8 NMI Interrupt Input Timing

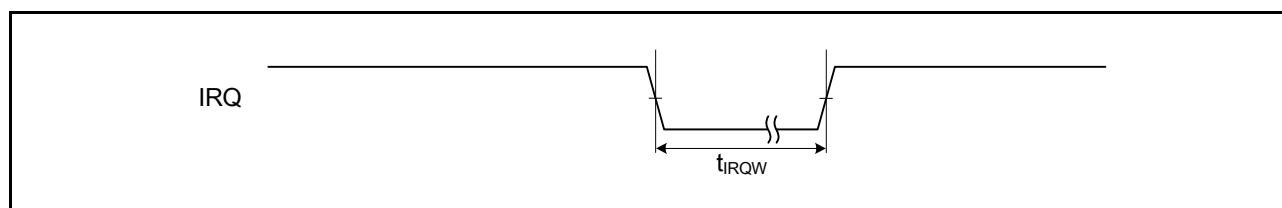


Figure 41.9 IRQ Interrupt Input Timing

41.3.3 Bus Timing

Table 41.10 Bus Timing [176-pin LFBGA/145-pin TFLGA/144-pin LQFP]

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

ICLK = 8 to 100MHz, BCLK = 8 to 100MHz, SDCLK = 8 to 50MHz

T_a = -40 to +85°COutput load conditions: V_{OH} = VCC×0.5, V_{OL} = VCC×0.5, I_{OH} = -1.0mA, I_{OL} = 1.0mA, C = 30pF

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t _{AD}	—	15	ns	Figure 41.10 to Figure 41.13
Byte control delay time	t _{BCD}	—	15	ns	
CS# delay time	t _{CSD}	—	15	ns	
RD# delay time	t _{RSD}	—	15	ns	
Read data setup time	t _{RDS}	15	—	ns	
Read data hold time	t _{RDH}	0.0	—	ns	
WR# delay time	t _{WRD}	—	15	ns	
Write data delay time	t _{WDD}	—	15	ns	
Write data hold time	t _{WDH}	0	—	ns	
WAIT# setup time	t _{WTS}	15	—	ns	
WAIT# hold time	t _{WTH}	0.0	—	ns	
Address delay time 2 (SDRAM)	t _{AD2}	1	15	ns	Figure 41.22 to Figure 41.28
CS# delay time 2 (SDRAM)	t _{CSD2}	1	15	ns	
DQM delay time (SDRAM)	t _{DQMD}	1	15	ns	
CKE delay time (SDRAM)	t _{CKED}	1	15	ns	
Read data setup time 2 (SDRAM)	t _{RDS2}	12	—	ns	
Read data hold time 2 (SDRAM)	t _{RDH2}	0	—	ns	
Write data delay time 2 (SDRAM)	t _{WDD2}	—	15	ns	
Write data hold time 2 (SDRAM)	t _{WDH2}	1	—	ns	
WE# delay time (SDRAM)	t _{WED}	1	15	ns	
RAS# delay time (SDRAM)	t _{RASD}	1	15	ns	
CAS# delay time (SDRAM)	t _{CASD}	1	15	ns	

Table 41.11 Bus Timing [100-pin LQFP/85-pin TFLGA]

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

ICLK = 8 to 100MHz, PCLK = 8 to 50MHz, BCLK = 8 to 50MHz

T_a = -40 to +85°COutput load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, I_{OH} = -1.0mA, I_{OL} = 1.0mA, C = 30pF

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t _{AD}	—	30	ns	Figure 41.10 to Figure 41.13
Byte control delay time	t _{BCD}	—	30	ns	
CS# delay time	t _{CSD}	—	30	ns	
RD# delay time	t _{RSD}	—	30	ns	
Read data setup time	t _{RDS}	15	—	ns	
Read data hold time	t _{RDH}	0.0	—	ns	
WR# delay time	t _{WRD}	—	30	ns	
Write data delay time	t _{WDD}	—	35	ns	
Write data hold time	t _{WDH}	0	—	ns	
WAIT# setup time	t _{WTS}	15	—	ns	Figure 41.14
WAIT# hold time	t _{WTH}	0.0	—	ns	

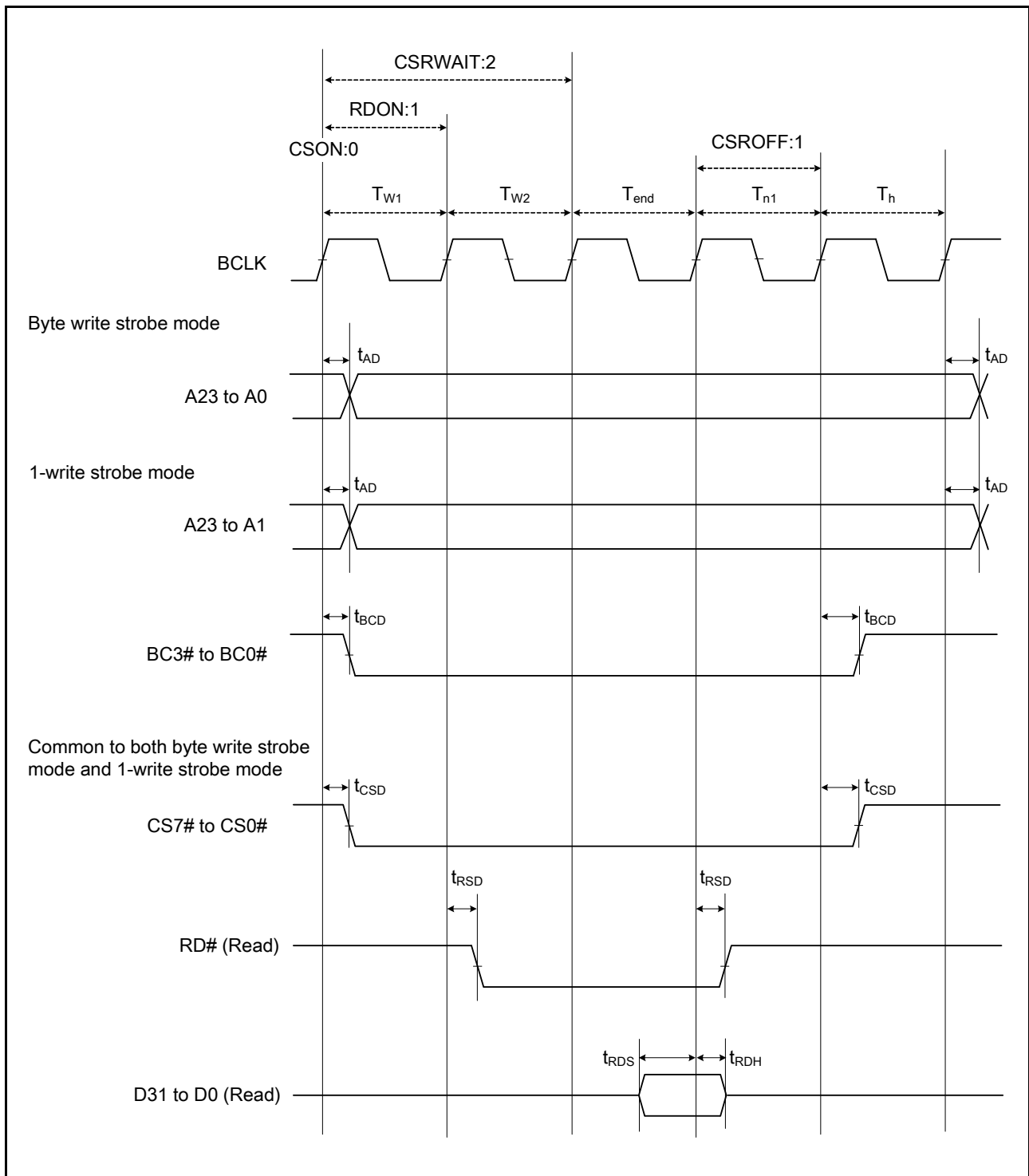


Figure 41.10 External Bus Timing/Normal Read Cycle (Bus Clock Synchronized)

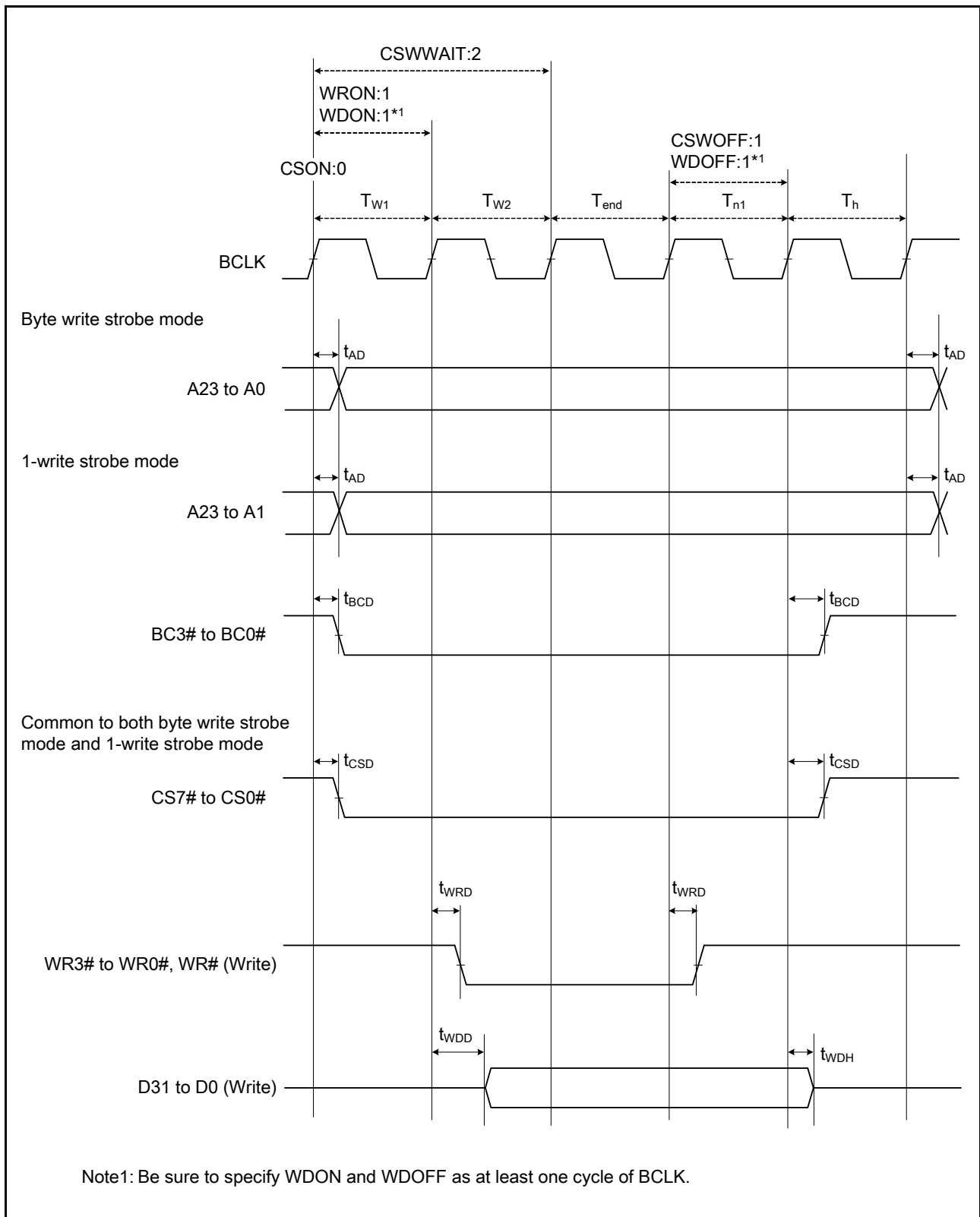


Figure 41.11 External Bus Timing/Normal Write Cycle (Bus Clock Synchronized)

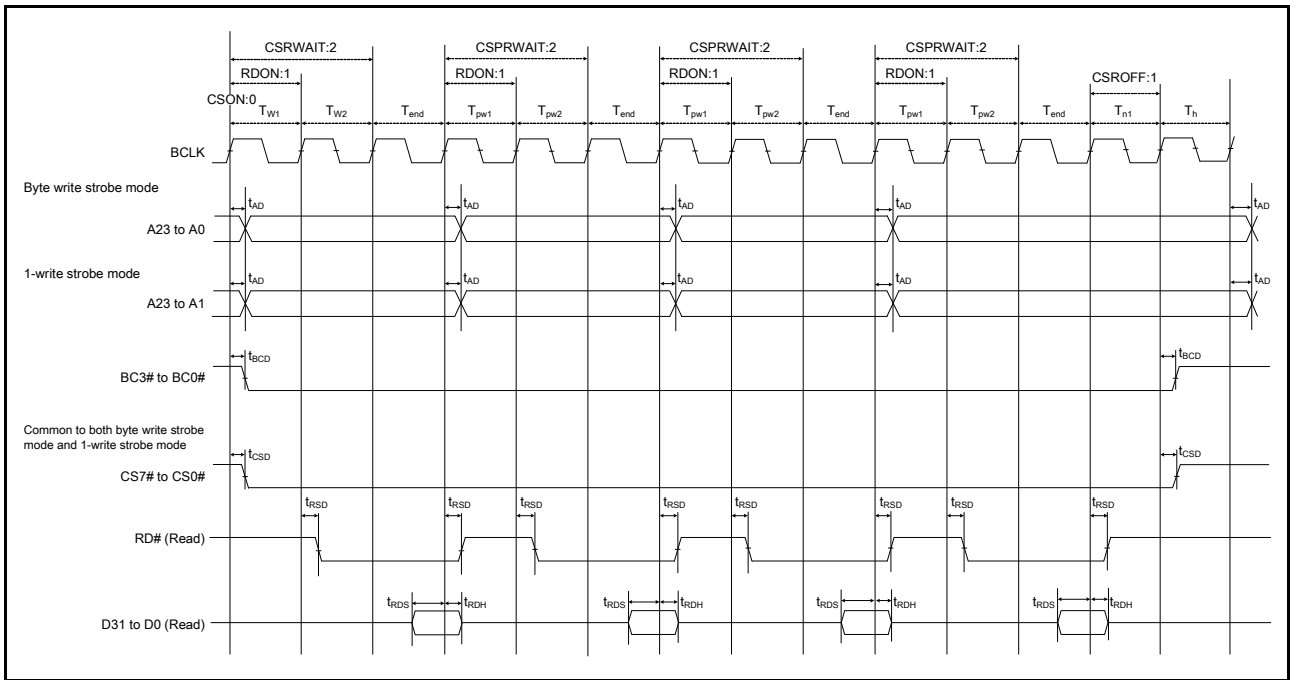


Figure 41.12 External Bus Timing/Page Read Cycle (Bus Clock Synchronized)

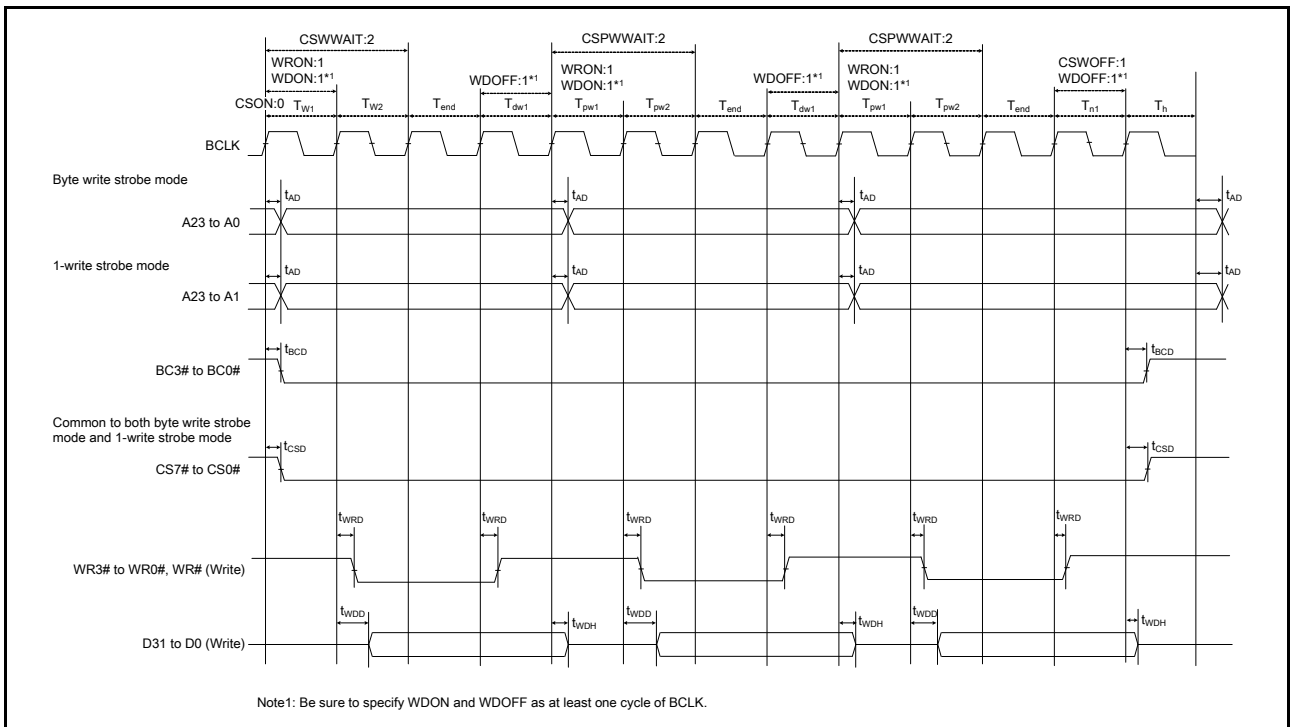


Figure 41.13 External Bus Timing/Page Write Cycle (Bus Clock Synchronized)

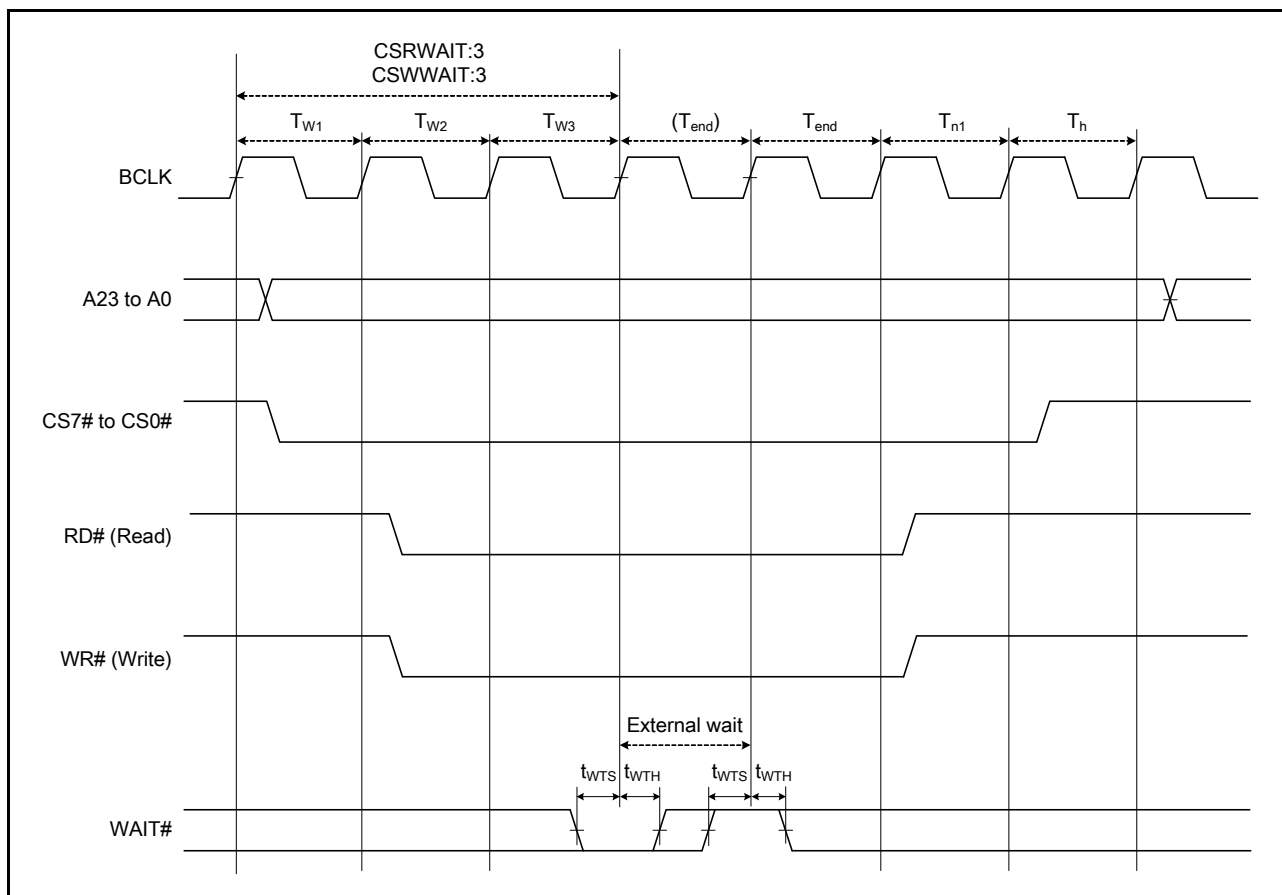


Figure 41.14 External Bus Timing/External Wait Control

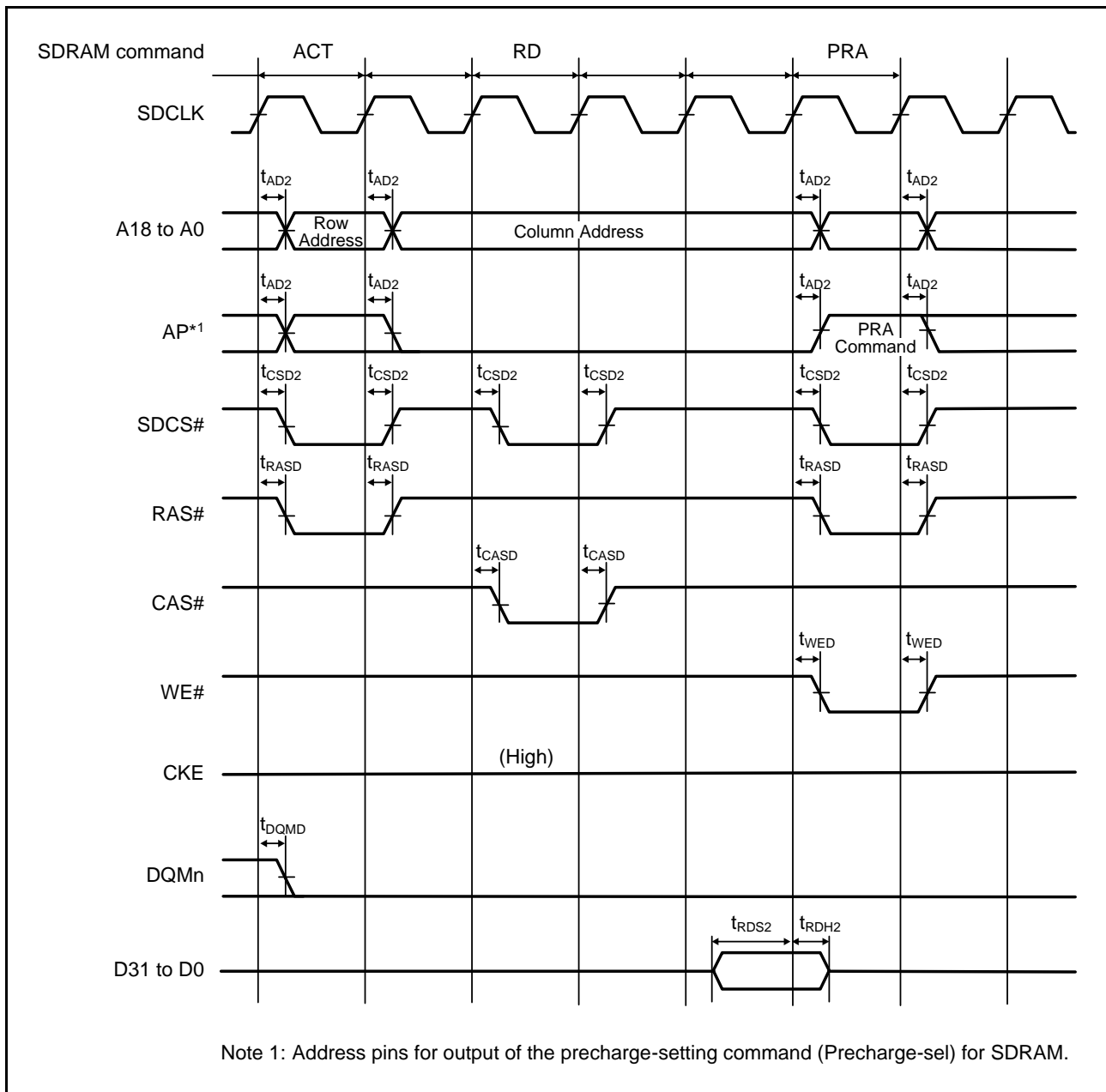


Figure 41.15 SDRAM Space Single Read Bus Timing

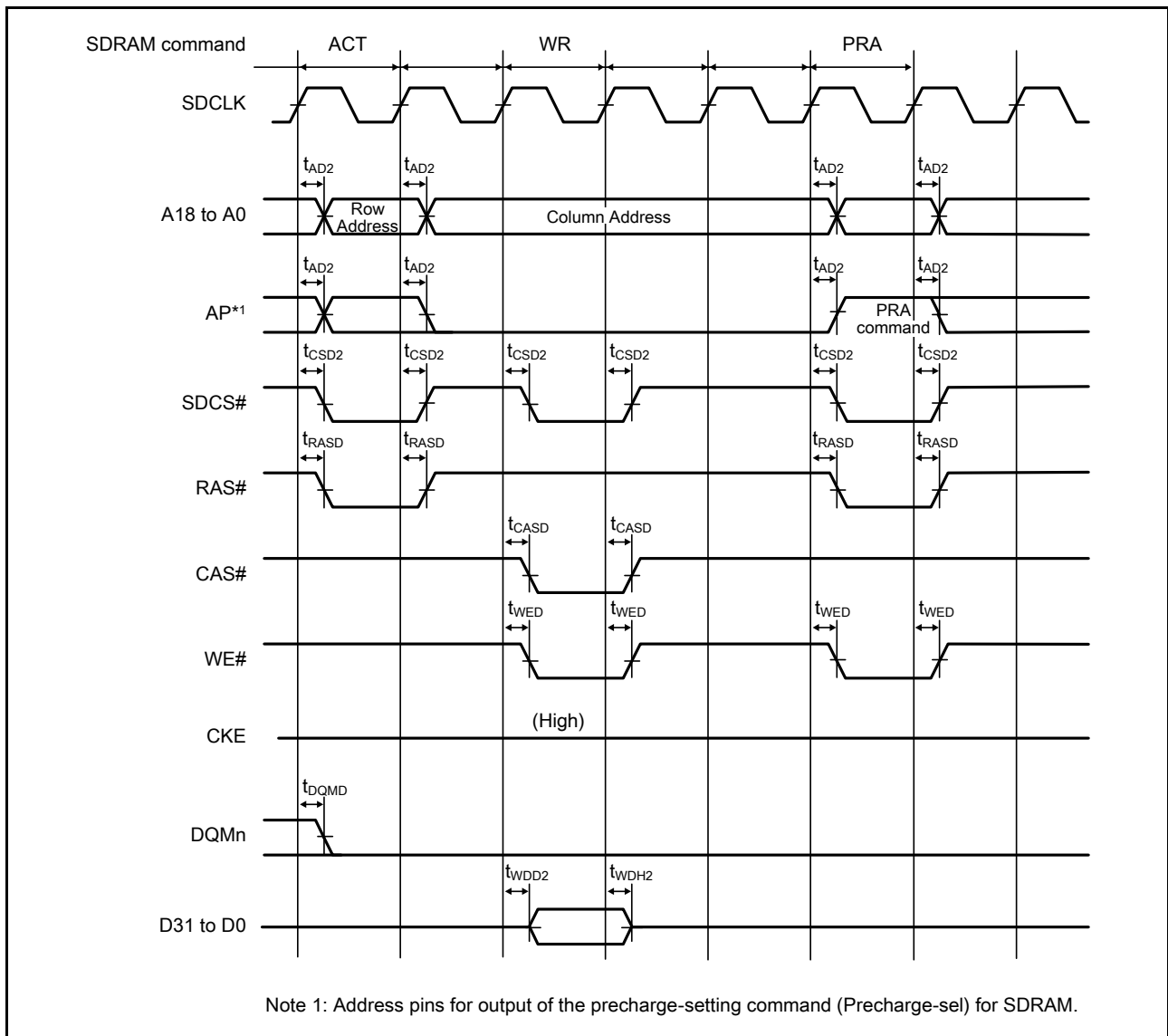


Figure 41.16 SDRAM Space Single Write Bus Timing

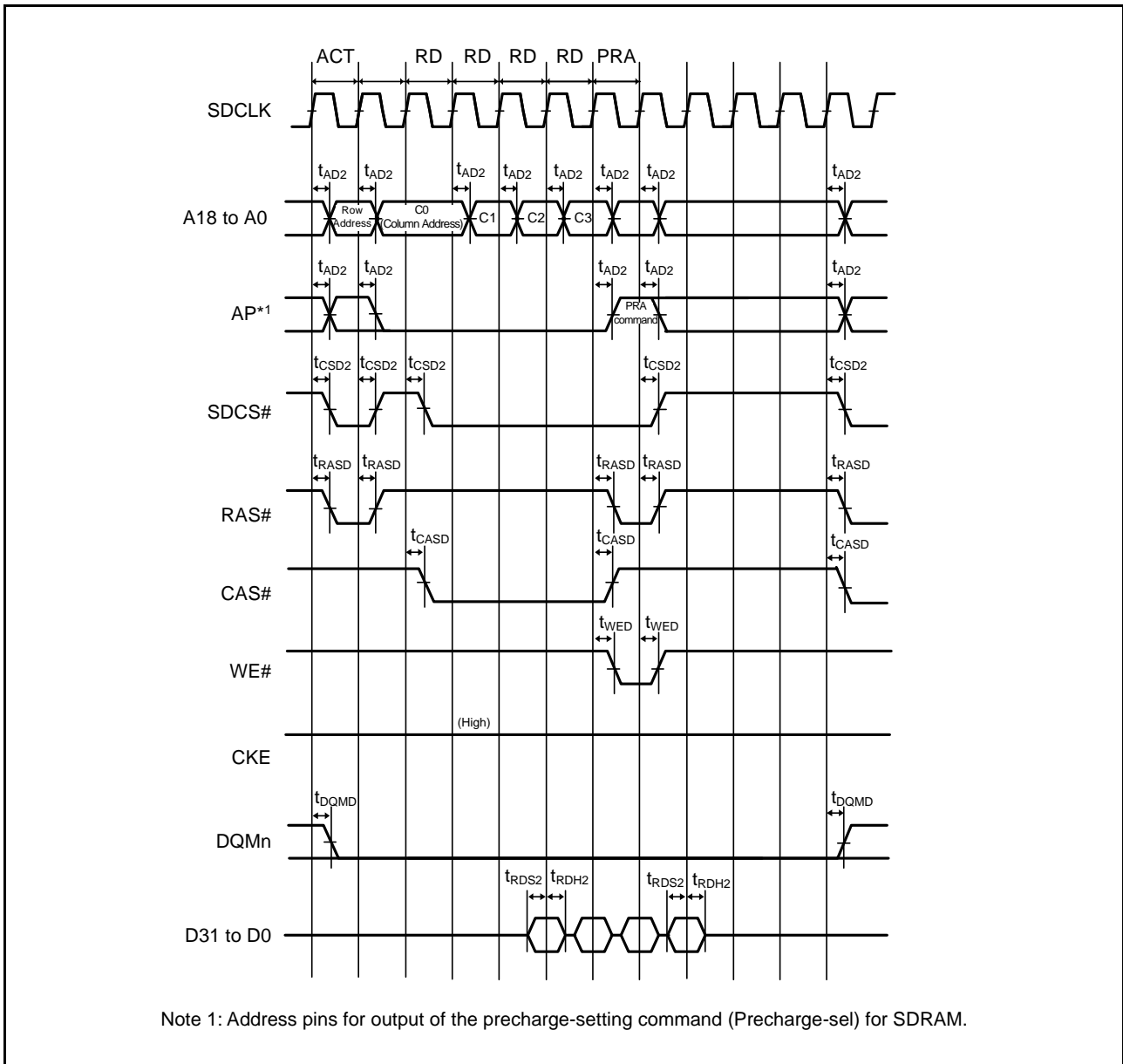


Figure 41.17 SDRAM Space Multiple Read Bus Timing

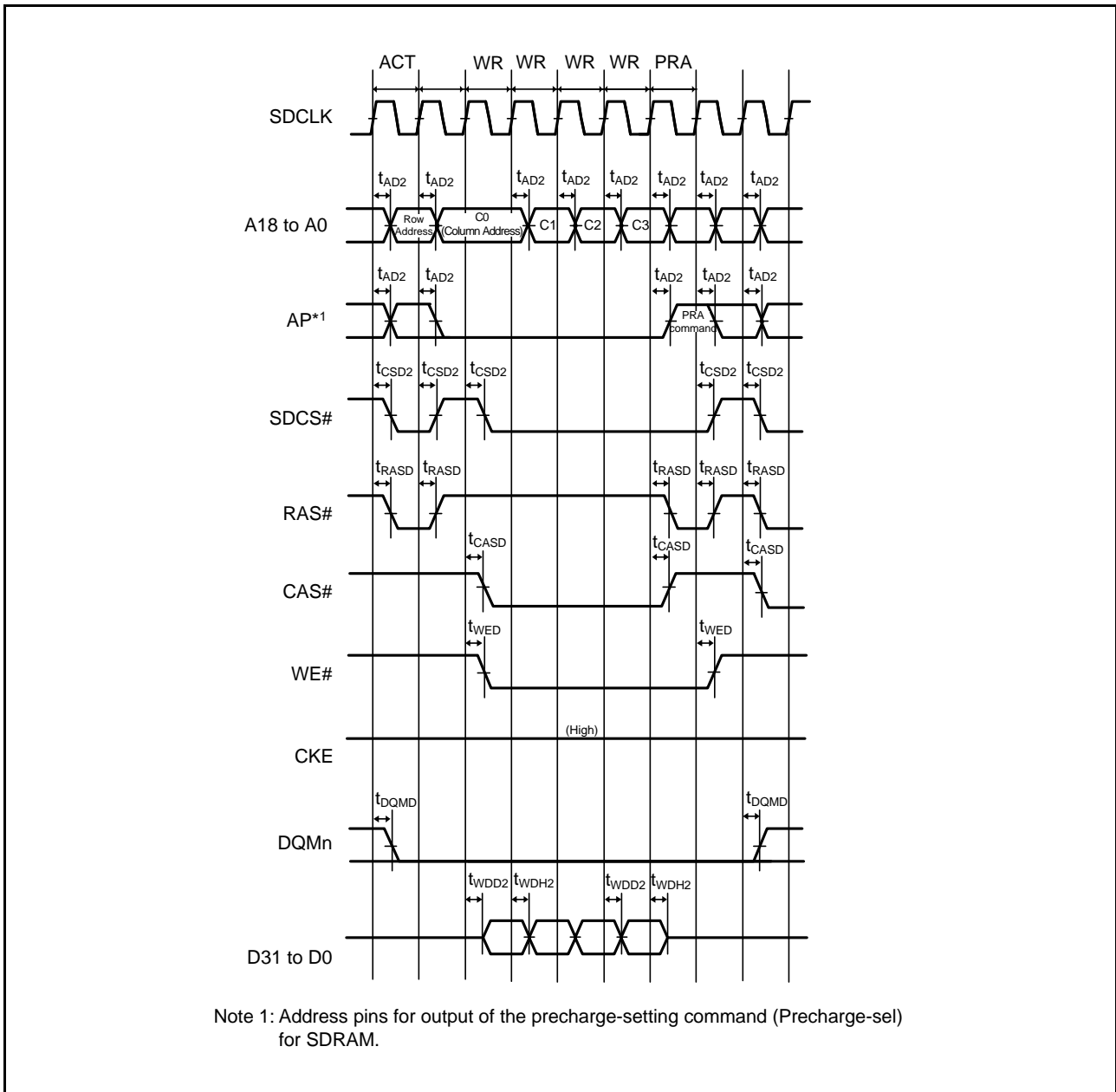


Figure 41.18 SDRAM Space Multiple Write Bus Timing

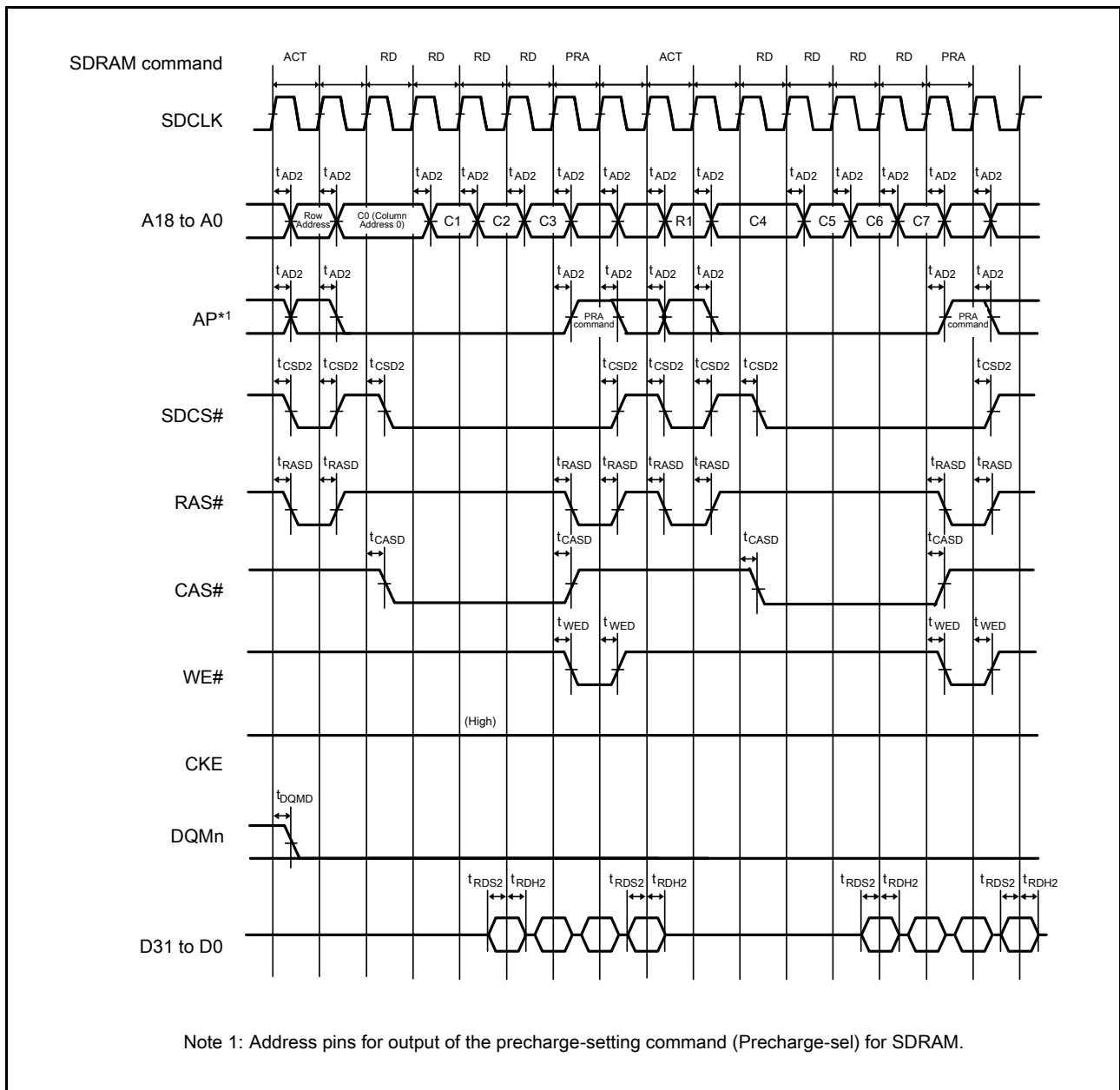


Figure 41.19 SDRAM Space Multiple Read Line Stride Bus Timing

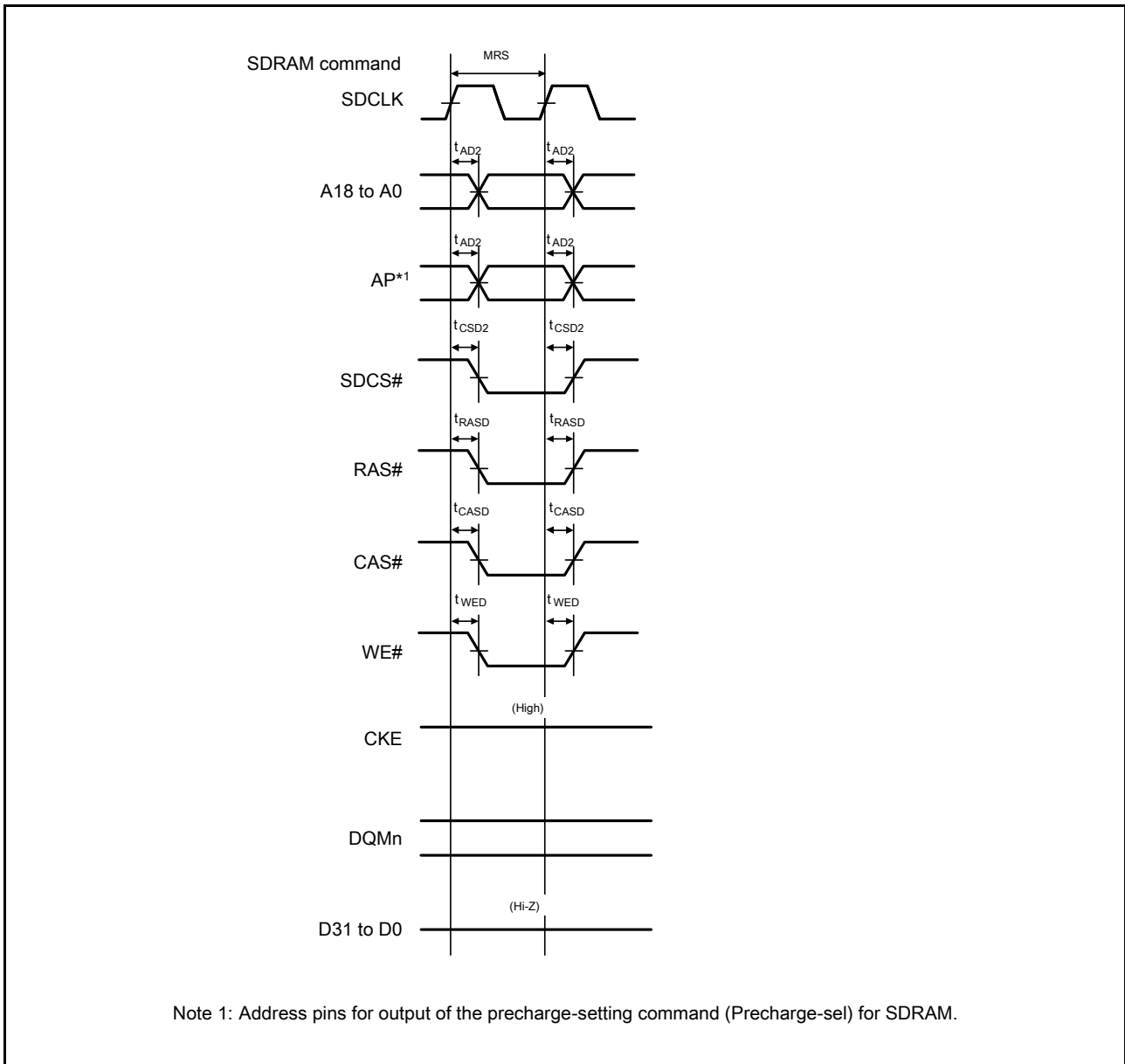


Figure 41.20 SDRAM Space Mode Register Set Bus Timing

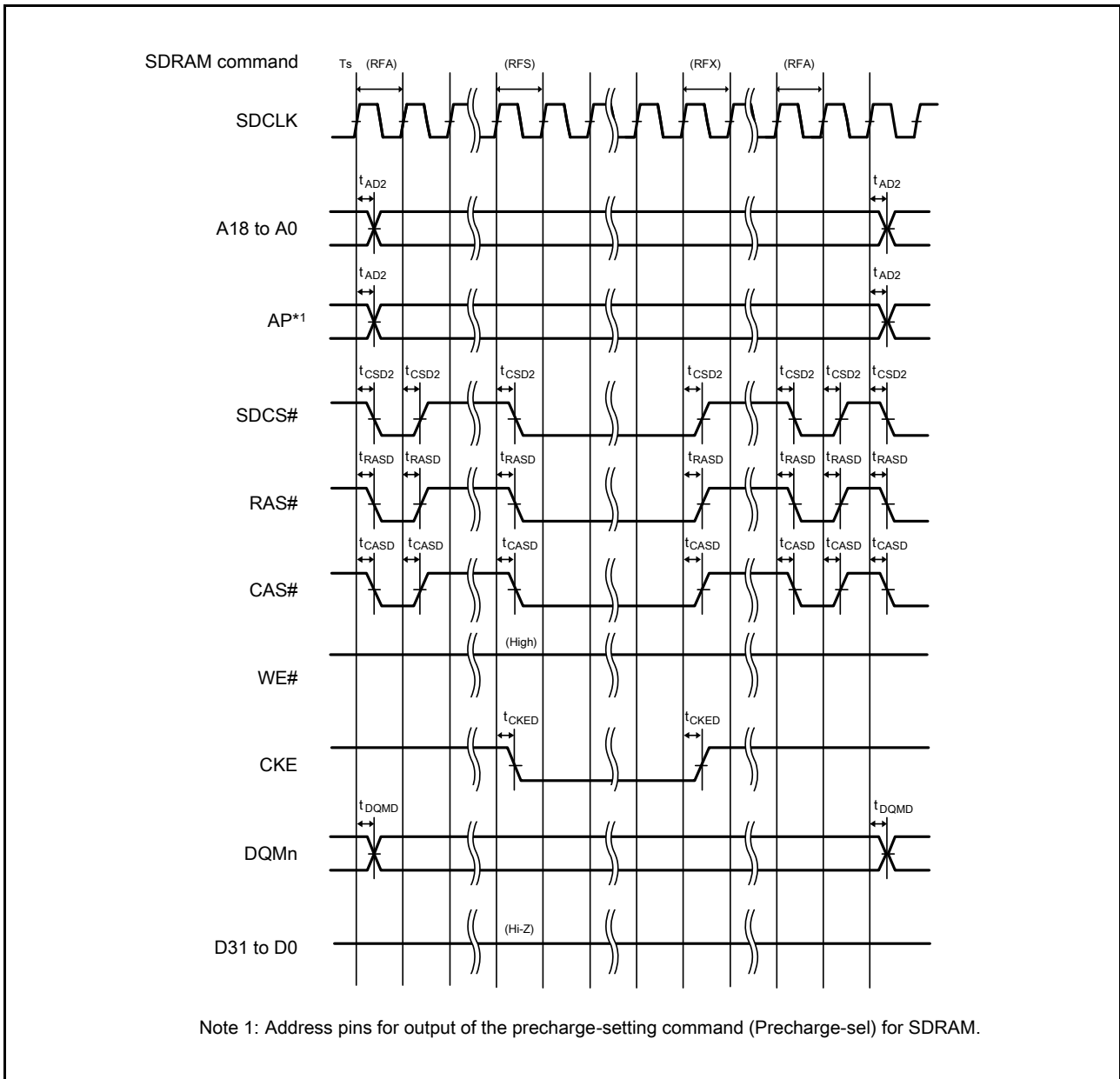


Figure 41.21 SDRAM Space Self-Refresh Bus Timing

41.3.4 EXDMAC Timing

Table 41.12 EXDMAC Timing

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC
 VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V
 ICLK = 8 to 100MHz, PCLK = 8 to 50MHz, BCLK = 8 to 100MHz, SDCLK = 8 to 50MHz
 T_a = -40 to +85°C

Item	Symbol	Min.	Max.	Unit	Test Conditions	
EXDMAC	EDREQ setup time	t _{EDRQS}	20	—	ns	Figure 41.22
	EDREQ hold time	t _{EDRQH}	5	—	ns	
	EDACK delay time	t _{EDACD}	—	15	ns	Figure 41.23 and Figure 41.24

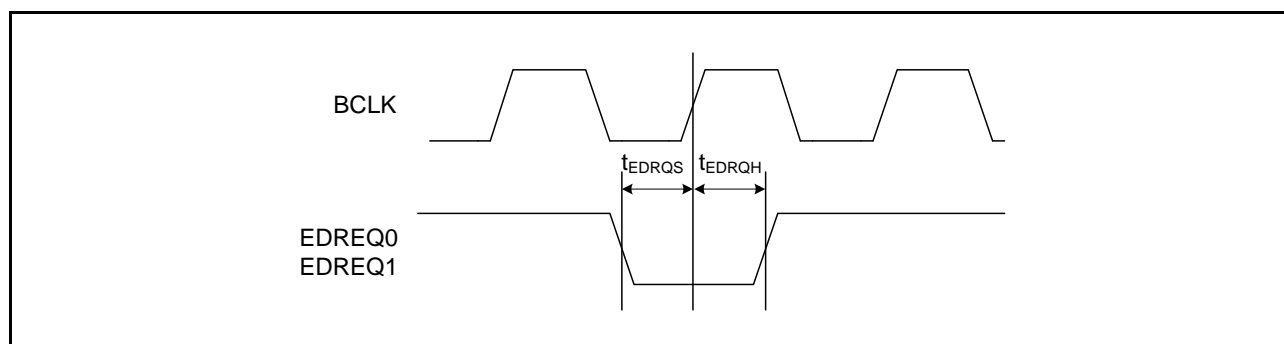


Figure 41.22 EDREQ0 and EDREQ1 Input Timing

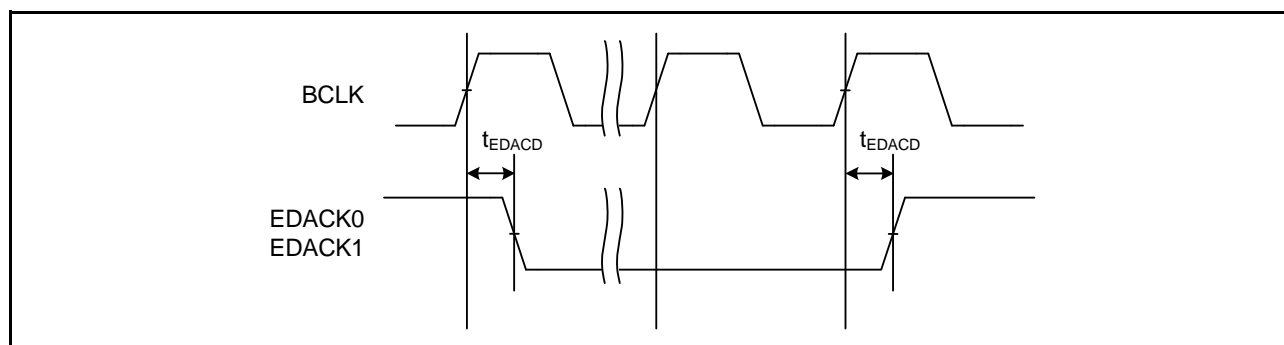


Figure 41.23 EDACK0 and EDACK1 Single-Address Transfer Timing (for a CS Area)

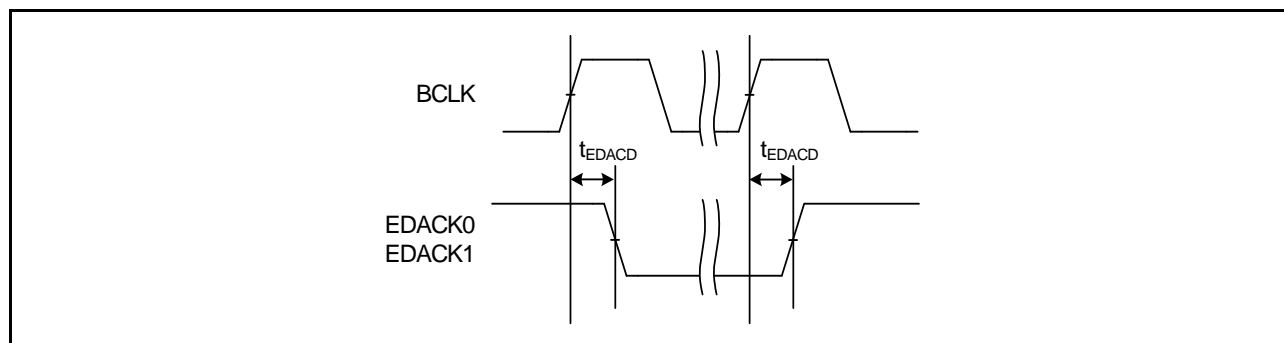


Figure 41.24 EDACK0 and EDACK1 Single-Address Transfer Timing (for SDRAM)

41.3.5 Timing of On-Chip Peripheral Modules

Table 41.13 Timing of On-Chip Peripheral Modules (1)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

PCLK = 8 to 50MHz

T_a = -40 to +85°C

Item		Symbol	Min.	Max.	Unit	Test Conditions
I/O ports	Output data delay time	t _{PWD}	—	40	ns	Figure 41.25
	Input data setup time	t _{PRS}	25	—	ns	
	Input data hold time	t _{PRH}	25	—	ns	
MTU2	Output compare output delay time	t _{TOCD}	—	40	ns	Figure 41.26
	Input capture input setup time	t _{TICS}	20	—	ns	
	Input capture input pulse width (single-edge setting)	t _{TICW}	1.5 × t _{PCyc}	—	ns	
	Input capture input pulse width (both-edge setting)	t _{TICW}	2.5 × t _{PCyc}	—	ns	
	Timer input setup time	t _{TCKS}	20	—	ns	Figure 41.27
	Timer clock pulse width (single-edge setting)	t _{TCKWH/L}	1.5 × t _{PCyc}	—	ns	
	Timer clock pulse width (both-edge setting)	t _{TCKWH/L}	2.5 × t _{PCyc}	—	ns	
	Timer clock pulse width (phase coefficient mode)	t _{TCKWH/L}	2.5 × t _{PCyc}	—	ns	
POE2	POE# input setup time	t _{POES}	50	—	ns	Figure 41.28
	POE# input pulse width	t _{POEW}	1.5 × t _{PCyc}	—	ns	
PPG	Pulse output delay time	t _{POD}	—	40	ns	Figure 41.29
8-bit timer	Timer output delay time	t _{TMOD}	—	40	ns	Figure 41.30
	Timer reset input setup time	t _{TMRS}	25	—	ns	Figure 41.31
	Timer clock input setup time	t _{TMCS}	25	—	ns	Figure 41.32
	Timer clock pulse width	Single-edge setting	t _{TMCWH}	1.5 × t _{PCyc}	—	
Both-edge setting		t _{TMCWL}	2.5 × t _{PCyc}	—	ns	
WDT	Overflow output delay time	t _{WOVD}	—	40	ns	Figure 41.33

Table 41.13 Timing of On-Chip Peripheral Modules (2)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

PCLK = 8 to 50MHz

T_a = -40 to +85°C

Item		Symbol	Min.	Max.	Unit	Test Conditions	
SCI	Input clock cycle	Asynchronous	t _{SCyc}	4 × t _{PCyc}	—	ns	Figure 41.34 and Figure 41.35
		Clock synchronous		6 × t _{PCyc}	—		
	Input clock pulse width		t _{SCKW}	0.4 × t _{SCyc}	0.6 × t _{SCyc}	ns	
	Input clock rise time		t _{SCKr}	—	20	ns	
	Input clock fall time		t _{SCKf}	—	20	ns	
	Output clock cycle	Asynchronous	t _{SCyc}	4 × t _{PCyc}	—	ns	
		Clock synchronous		6 × t _{PCyc}	—		
	Output clock pulse width		t _{SCKW}	0.4 × t _{SCyc}	0.6 × t _{SCyc}	ns	
	Output clock rise time		t _{SCKr}	—	20	ns	
	Output clock fall time		t _{SCKf}	—	20	ns	
	Transmit data delay time (clock synchronous)		t _{TXD}	—	40	ns	
	Receive data setup time (clock synchronous)		t _{RXS}	40	—	ns	
Receive data hold time (clock synchronous)		t _{RXH}	40	—	ns		
A/D converter	10-bit A/D converter trigger input setup time	t _{TRGS}	25	—	ns	Figure 41.36	
	12-bit A/D converter trigger input setup time	t _{TRGS}	25	—	ns		

Table 41.14 Timing of On-Chip Peripheral Modules (3)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

PCLK = 8 to 50MHz

T_a = -40 to +85°C

Item		Symbol	Min.	Max.	Unit	Test Conditions		
CAN	Transmit data delay time	t _{CTXD}	—	40.0	ns	Figure 41.37		
	Receive data setup time	t _{CRXS}	40.0	—	ns			
	Receive data hold time	t _{CRXH}	40.0	—	ns			
RSPI	RSPCK clock cycle	Master	t _{SPcyc}	2	4096	t _{Pcyc} *1	Figure 41.38	
		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 [176-pin LFBGA/ 145-pin TFLGA/ 144-pin LQFP]	t _{SPCKr} , t _{SPCKf}	—	5	ns		
		Output [100-pin LQFP/ 85-pin TFLGA]		—	10			
		Input		—	1			μs

Note 1. t_{Pcyc}: PCLK cycle

Table 41.14 Timing of On-Chip Peripheral Modules (4)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

PCLK = 8 to 50MHz

T_a = -40 to +85°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
RSPI	Data input setup time	Master [176-pin LFBGA/ 145-pin TFLGA/ 144-pin LQFP]	t _{SU}	16	—	ns	Figure 41.39 to Figure 41.42
		Master [100-pin LQFP/ 85-pin TFLGA]		30	—		
		Slave		20-2 × t _{Pcyc}	—		
	Data input hold time	Master	t _H	0	—	ns	
		Slave		20+2 × t _{Pcyc}	—		
	SSL setup time	Master	t _{LEAD}	1	8	t _{SPcyc}	
		Slave		4	—	t _{Pcyc}	
	SSL hold time	Master	t _{LAG}	1	8	t _{SPcyc}	
		Slave		4	—	t _{Pcyc}	
	Data output delay time	Master [176-pin LFBGA/ 145-pin TFLGA/ 144-pin LQFP]	t _{OD}	—	20	ns	
		Master [100-pin LQFP/ 85-pin TFLGA]		—	30		
		Slave [176-pin LFBGA/ 145-pin TFLGA/ 144-pin LQFP]		—	3 × t _{Pcyc} +40		
		Slave [100-pin LQFP/ 85-pin TFLGA]		—	3 × t _{Pcyc} +50		

Note 1. t_{Pcyc}: PCLK cycle

Table 41.15 Timing of On-Chip Peripheral Modules (5)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

PCLK = 8 to 50MHz

T_a = -40 to +85°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
RSPi	Data output hold time	Master	t _{OH}	0	—	ns	Figure 41.39 to Figure 41.42
		Slave		0	—		
	Successive transmission delay time	Master	t _{TD}	t _{SPcyc} +2 × t _{Pcyc}	8 × t _{SPcyc} +2 × t _{Pcyc}	ns	
		Slave		4 × t _{Pcyc}	—		
	MOSI, MISO rise/fall time	Output [176-pin LFBGA/ 145-pin TFLGA/ 144-pin LQFP]	t _{Dr} , t _{Df}	—	5	ns	
		Output [100-pin LQFP/ 85-pin TFLGA]		—	10		
		Input		—	1	μs	
	SSL rise/fall time	Output [176-pin LFBGA 145-pin TFLGA 144-pin LQFP]	t _{SSLr} , t _{SSLf}	—	5	ns	
		Output [100-pin LQFP/ 85-pin TFLGA]		—	10		
		Input		—	1	μs	
	Slave access time		t _{SA}	—	4	t _{Pcyc}	
Slave output release time		t _{REL}	—	3	t _{Pcyc}		

Note 1. t_{Pcyc}: PCLK cycle

Table 41.16 Timing of On-Chip Peripheral Modules (6)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

PCLK = 8 to 50MHz

T_a = -40 to +85°C

Item	Symbol	Min. ^{*1*2}	Max.	Unit	Test Conditions	
RIIC (Standard-mode, SMBus) ICFER.FMPE = 0	SCL input cycle time	t _{SCL}	6(12) × t _{IICcyc} + 1300	—	ns	Figure 41.43
	SCL input high pulse width	t _{SCLH}	3(6) × t _{IICcyc} + 300	—	ns	
	SCL input low pulse width	t _{SCLL}	3(6) × t _{IICcyc} + 300	—	ns	
	SCL, SDA input rising time	t _{Sr}	—	1000	ns	
	SCL, SDA input falling time	t _{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t _{SP}	0	1(4) × t _{IICcyc}	ns	
	SDA input bus free time	t _{BUF}	3(6) × t _{IICcyc} + 300	—	ns	
	Start condition input hold time	t _{STAH}	t _{IICcyc} + 300	—	ns	
	Re-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	
	SCL, SDA capacitive load	C _b	—	400	pF	
RIIC (Fast-mode)	SCL input cycle time	t _{SCL}	6(12) × t _{IICcyc} + 600	—	ns	
	SCL input high pulse width	t _{SCLH}	3(6) × t _{IICcyc} + 300	—	ns	
	SCL input low pulse width	t _{SCLL}	3(6) × t _{IICcyc} + 300	—	ns	
	SCL, SDA input rising time	t _{Sr}	20+0.1C _b	300	ns	
	SCL, SDA input falling time	t _{Sf}	20+0.1C _b	300	ns	
	SCL, SDA input spike pulse removal time	t _{SP}	0	1(4) × t _{IICcyc}	ns	
	SDA input bus free time	t _{BUF}	3(6) × t _{IICcyc} + 300	—	ns	
	Start condition input hold time	t _{STAH}	t _{IICcyc} + 300	—	ns	
	Re-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	
	SCL, SDA capacitive load	C _b	—	400	pF	

Note: t_{IICcyc}: RIIC internal reference clock (IICφ) cycles

Note 1. The value in parentheses is used when ICMR3.NF[1:0] are set to 11b while a digital filter is enabled with ICFER.NFE = 1.

Note 2. C_b indicates the total capacity of the bus line.

Table 41.16 Timing of On-Chip Peripheral Modules (7)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

PCLK = 8 to 50MHz

T_a = -40 to +85°C

Item	Symbol	Min.*1*2	Max.	Unit	Test Conditions	
RIIC (Fast-mode+) ICFER.FMPE = 1	SCL input cycle time	t _{SCL}	6(12) × t _{IICcyc} + 240	—	ns	Figure 41.43
	SCL input high pulse width	t _{SCLH}	3(6) × t _{IICcyc} + 120	—	ns	
	SCL input low pulse width	t _{SCLL}	3(6) × t _{IICcyc} + 120	—	ns	
	SCL, SDA input rising time	t _{Sr}	—	120	ns	
	SCL, SDA input falling time	t _{Sf}	—	120	ns	
	SCL, SDA input spike pulse removal time	t _{SP}	0	1(4) × t _{IICcyc}	ns	
	SDA input bus free time	t _{BUF}	3(6) × t _{IICcyc} + 120	—	ns	
	Start condition input hold time	t _{STAH}	t _{IICcyc} + 120	—	ns	
	Re-start condition input setup time	t _{STAS}	120	—	ns	
	Stop condition input setup time	t _{STOS}	120	—	ns	
	Data input setup time	t _{SDAS}	t _{IICcyc} + 20	—	ns	
	Data input hold time	t _{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C _b	—	550	pF	

Note: t_{IICcyc}: RIIC internal reference clock (IICφ) cycles

Note 1. The value in parentheses is used when ICMR3.NF[1:0] are set to 11b while a digital filter is enabled with ICFER.NFE = 1.

Note 2. C_b indicates the total capacity of the bus line.

Table 41.17 Timing of On-Chip Peripheral Modules (8)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

ICLK = 12.5 to 100MHz

T_a = -40 to +85°C

Item	Symbol	Min.	Max.	Unit	Test Conditions	
ETHERC(RMII)	REF50CK cycle time	T _{ck}	20	—	ns	Figure 41.44 to Figure 41.47
	REF50CK frequency Typ. 50MHz	—	—	50 + 100ppm	MHz	
	REF50CK duty	—	35	65	%	
	REF50CK rise/fall time	T _{ckr/ckf}	0.5	3.5	ns	
	RMII_xxxx ^{*1} output delay time	T _{co}	2.5	12.5	ns	
	RMII_xxxx ^{*2} setup time	T _{su}	3	—	ns	
	RMII_xxxx ^{*2} hold time	T _{hd}	1	—	ns	
	RMII_xxxx ^{*1*2} rise/fall time	Tr/Tf	0.5	6	ns	
	ET_MDIO setup time	t _{MDIOs}	10	—	ns	Figure 41.48
	ET_MDIO hold time	t _{MDIOh}	10	—	ns	
	ET_MDIO output hold time ^{*3}	t _{MDIODh}	5	—	ns	Figure 41.49
	ET_WOL output delay time	t _{WOLd}	1	20	ns	Figure 41.50

Note 1. RMII_TXD_EN, RMII_TXD1, RMII_TXD0

Note 2. RMII_CRSDV, RMII_RXD1, RMII_RXD0, RMII_RX_ER

Note 3. The user program must make settings so that this stipulation is satisfied.

Table 41.17 Timing of On-Chip Peripheral Modules (9)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

ICLK = 12.5 to 100MHz

T_a = -40 to +85°C

Item	Symbol	Min.	Max.	Unit	Test Conditions	
ETHERC(MII)	ET_TX_CLK cycle time	t _{Tcyc}	40	—	ns	—
	ET_TX_EN output delay time	t _{TENd}	1	20	ns	Figure 41.51
	ET_ETXD0 to ET_ETXD3 output delay time	t _{MTDd}	1	20	ns	
	ET_CRs setup time	t _{CRSs}	10	—	ns	
	ET_CRs hold time	t _{CRSh}	10	—	ns	
	ET_COL setup time	t _{COLs}	10	—	ns	Figure 41.52
	ET_COL hold time	t _{COLh}	10	—	ns	
	ET_RX_CLK cycle time	t _{TRcyc}	40	—	ns	—
	ET_RX_DV setup time	t _{RDVs}	10	—	ns	Figure 41.53
	ET_RX_DV hold time	t _{RDVh}	10	—	ns	
	ET_ERXD0 to ET_ERXD3 setup time	t _{MRDs}	10	—	ns	
	ET_ERXD0 to ET_ERXD3 hold time	t _{MRDh}	10	—	ns	
	ET_RX_ER setup time	t _{RERs}	10	—	ns	Figure 41.54
	ET_RX_ER hold time	t _{RESh}	10	—	ns	
	ET_MDIO setup time	t _{MDIOs}	10	—	ns	Figure 41.55
	ET_MDIO hold time	t _{MDIOh}	10	—	ns	
	ET_MDIO output hold time	t _{MDIOdh}	5	—	ns	Figure 41.56
	ET_WOL output delay time	t _{WOLd}	1	20	ns	Figure 41.57

Note 1. RMII_TXD_EN, RMII_TXD1, RMII_TXD0

Note 2. RMII_CRs_DV, RMII_RXD1, RMII_RXD0, RMII_RX_ER

Note 3. The user program must make settings so that this stipulation is satisfied.

Table 41.18 Timing of On-Chip Peripheral Modules (10)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC
 VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V
 PCLK = 8 to 50MHz
 T_a = -40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
TCK clock cycle time	t _{TCKcyc}	100	—	—	ns	Figure 41.58
TCK clock high pulse width	t _{TCKH}	45	—	—	ns	
TCK clock low pulse width	t _{TCKL}	45	—	—	ns	
TCK clock rising time	t _{TCKr}	—	—	5	ns	
TCK clock falling time	t _{TCKf}	—	—	5	ns	
TRST# pulse width	t _{TRSTW}	20	—	—	t _{TCKcyc}	Figure 41.59
TMS setup time	t _{TMSS}	20	—	—	ns	Figure 41.60
TMS hold time	t _{TMSH}	20	—	—	ns	
TDI setup time	t _{TDIS}	20	—	—	ns	
TDI hold time	t _{TDIH}	20	—	—	ns	
TDO data delay time	t _{TDOD}	—	—	40	ns	

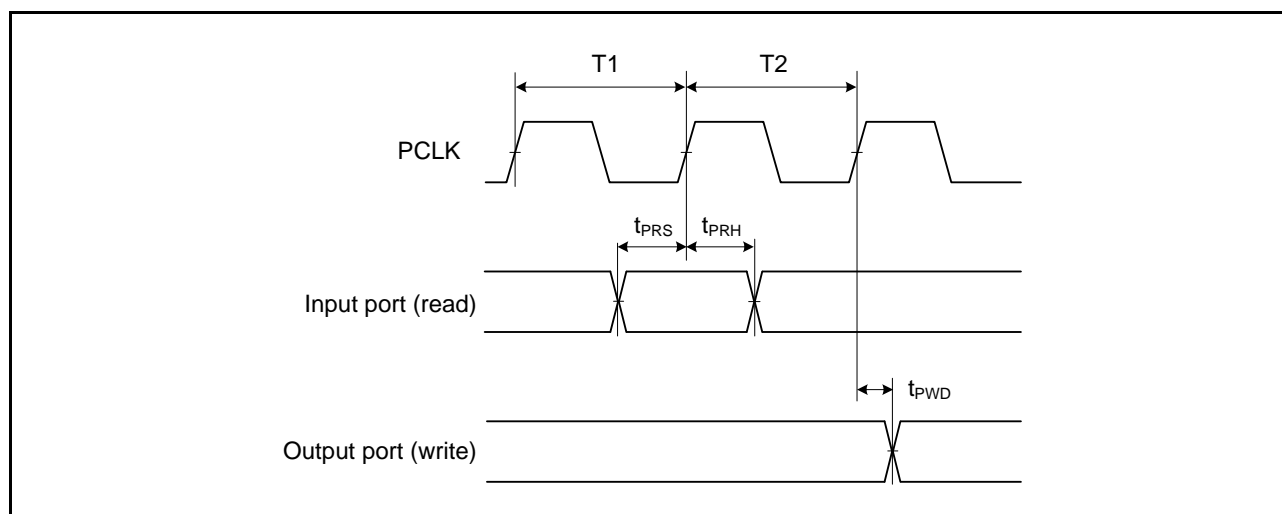


Figure 41.25 I/O Port Input/Output Timing

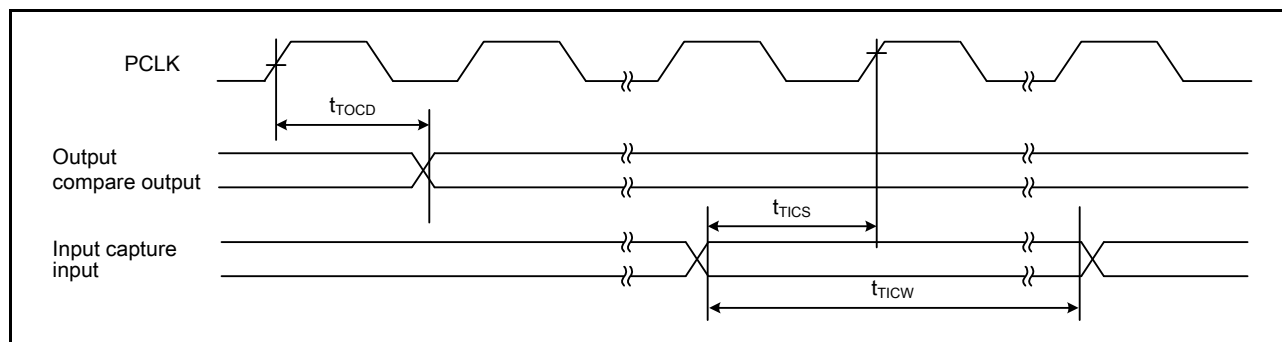


Figure 41.26 MTU2 Input/Output Timing

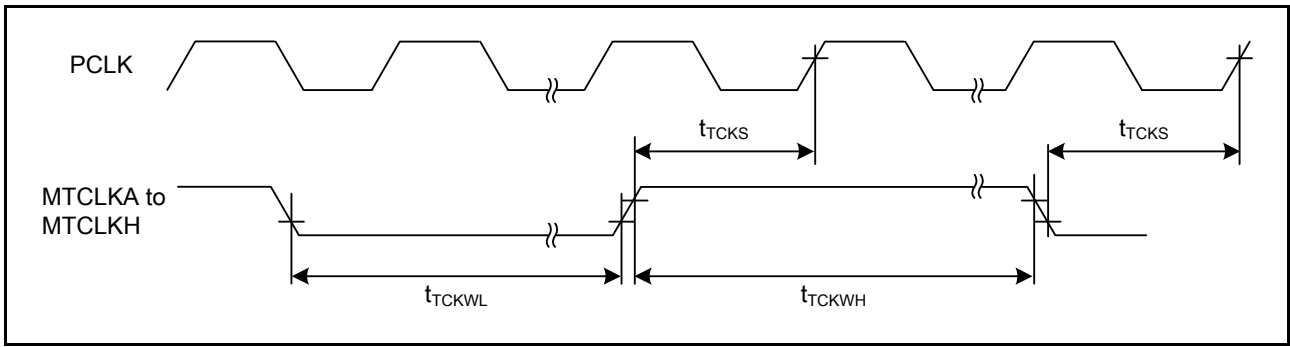


Figure 41.27 MTU2 Clock Input Timing

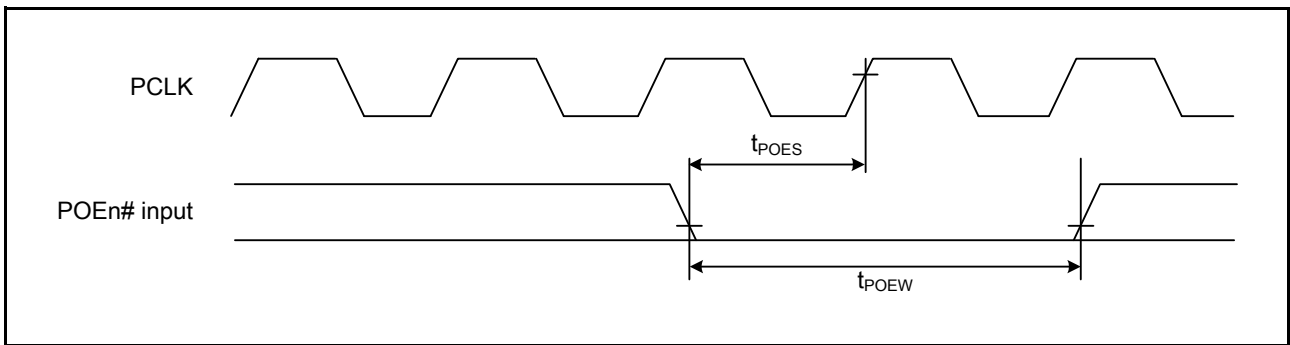


Figure 41.28 POE# Input Timing

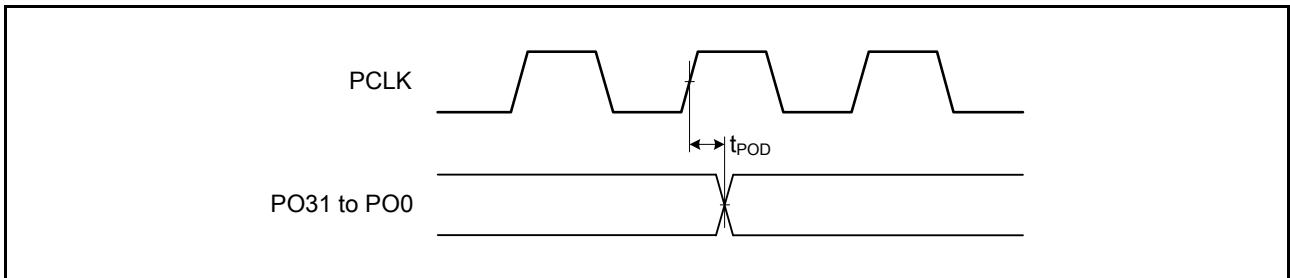


Figure 41.29 PPG Output Timing

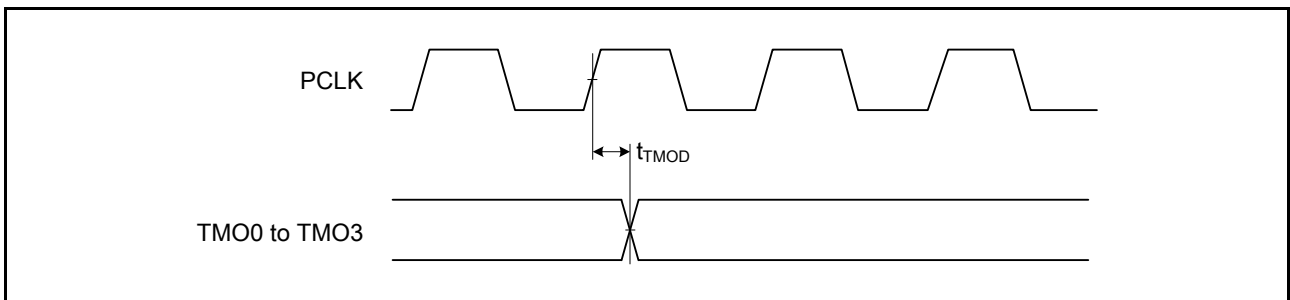


Figure 41.30 8-Bit Timer Output Timing

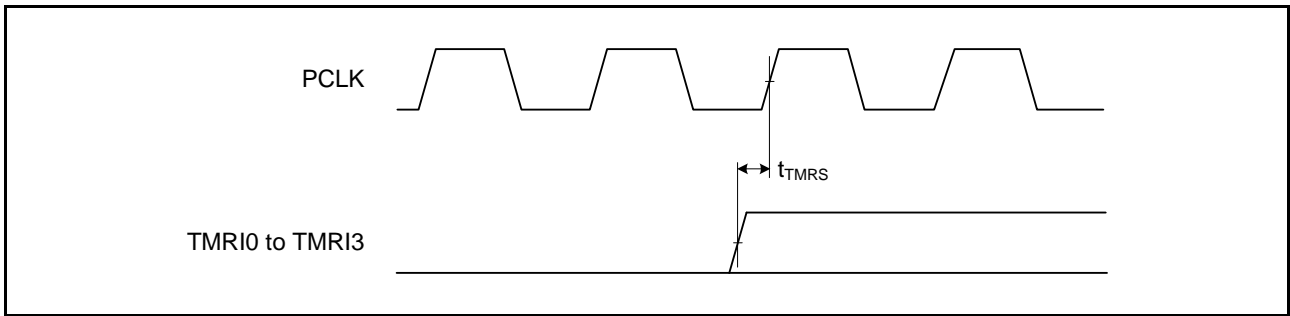


Figure 41.31 8-Bit Timer Reset Input Timing

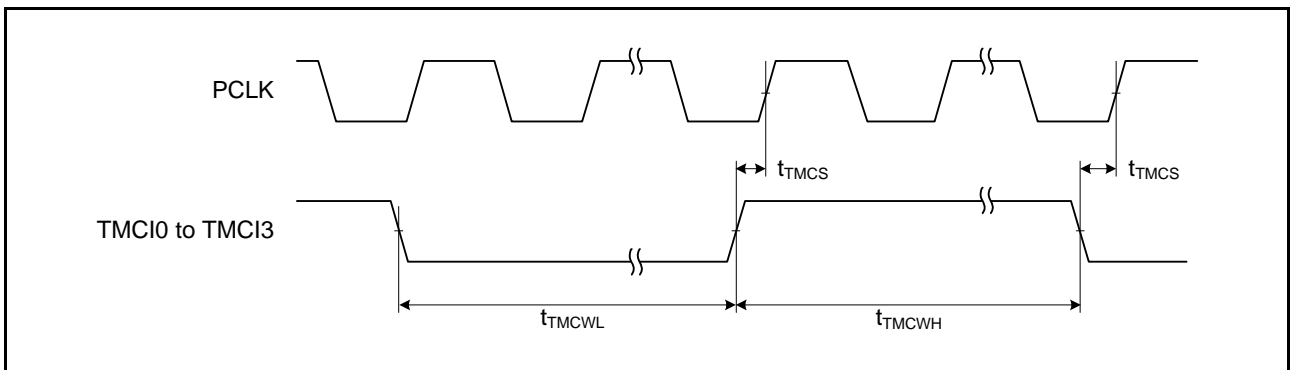


Figure 41.32 8-Bit Timer Clock Input Timing

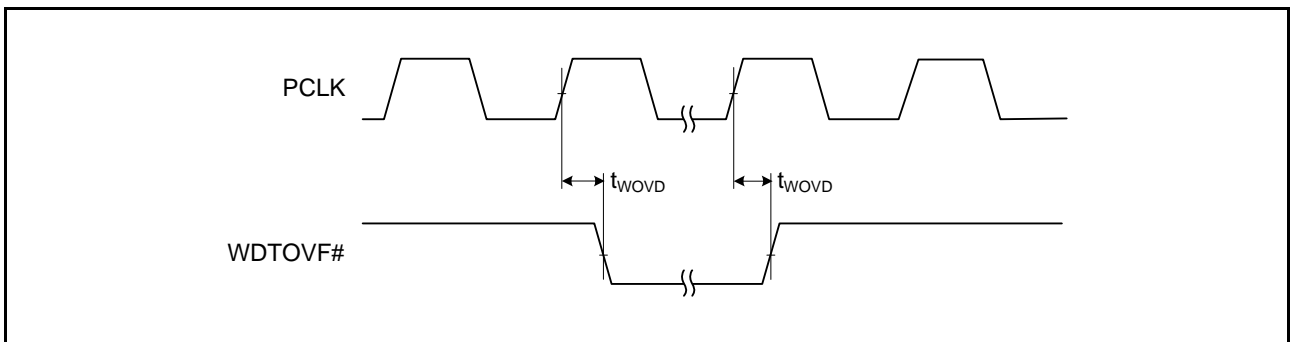


Figure 41.33 WDT Output Timing

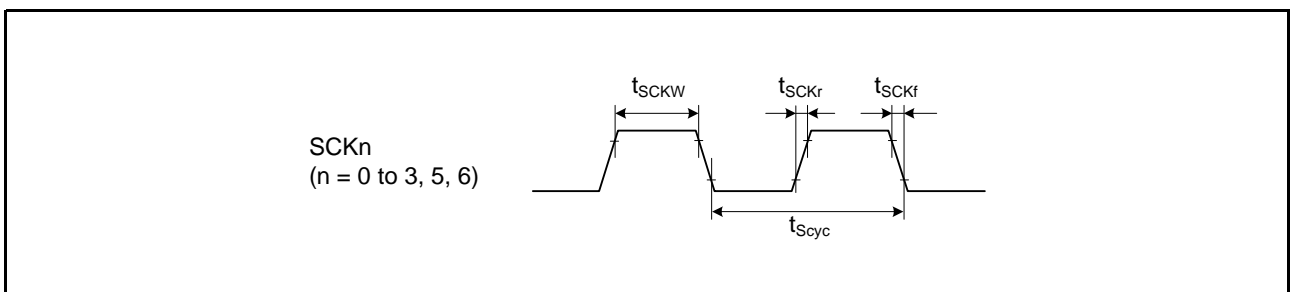


Figure 41.34 SCK Clock Input Timing

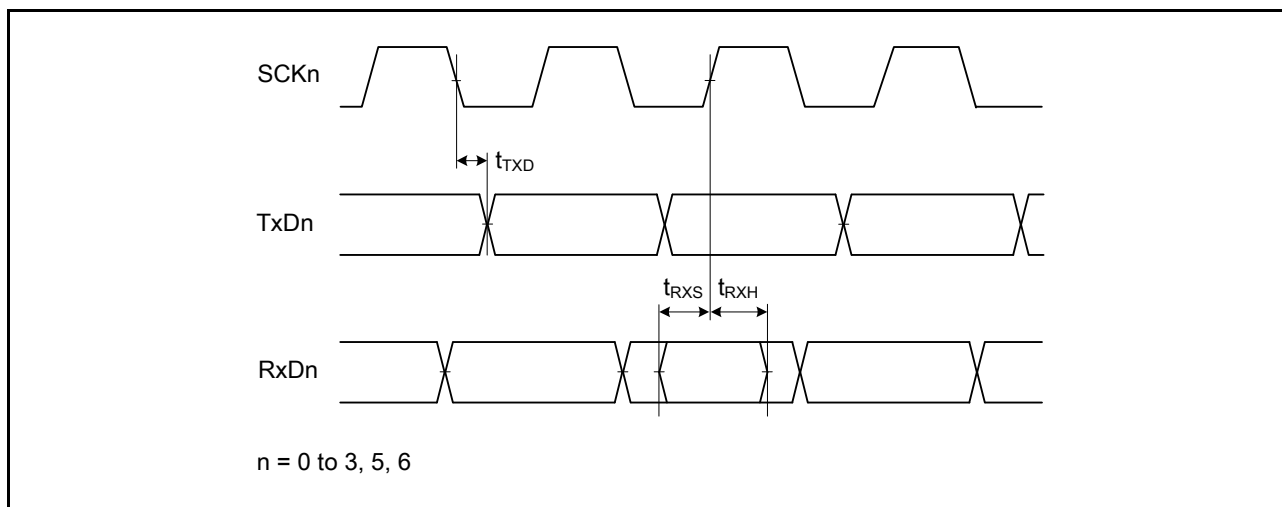


Figure 41.35 SCI Input/Output Timing: Clock Synchronous Mode

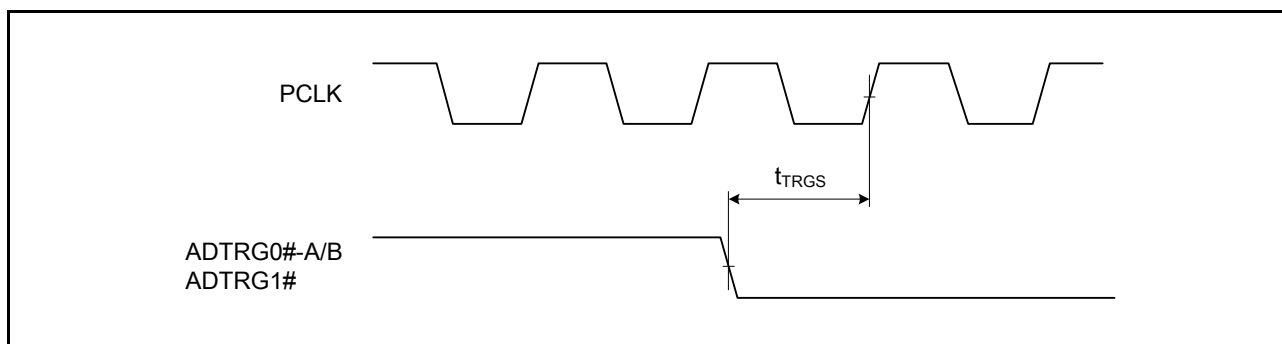


Figure 41.36 A/D Converter External Trigger Input Timing

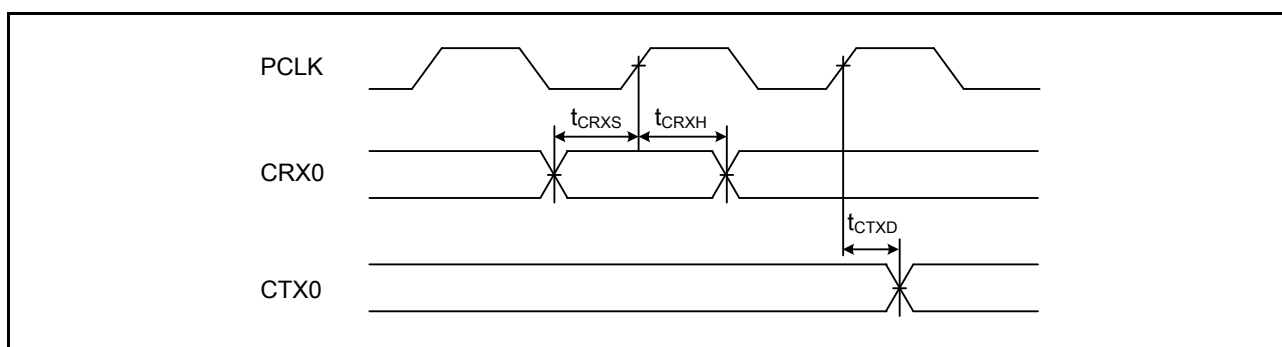


Figure 41.37 CAN Input/Output Timing

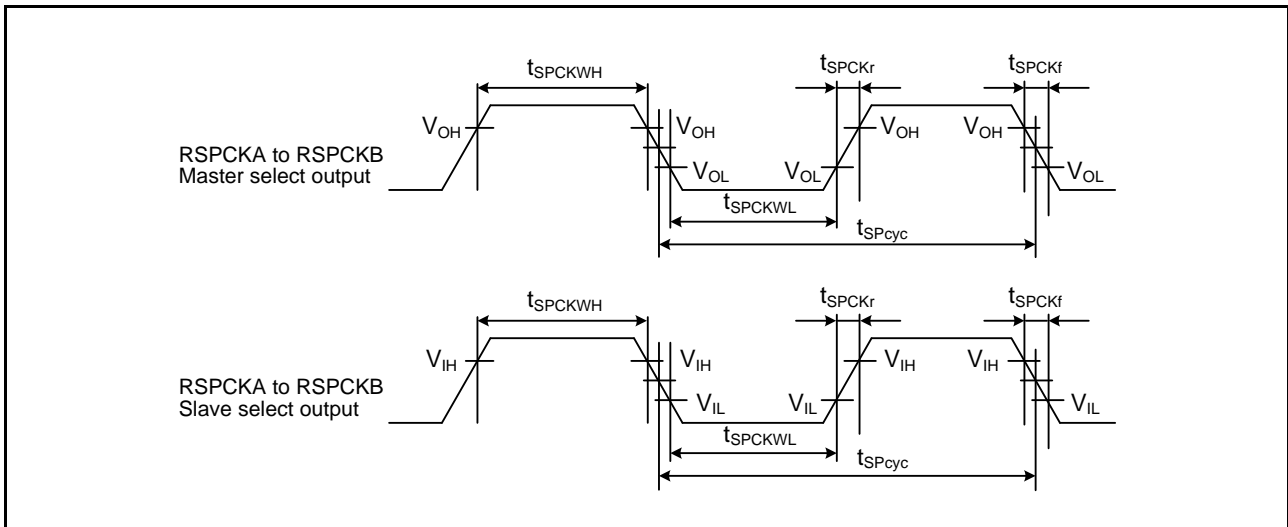


Figure 41.38 RSPCKA to RSPCKB Master select output

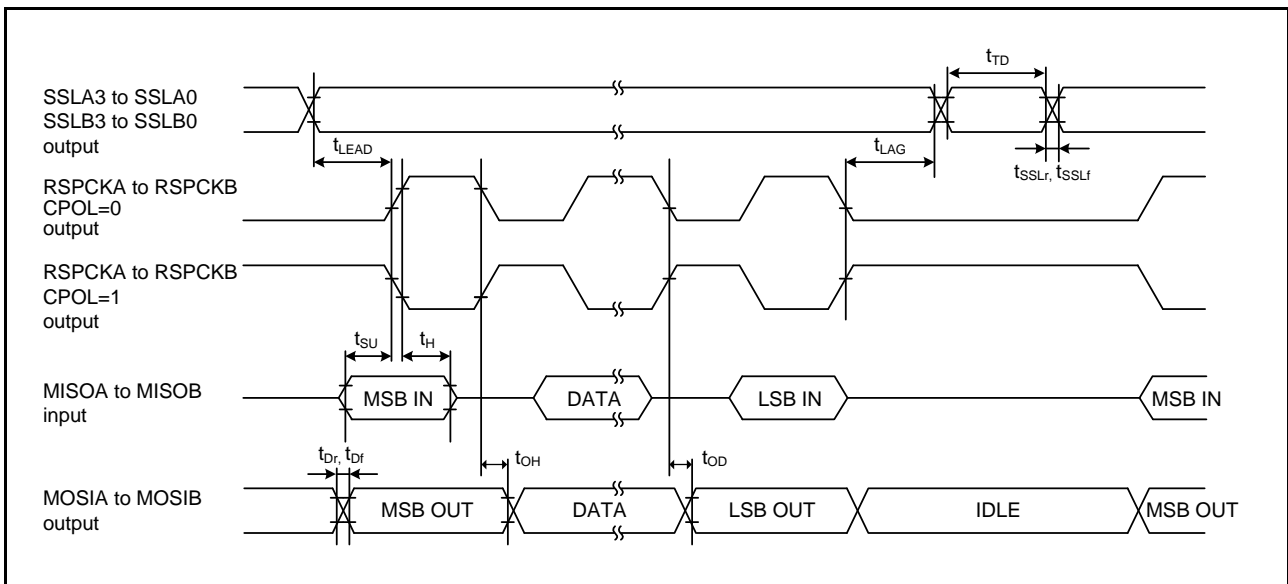


Figure 41.39 RSPCKA to RSPCKB Master, CPOL = 0

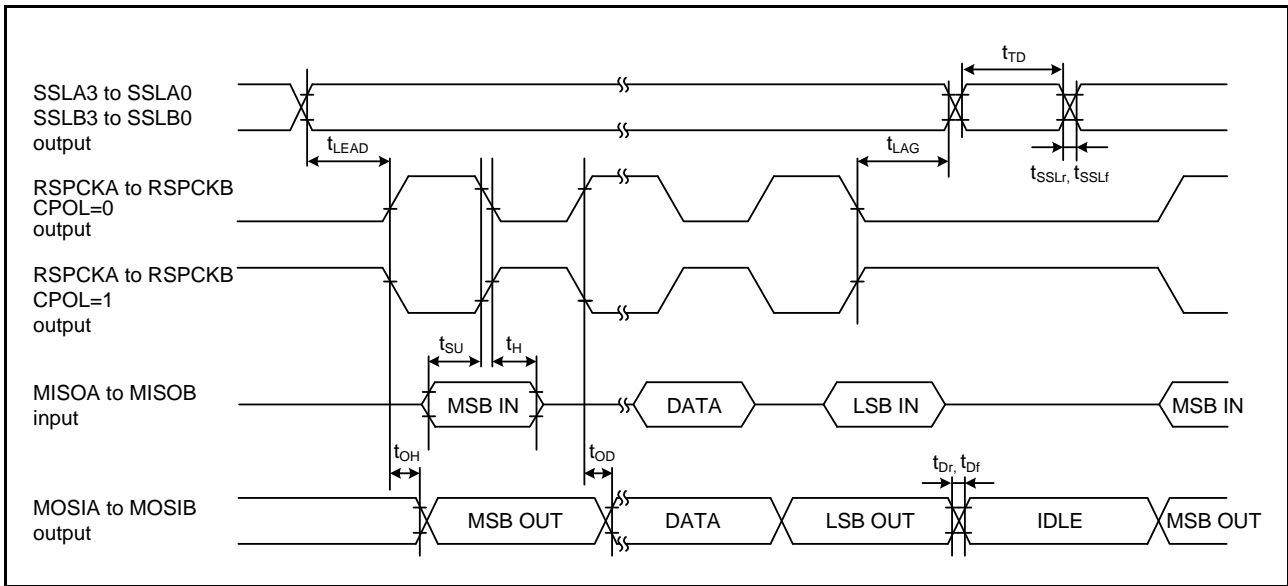


Figure 41.40 RSPI Timing (Master, CPHA = 1)

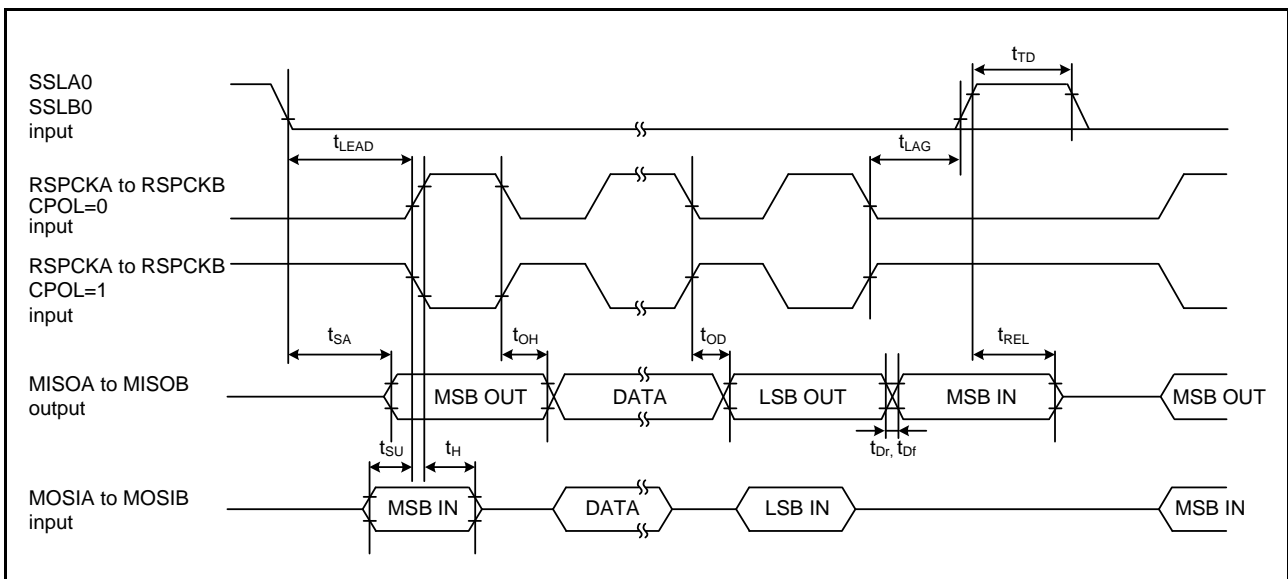


Figure 41.41 RSPI Timing (Slave, CPHA = 0)

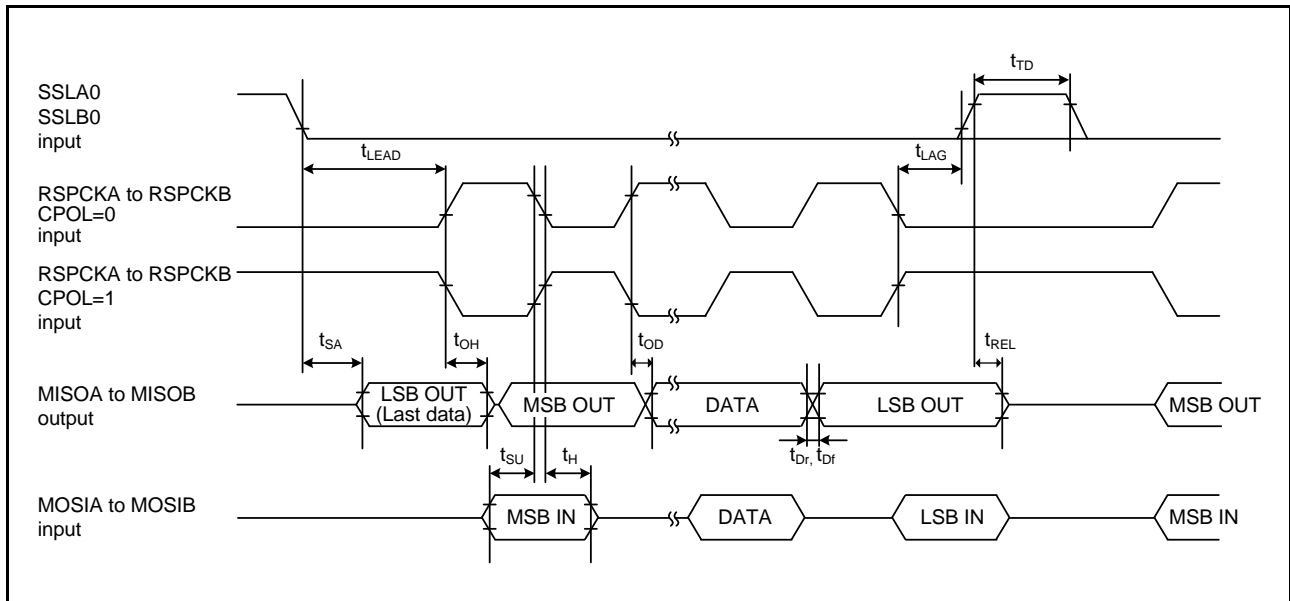


Figure 41.42 RSPI Timing (Slave, CPHA = 1)

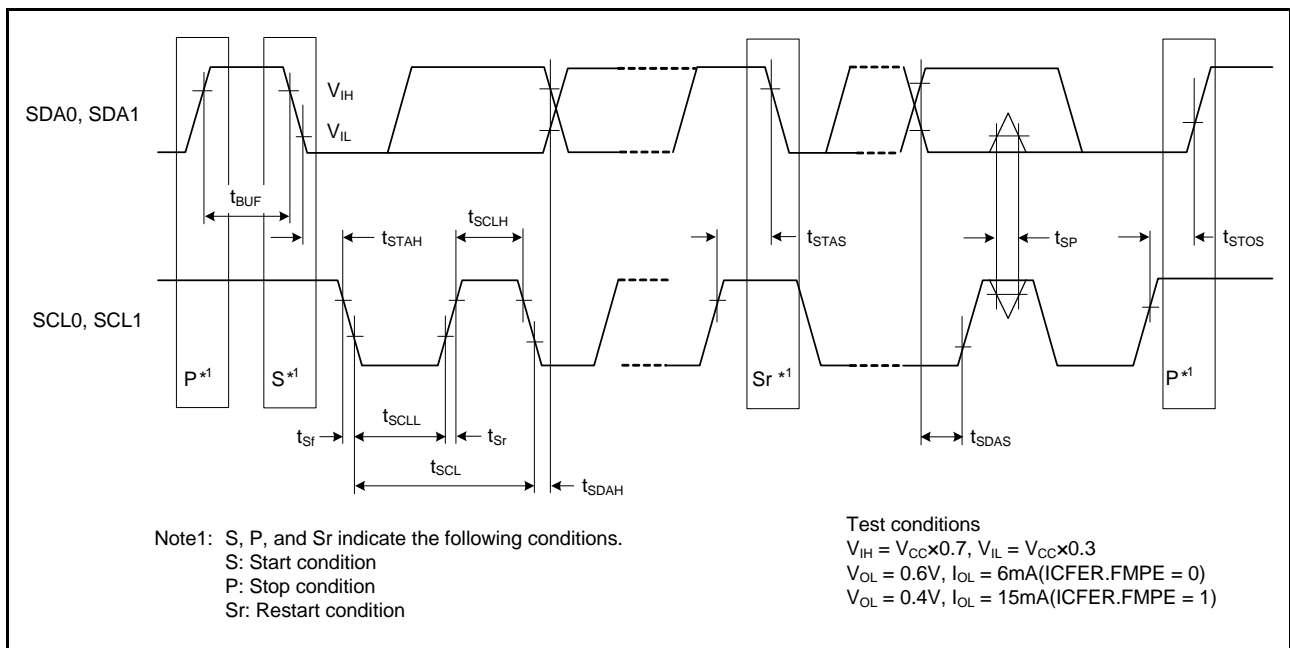


Figure 41.43 I2C Bus Interface Input/Output Timing

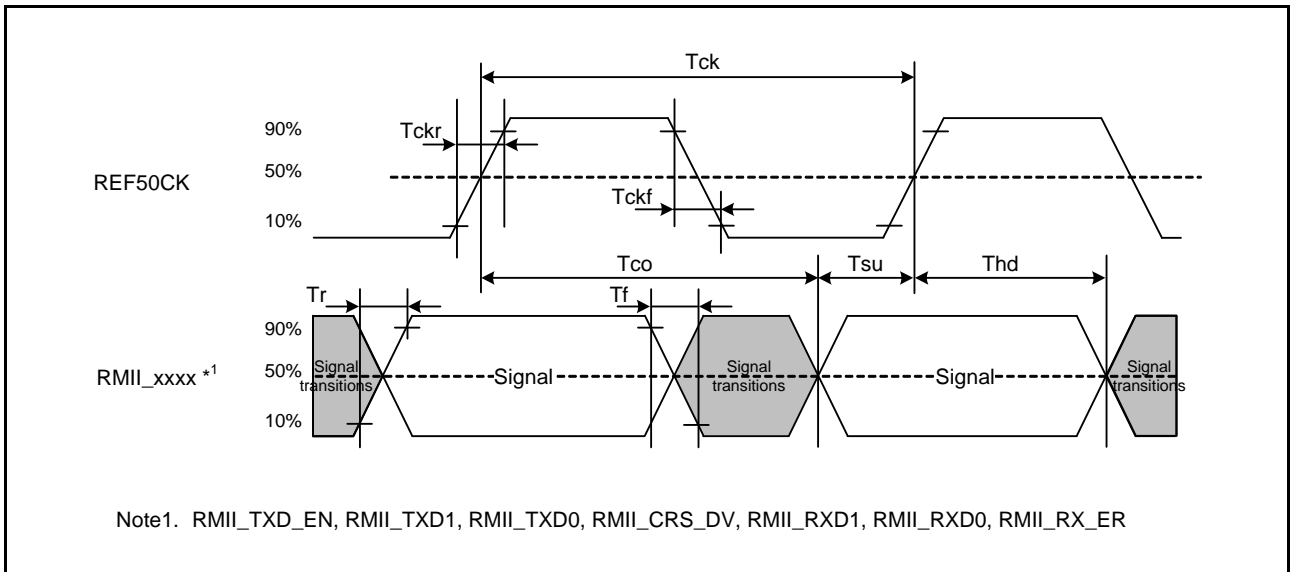


Figure 41.44 REF50CK and RMII Signal Timing

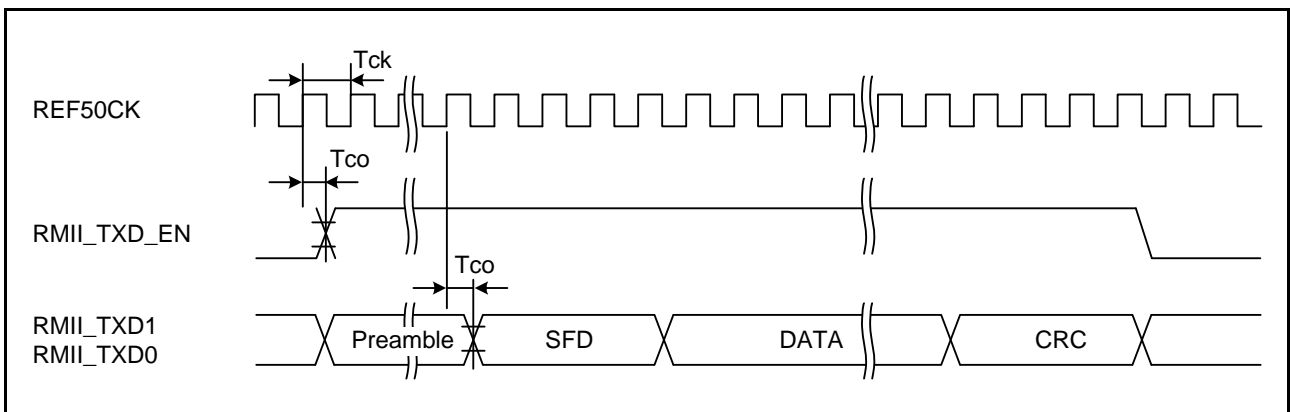


Figure 41.45 RMII Transmission Timing

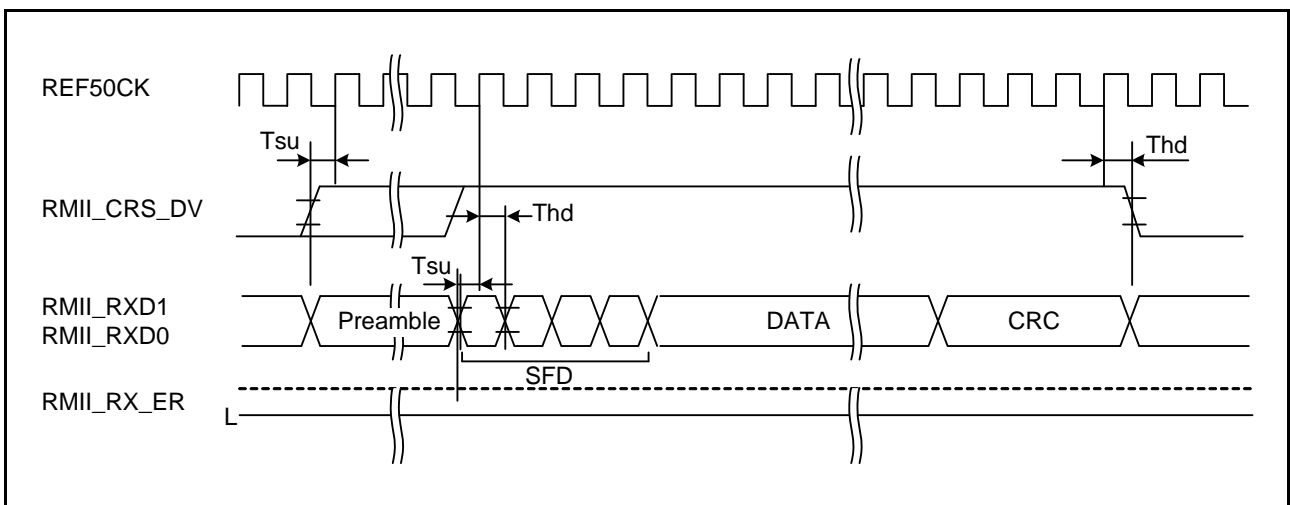


Figure 41.46 RMII Reception Timing (Normal Operation)

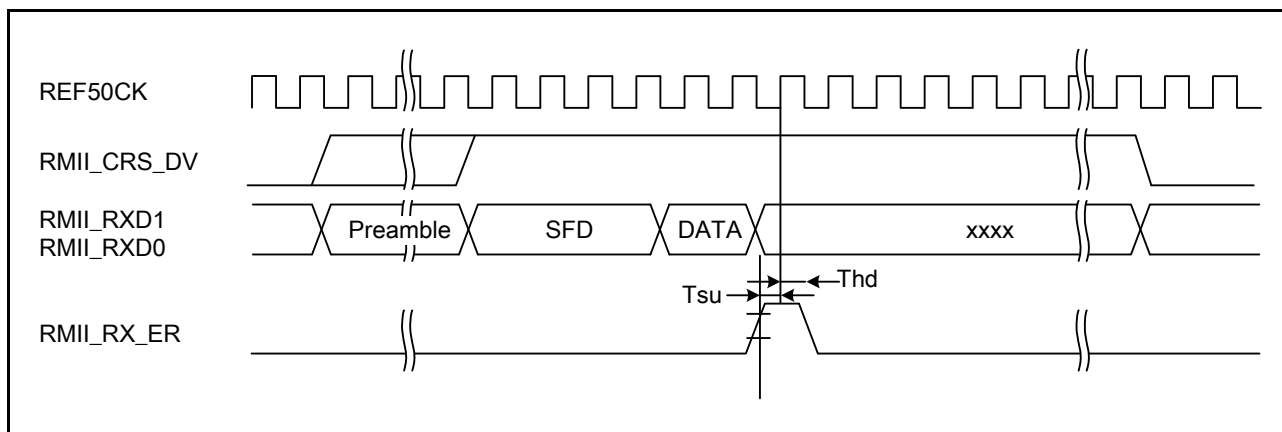


Figure 41.47 RMI Reception Timing (Error Occurrence)

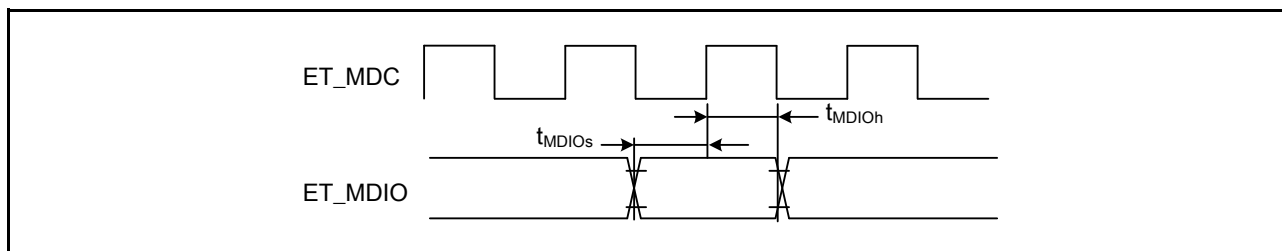


Figure 41.48 MDIO Input Timing (RMI)

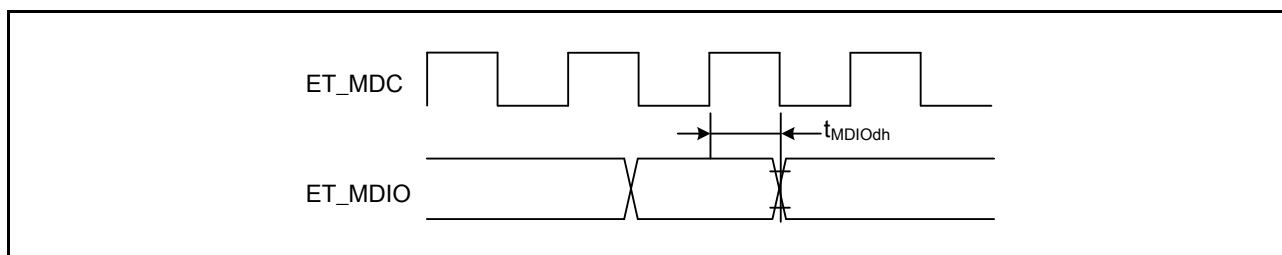


Figure 41.49 MDIO Output Timing (RMI)

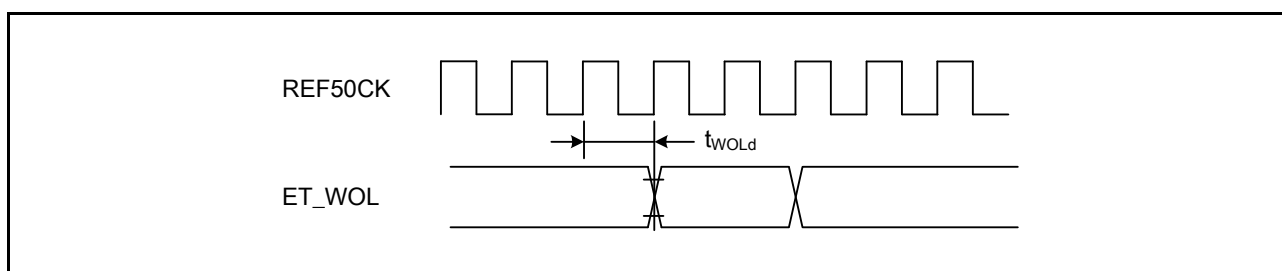


Figure 41.50 WOL Output Timing (RMI)

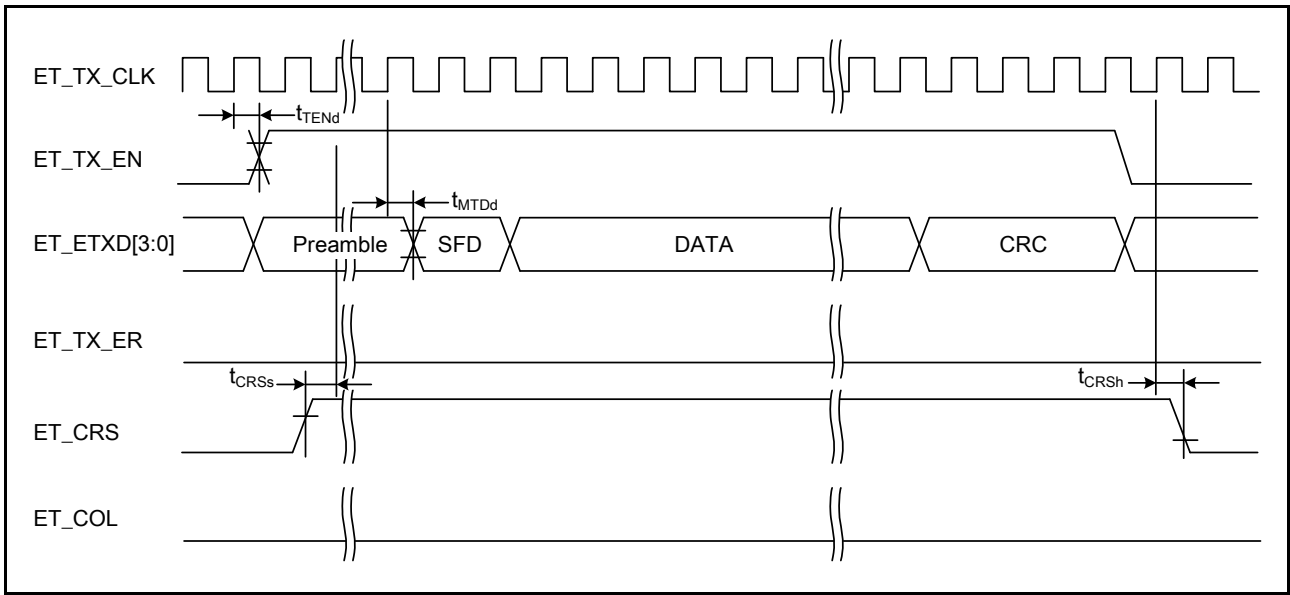


Figure 41.51 MII Transmission Timing (Normal Operation)

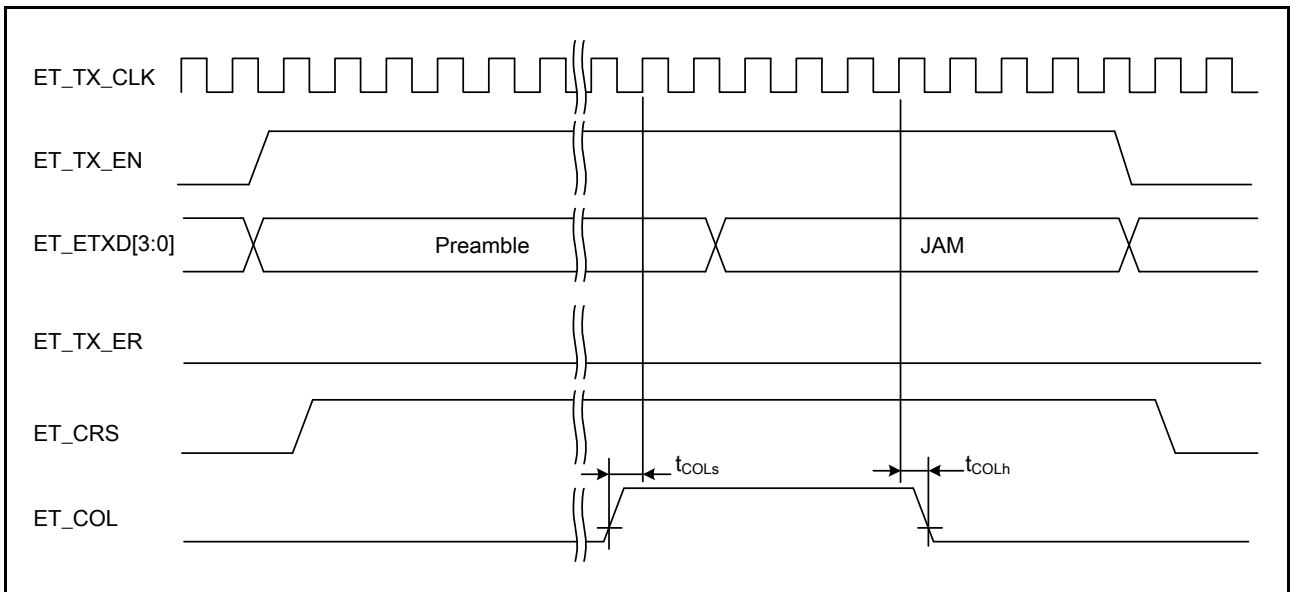


Figure 41.52 MII Transmission Timing (Conflict Occurrence)

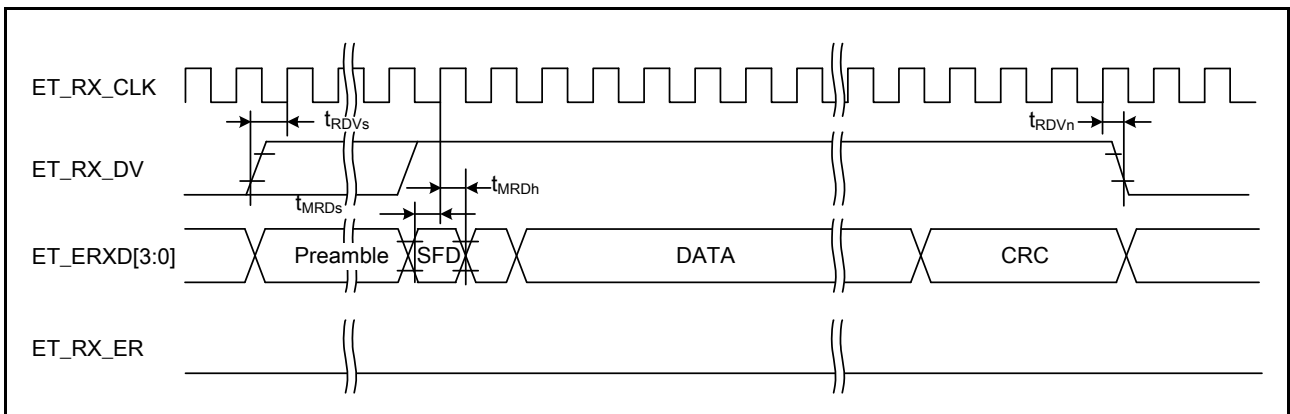


Figure 41.53 MII Reception Timing (Normal Operation)

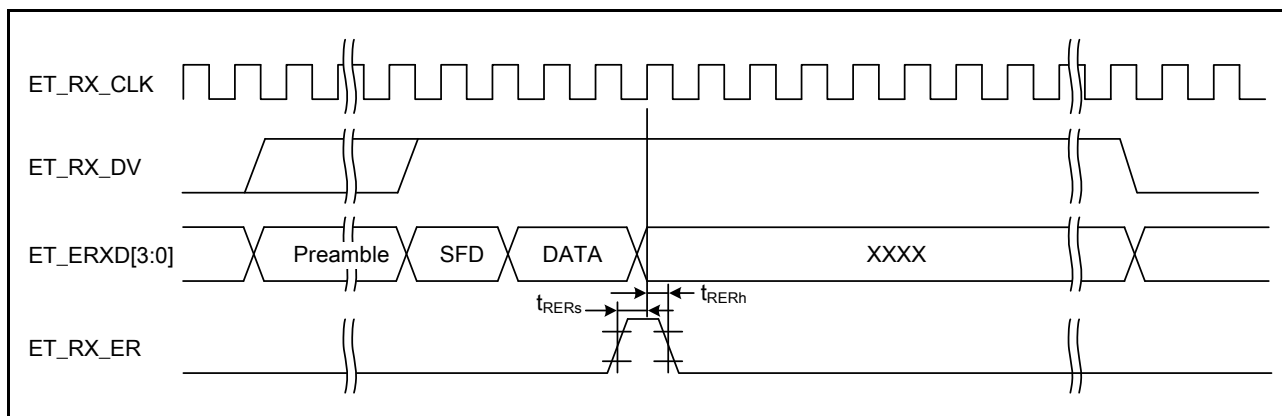


Figure 41.54 MII Reception Timing (Error Occurrence)

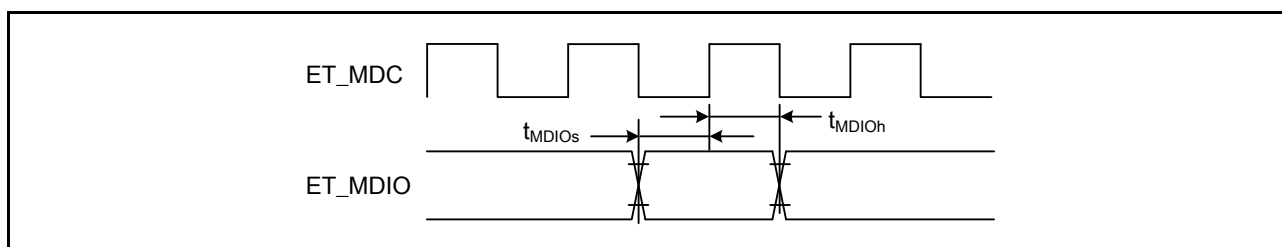


Figure 41.55 MDIO Input Timing (MII)

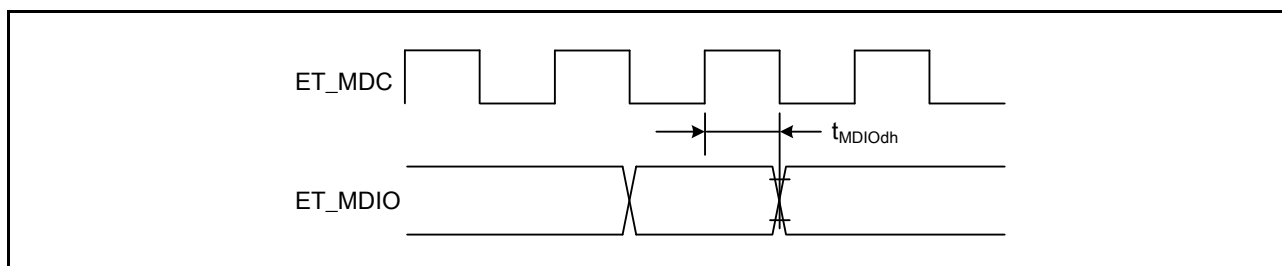


Figure 41.56 MDIO Output Timing (MII)

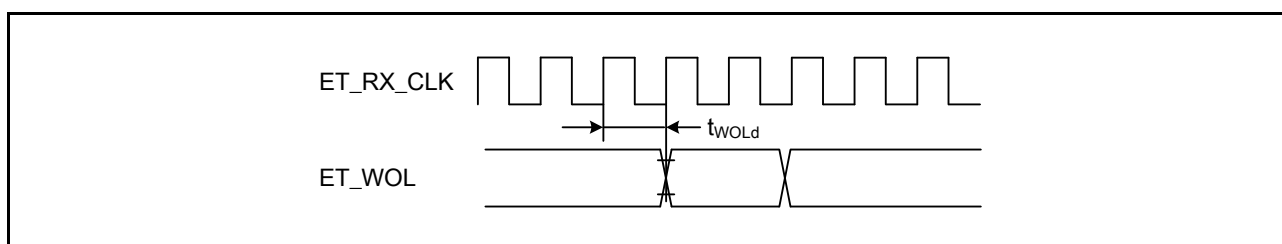


Figure 41.57 WOL Output Timing (MII)

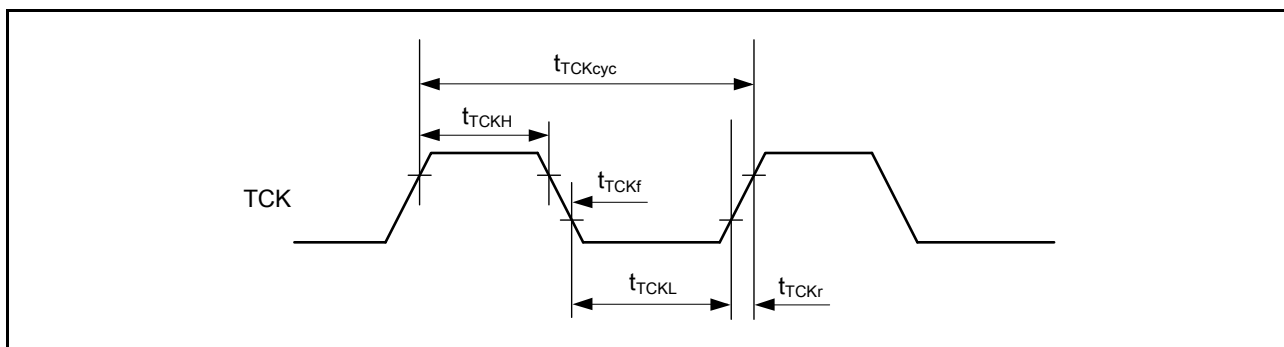


Figure 41.58 Boundary Scan TCK Timing

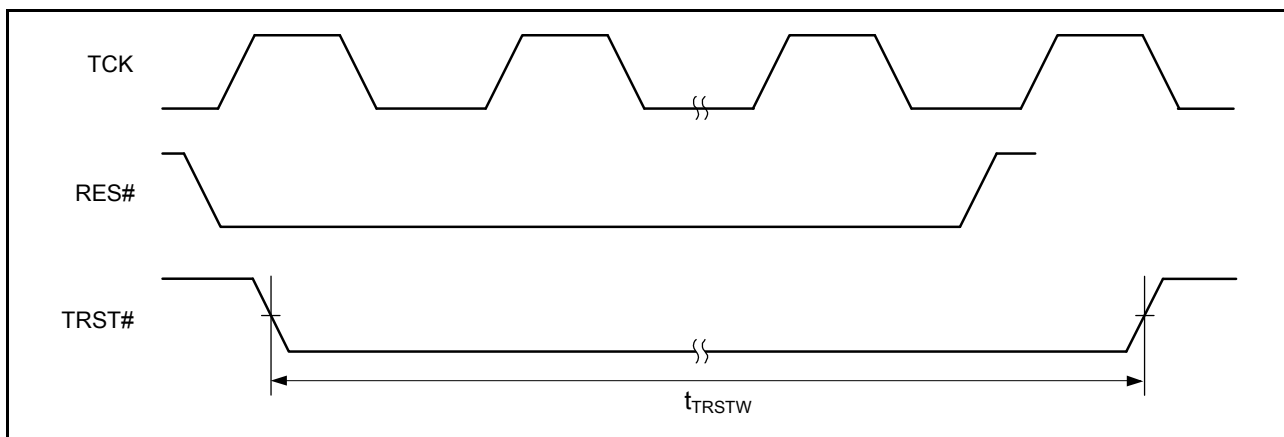


Figure 41.59 Boundary Scan TRST# Timing

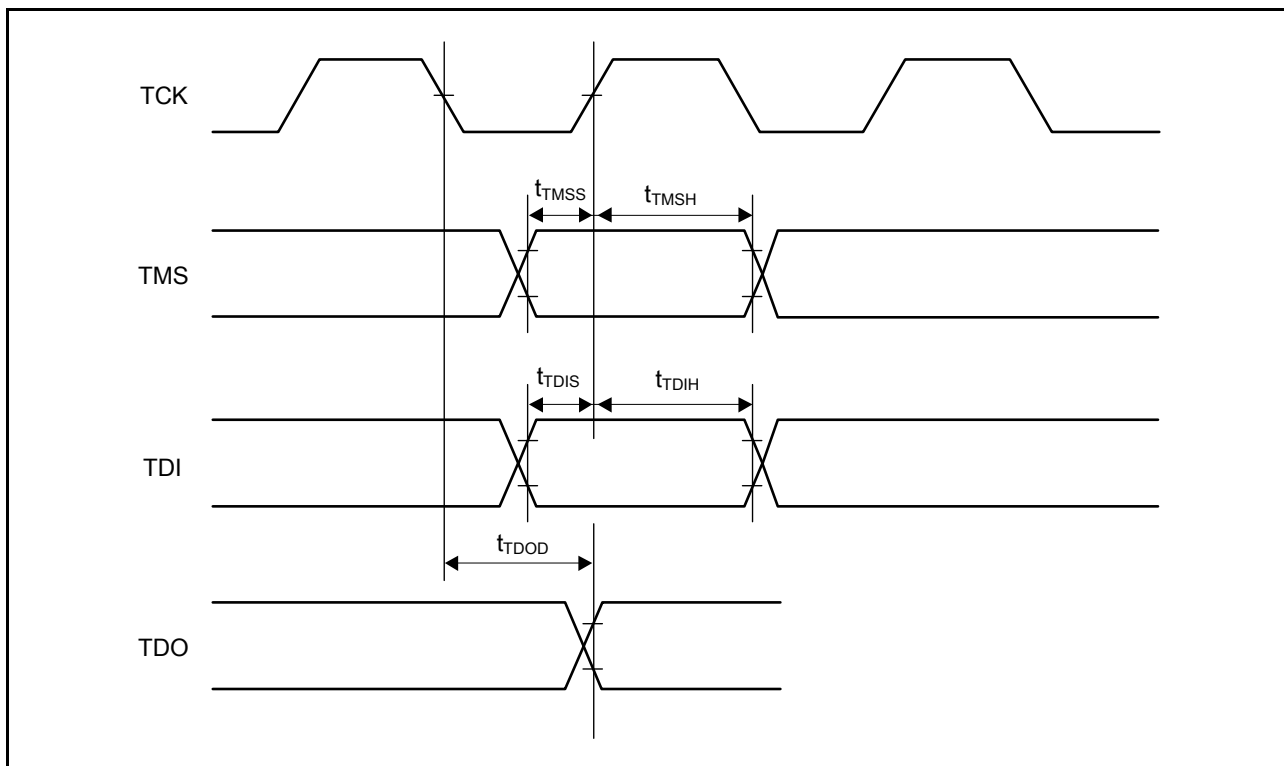


Figure 41.60 Boundary Scan Input/Output Timing

41.4 USB Characteristics

Table 41.19 Internal USB Full-Speed Characteristics (DP, DM Pin Characteristics)

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 3.0 to 3.6V, VREFH = 3.0V to AVCC

VSS = PLLVSS = AVSS = VREFL = VCC_USB = 0V

PCLK = 24 to 50MHz

T_a = -40 to +85°C

Item	Symbol	Min.	Max.	Unit	Test Conditions		
Input characteristics	Input high level voltage	V _{IH}	2.0	—	V	Figure 41.61 and Figure 41.62	
	Input low level voltage	V _{IL}	—	0.8	V		
	Differential input Sensitivity	V _{DI}	0.2	—	V		DP — DM
	Differential common mode range	V _{CM}	0.8	2.5	V		
Output characteristics	Output high level voltage	V _{OH}	2.8	3.6	V	I _{OH} = -200μA	
	Output low level voltage	V _{OL}	0.0	0.3	V	I _{OL} = 2mA	
	Cross over voltage	V _{CRS}	1.3	2.0	V		
	Rising time	t _{Lr}	4	20	ns		
	Falling time	t _{Lf}	4	20	ns		
	Rising/falling time ratio	t _{Lr} / t _{Lf}	90	111.11	%	t _{Lr} / t _{Lf}	
	Output resistance	Z _{DRV}	28	44	Ω	Rs = 22Ω included	

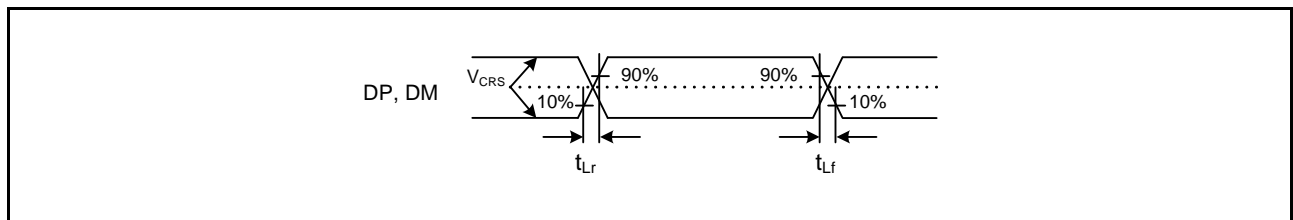


Figure 41.61 DP, DM Output Timing (Full-Speed)

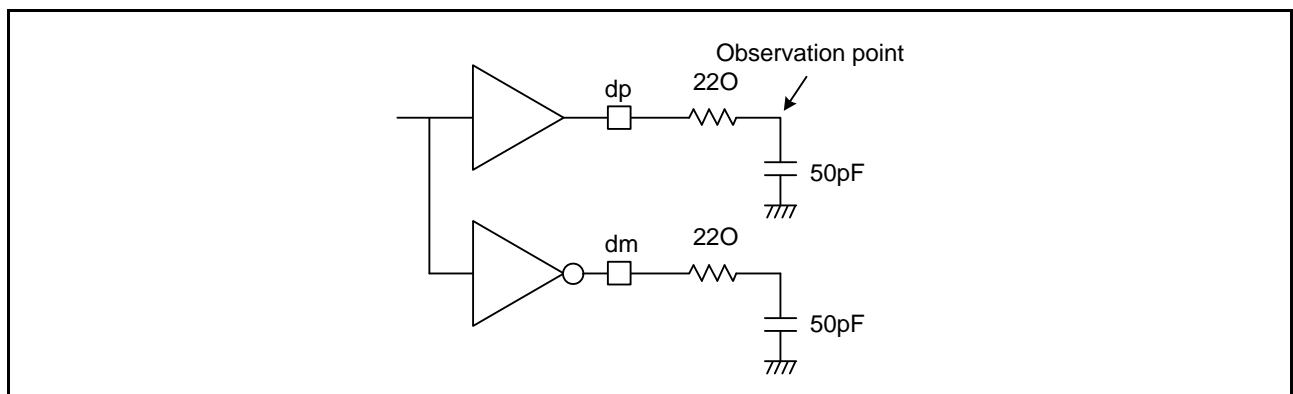


Figure 41.62 Test Circuit (Full-Speed)

41.5 A/D Conversion Characteristics

Table 41.20 10-Bit A/D Conversion Characteristics

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

PCLK = 8 to 50MHz

T_a = -40 to +85°C

Item	Min.	Typ.	Max.	Unit	Test Conditions		
Resolution	10	10	10	bits			
Conversion time*1 (PCLK = 50-MHz operation)	With 0.1-μF external capacitor	When the capacitor is charged enough*2	0.8 (0.3) ^{*3}	—	—	μs	Sampling 15 states
	Without external capacitor	Permissible signal source impedance (max.) = 1.0 kΩ	1.0 (0.5) ^{*3}	—	—		Sampling 25 states
		Permissible signal source impedance (max.) = 5.0 kΩ	2.6 (2.1) ^{*3}	—	—		Sampling 105 states
Analog input capacitance	—	—	6.0	pF			
INL integral nonlinearity error	—	±1.5	±3.0	LSB			
Offset error	—	±1.5	±3.0	LSB			
Full-scale error	—	±1.5	±3.0	LSB			
Quantization error	—	±0.5	—	LSB			
Absolute accuracy	—	±1.5	±3.0	LSB			
DNL differential nonlinearity error	—	±0.5	±1.0	LSB			

Note 1. The conversion time includes the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The scanning is not supported.

Note 3. The value in parentheses indicates the sampling time.

Table 41.21 12-Bit A/D Conversion Characteristics

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

PCLK = 8 to 50MHz

T_a = -40 to +85°C

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	12	12	12	bits	
Conversion time*1	1.0	—	—	μs	AVCC ≥ 3.0
	2.0	—	—	μs	AVCC ≥ 2.7
Analog input capacitance	—	—	30	pF	
Offset error	—	±2.0	±7.5	LSB	
Full-scale error	—	±2.0	±7.5	LSB	
Quantization error	—	±0.5	—	LSB	
Absolute accuracy	—	±2.5	±8.0	LSB	
Nonlinearity error	—	±2.0	±4.0	LSB	

Note 1. The time conversion takes is the sum of the sampling interval and the time comparison takes (permissible signal-source impedance is up to 1.0 kΩ)

41.6 D/A Conversion Characteristics

Table 41.22 D/A Conversion Characteristics

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

T_a = -40 to +85°C

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	10	10	10	bits	
Conversion time	—	—	3.0	μs	20-pF capacitive load
Absolute accuracy	—	±2.0	±4.0	LSB	2-MΩ resistive load
	—	—	±3.0	LSB	4-MΩ resistive load
	—	—	±2.0	LSB	10-MΩ resistive load
RO output resistance	—	3.6	—	kΩ	

41.7 Power-on Reset Circuit, Voltage Detection Circuit Characteristics

Table 41.23 Power-on Reset Circuit, Voltage Detection Circuit Characteristics

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

T_a = -40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Voltage detection level	Power-on reset (POR)	V _{POR}	2.48	2.58	2.68	V	Figure 41.63
	Voltage detection circuit (LVD)	V _{det1}	2.75	2.85	2.95		Figure 41.64 and Figure 41.65
		V _{det2}	3.05	3.15	3.25		
Internal reset time	t _{POR}	20	35	50	ms		
Min. VCC down time*1	t _{VOFF}	200	—	—	μs	Figure 41.64 and Figure 41.65	
Reply delay time	t _{det}	—	—	200	μs		

Note 1. The power-off time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det1}, and V_{det2} for the POR/ LVD.

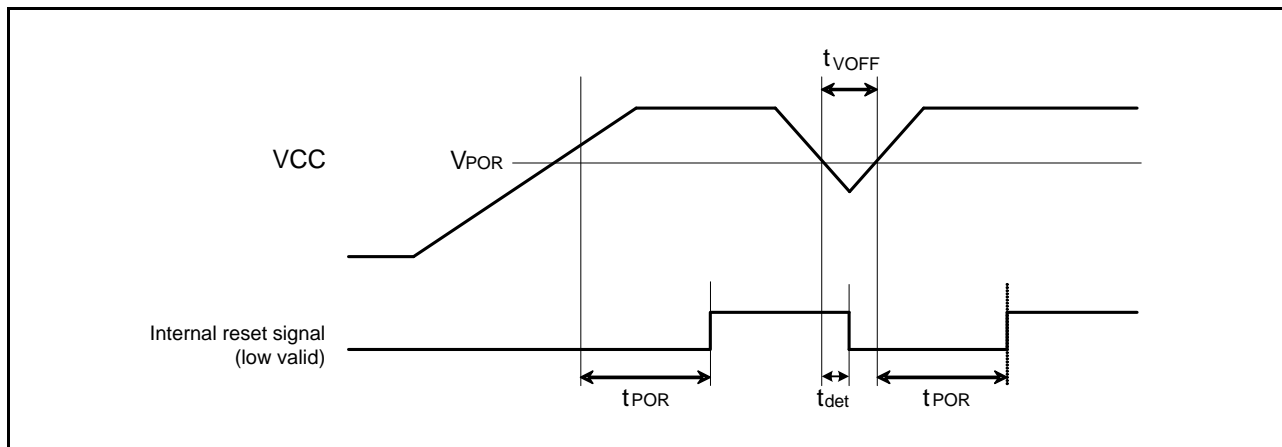


Figure 41.63 Power-on Reset Timing

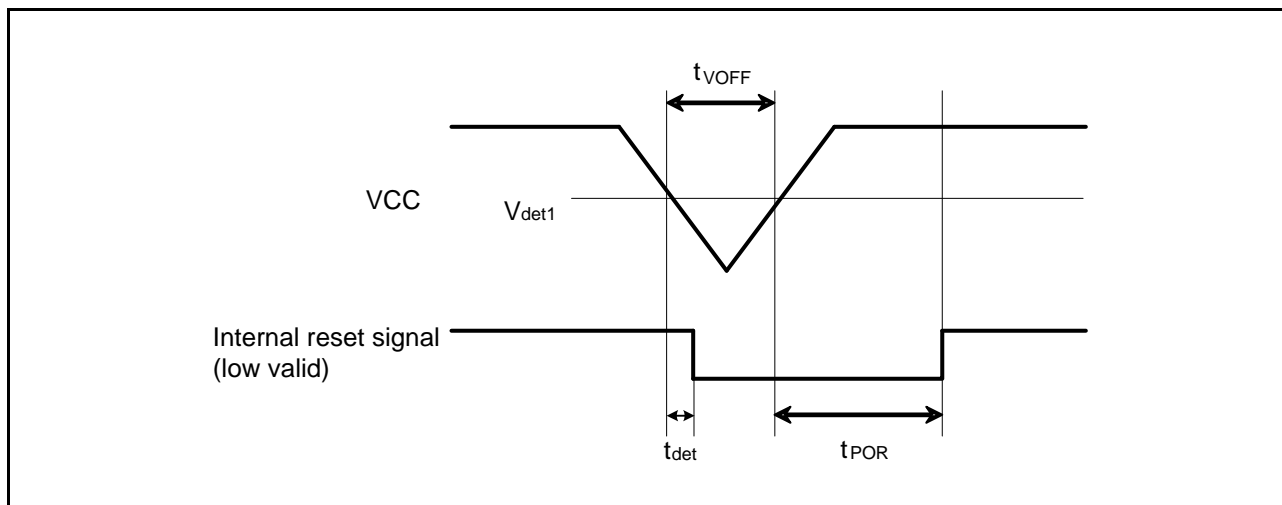


Figure 41.64 Voltage Detection Circuit Timing (Vdet1)

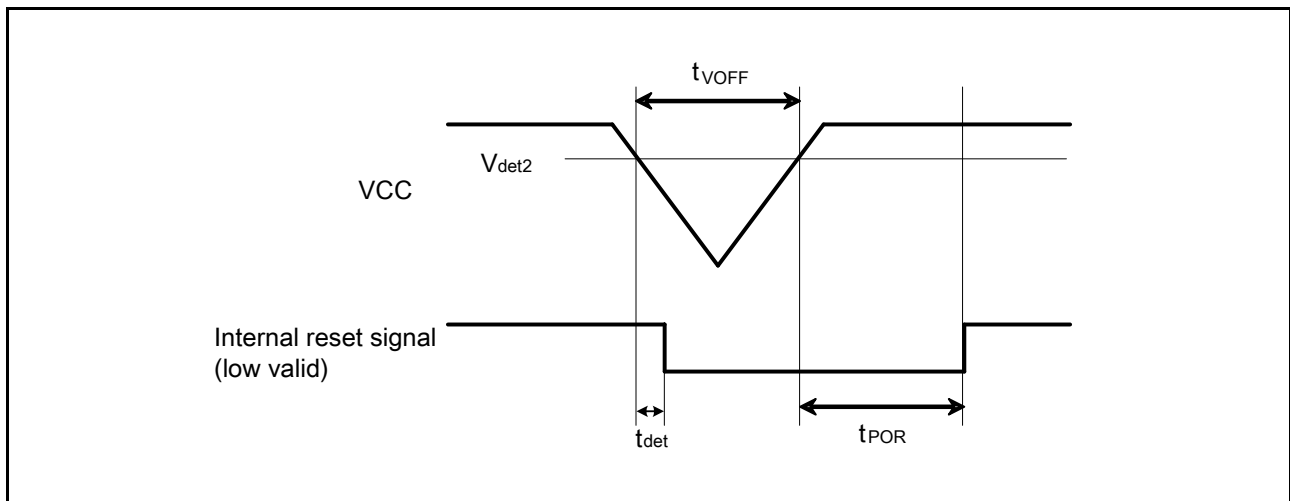


Figure 41.65 Voltage Detection Circuit Timing (Vdet2)

41.8 Oscillation Stop Detection Timing

Table 41.24 Oscillation Stop Detection Circuit Characteristics

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

T_a = -40 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t _{dr}	—	—	1.0	ms	Figure 41.66
Internal oscillation frequency when oscillation stop is detected	f _{MAIN}	0.5	—	7.0	MHz	

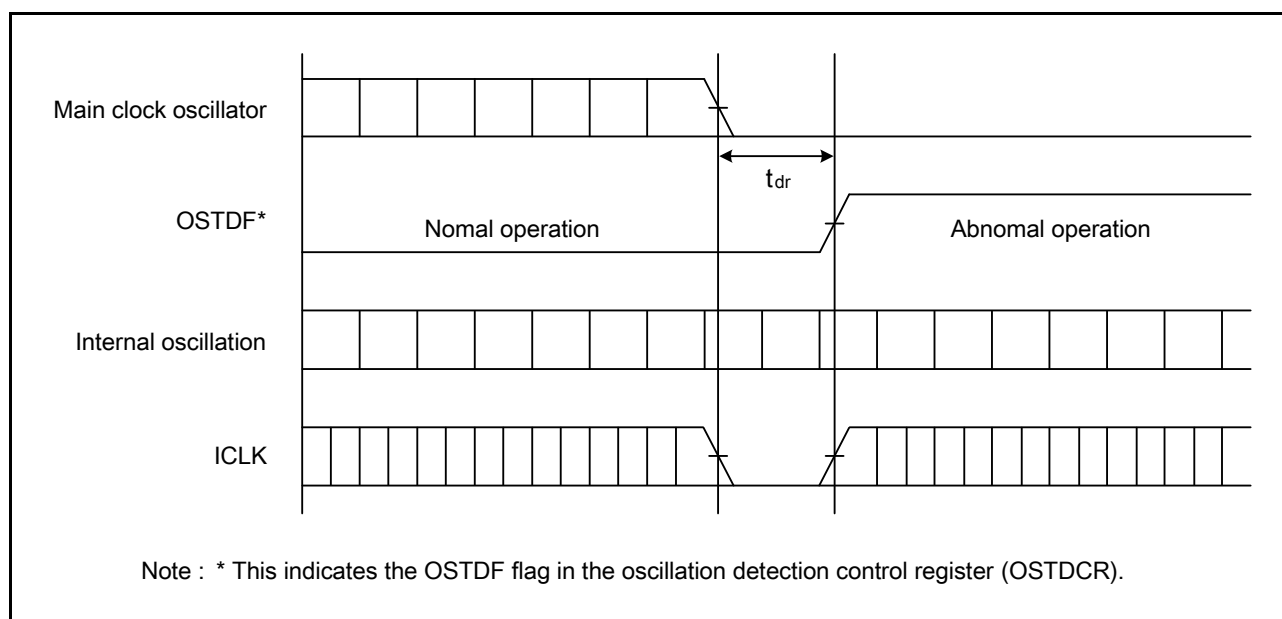


Figure 41.66 Oscillation Stop Detection Timing

41.9 ROM (Flash Memory for Code Storage) Characteristics

Table 41.25 ROM (Flash Memory for Code Storage) Characteristics

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

Temperature range for the programming/erasure operation: T_a = -40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Programming time	256 bytes	t _{P256}	—	2	12	ms	PCLK = 50MHz N _{PEC} ≤ 100	
	4 Kbytes	t _{P4K}	—	23	50	ms		
	16 Kbytes	t _{P16K}	—	90	200	ms		
	Programming time	256 byte	t _{P256}	—	2.4	14.4	ms	PCLK = 50MHz N _{PEC} > 100
		4 Kbytes	t _{P4K}	—	27.6	60	ms	
		16 Kbytes	t _{P16K}	—	108	240	ms	
Erasure time	4 Kbytes	t _{E4K}	—	25	60	ms	PCLK = 50MHz N _{PEC} ≤ 100	
	16 Kbytes	t _{E16K}	—	100	240	ms		
	Erasure time	4 Kbytes	t _{E4K}	—	30	72	ms	PCLK = 50MHz N _{PEC} > 100
		16 Kbytes	t _{E16K}	—	120	288	ms	
Rewrite/erase cycle*1		N _{PEC}	1000*2	—	—	Times		
Suspend delay time during writing		t _{SPD}	—	—	120	μs	Figure 41.67 PCLK = 50-MHz operation	
First suspend delay time during erasing (in suspend priority mode)		t _{SESD1}	—	—	120	μs		
Second suspend delay time during erasing (in suspend priority mode)		t _{SESD2}	—	—	1.7	ms		
Suspend delay time during erasing (in erasure priority mode)		t _{SEED}	—	—	1.7	ms		
Data hold time*3		t _{DRP}	10	—	—	Year		

Note 1. Definition of rewrite/erase cycle:

The rewrite/erase cycle is the number of erasing for each block. When the rewrite/erase cycle is n times (n = 1000), erasing can be performed n times for each block. For instance, when 256-byte writing is performed 16 times for different addresses in 4-Kbyte block and then the entire block is erased, the rewrite/erase cycle is counted as one. However, writing to the same address for several times as one erasing is not enabled (over writing is prohibited).

Note 2. This indicates the minimum number that guarantees the characteristics after rewriting. (The guaranteed value is in the range from one to the minimum number.)

Note 3. This indicates the characteristic when rewrite is performed within the specification range including the minimum number.

41.10 Data Flash (Flash Memory for Data Storage) Characteristics

Table 41.26 Data Flash (Flash Memory for Data Storage) Characteristics

Conditions: VCC = PLLVCC = AVCC = VCC_USB = 2.7 to 3.6V, VREFH = 2.7V to AVCC

VSS = PLLVSS = AVSS = VREFL = VSS_USB = 0V

Temperature range for the programming/erasure operation: T_a = -40 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Programming time	8 bytes	t _{DP8}	—	0.4	2	ms	PCLK = 50-MHz operation
	128 bytes	t _{DP128}	—	1	5	ms	
Erasure time	2 Kbytes	t _{DE2K}	—	70	250	ms	PCLK = 50-MHz operation
Blank check time	8 bytes	t _{DBC8}	—	—	30	μs	PCLK = 50-MHz operation
	2 Kbytes	t _{DBC2K}	—	—	0.7	ms	
Rewrite/erase cycle ^{*1}		N _{DPEC}	30000 ^{*2}	—	—	Times	
Suspend delay time during writing		t _{DSPD}	—	—	120	μs	Figure 41.67 PCLK = 50-MHz operation
First suspend delay time during erasing (in suspend priority mode)		t _{DSESD1}	—	—	120	μs	
Second suspend delay time during erasing (in suspend priority mode)		t _{DSESD2}	—	—	1.7	ms	
Suspend delay time during erasing (in erasure priority mode)		t _{DSEED}	—	—	1.7	ms	
Data hold time ^{*3}		t _{DDRP}	10	—	—	Year	

Note 1. Definition of rewrite/erase cycle:

The rewrite/erase cycle is the number of erasing for each block. When the rewrite/erase cycle is n times (n = 30000), erasing can be performed n times for each block. For instance, when 128-byte writing is performed 16 times for different addresses in 2-Kbyte block and then the entire block is erased, the rewrite/erase cycle is counted as one. However, writing to the same address for several times as one erasing is not enabled (over writing is prohibited).

Note 2. This indicates the minimum number that guarantees the characteristics after rewriting. (The guaranteed value is in the range from one to the minimum number.)

Note 3. This indicates the characteristic when rewrite is performed within the specification range including the minimum number.

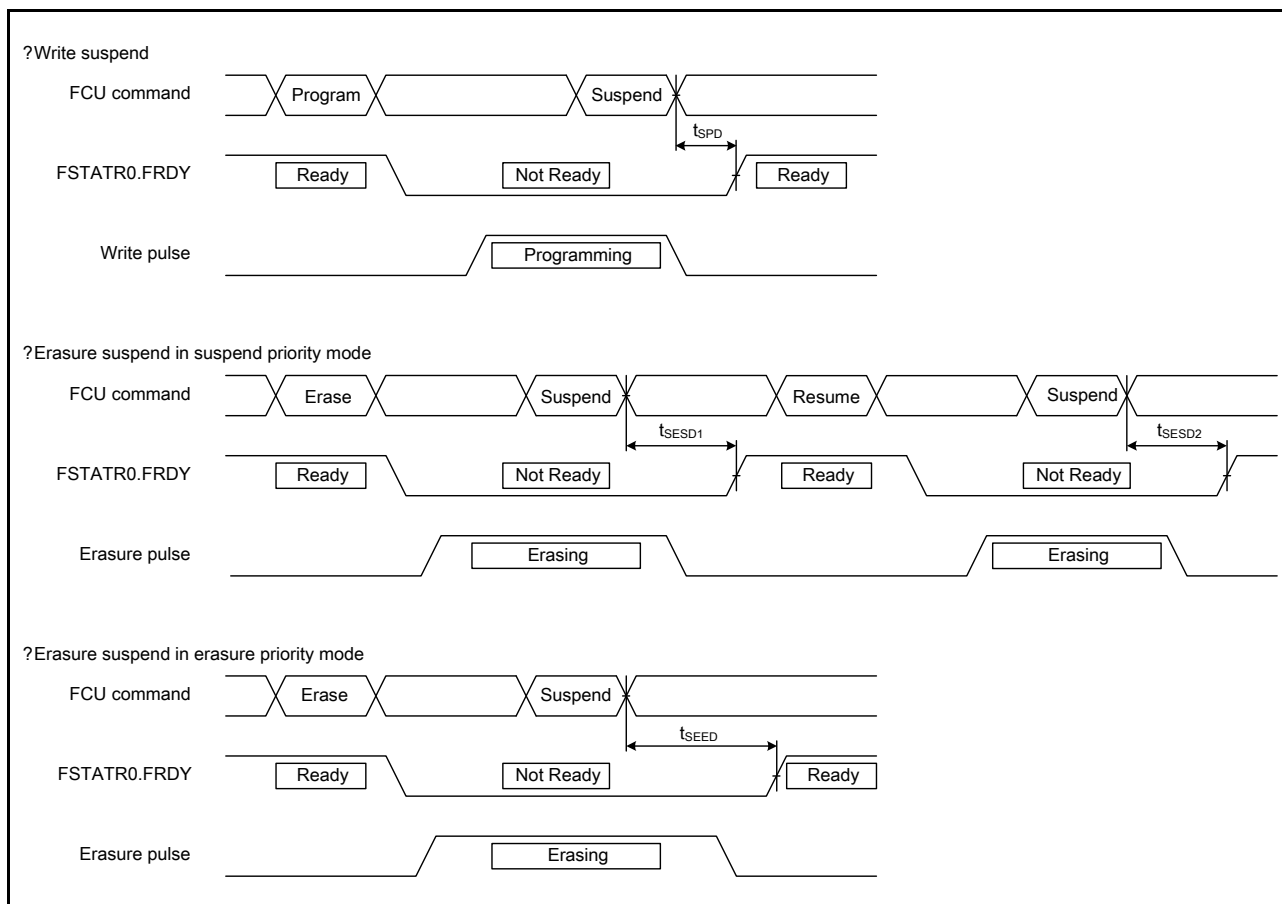


Figure 41.67 Flash Memory Write/Erase Suspend Timing

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing State (1 / 5)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP=1/0	After Deep Software Standby Mode is Canceled (Return of Start-up Mode)	
			OPE=1	OPE=0		IOKEEP=1*1	IOKEEP=0
P00 to P02*7	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P03/IRQ11-A/ DA0*8	All	Hi-Z	[DAOE0=1] DA output retained [DAOE0=0] Keep-O*2		[DAOE0=1] Hi-Z [DAOE0=0] Keep	Keep	Hi-Z
P05/IRQ13-A/ DA1	All	Hi-Z	[DAOE1=1] DA output retained [DAOE1=0] Keep-O*2		[DAOE1=1] Hi-Z [DAOE1=0] Keep	Keep	Hi-Z
P07/ADTRG0- A/IRQ15-A*9	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P10*6 P11*6 P12,P13	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P14,P15*7 P16,P17 *7	All	Hi-Z	Keep-O*2		Keep*3	Keep	Hi-Z
P20 to P23	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
P24 to P27	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[CS# output] H [Other than the above] Keep-O	[CS# output] Hi-Z [Other than the above] Keep-O			
P30 to P33	All	Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
P34/MTIOC0A/ SCK6-B/ TMCI3-B/ IRQ4-A/PO12	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
Port 4	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P50/WR0#/ WR#/SSLB1-A/ TxD2-B	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[WR0#/WR# output] H	[WR0#/WR# output] Hi-Z			
P51/WR1#/ BC1#/ WAIT#-D/ SSLB2-A/ SCK2	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[WR1#/BC1# output] H [Other than the above] Keep-O	[WR1#/BC1# output] Hi-Z [Other than the above] Keep-O			

Table 1.1 Port States in Each Processing State (2 / 5)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP=1/0	After Deep Software Standby Mode is Canceled (Return of Start-up Mode)	
			OPE=1	OPE=0		IOKEEP=1*1	IOKEEP=0
P52/RD#/ SSLB3-A/ RxD2-B	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[RD# output] H	[RD# output] Hi-Z			
P53/BCLK	All	Hi-Z	[Clock output] H [Other than the above] Hi-Z		Keep	Keep	Hi-Z
P54 to P55*9	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
P56/WR2#/ BC2#/ EDACK1-C/ MTIOC3C-B*7	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[WR2#/BC2# output] H [Other than the above] Keep-O	[WR2#/BC2# output] Hi-Z [Other than the above] Keep-O			
P57/WAIT#-A/ WR3#/BC3#/ EDREQ1-C*6	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[WR3#/BC3# output] H [Other than the above] Keep-O	[WR3#/BC3# output] Hi-Z [Other than the above] Keep-O			
P60/CS0#-A*7	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[CS# output] H [Other than the above] Keep-O	[CS# output] Hi-Z [Other than the above] Keep-O			
P61/CS1#-A/ SDCS#*7	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[CS output] H [SDCS# output] SDCS# output retained [Other than the above] Keep-O	[CS output] Hi-Z [SDCS# output] Hi-Z [Other than the above] Keep-O			
P62/CS2#-A/ RAS#*7	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[CS# output] H [RAS# output] RAS# output retained [Other than the above] Keep-O	[CS# output] Hi-Z [RAS# output] Hi-Z [Other than the above] Keep-O			

Table 1.1 Port States in Each Processing State (3 / 5)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP=1/0	After Deep Software Standby Mode is Canceled (Return of Start-up Mode)	
			OPE=1	OPE=0		IOKEEP=1*1	IOKEEP=0
P63/CS3#-A/ CAS#*7	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[CS# output] H [CAS# output] CAS# output retained [Other than the above] Keep-O	[CS# output] Hi-Z [CAS# output] Hi-Z [Other than the above] Keep-O			
P64/CS4#-A/ WE#*7	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[CS# output] H [WE# output] WE# output retained [Other than the above] Keep-O	[CS# output] Hi-Z [WE# output] Hi-Z [Other than the above] Keep-O			
P65/CS5#-A/ CKE*7	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[CS# output] H [CKE output] CKE output retained [Other than the above] Keep-O	[CS# output] Hi-Z [CKE output] Hi-Z [Other than the above] Keep-O			
P66/CS6#-A/ DQM0*7	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[CS# output] H [DQM0 output] DQM0 output retained [Other than the above] Keep-O	[CS# output] Hi-Z [DQM0 output] Hi-Z [Other than the above] Keep-O			
P67/CS7#-A/ DQM1*7	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[CS# output] H [DQM1 output] DQM1 output retained [Other than the above] Keep-O	[CS# output] Hi-Z [DQM1 output] Hi-Z [Other than the above] Keep-O			
P70/SDCLK*7	All	Hi-Z	[Clock output] H [Other than the above] Keep-O		Keep	Keep	Hi-Z

Table 1.1 Port States in Each Processing State (4 / 5)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP=1/0	After Deep Software Standby Mode is Canceled (Return of Start-up Mode)	
			OPE=1	OPE=0		IOKEEP=1*1	IOKEEP=0
P71 to P77*7	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[CS# output] H [Other than the above] Keep-O	[CS# output] Hi-Z [Other than the above] Keep-O			
P80 to P83*7 P84, P85*6	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
P90 to P93*7 P94 to P97*6	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended mode (EXBE=1)		[Address output] Address output retained [Data input/ output] Hi-Z [Other than the above] Keep-O	[Address output] Hi-Z [Data input/ output] Hi-Z [Other than the above] Keep-O			
Port A	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[Address output] Address output retained [Other than the above] Keep-O	[Address output] Hi-Z [Other than the above] Keep-O			
Port B	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[Address output] Address output retained [Other than the above] Keep-O	[Address output] Hi-Z [Other than the above] Keep-O			
PC0 to PC3	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended(EXBE=1)		[Address output] Address output retained [Other than the above] Keep-O	[Address output] Hi-Z [Other than the above] Keep-O			
PC4 to PC7*9	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended(EXBE=1)		[Address output] Address output retained [CS# output] H [Other than the above] Keep-O	[Address output] Hi-Z [CS# output] Hi-Z [Other than the above] Keep-O			

Table 1.1 Port States in Each Processing State (5 / 5)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP=1/0	After Deep Software Standby Mode is Canceled (Return of Start-up Mode)	
			OPE=1	OPE=0		IOKEEP=1*1	IOKEEP=0
Port D	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended(EXBE=1)		Hi-Z				
PE0 to PE4*9	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[Data input/output] Hi-Z [Other than the above] Keep-O				
PE5 to PE7*9	Single-chip mode (EXBE=0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[Data input/output] Hi-Z [Other than the above] Keep-O*2				
PF0/ TxD1-B*6	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
PF1/SCK1-B*6	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
PF2/RxD1-B*6	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
PF3*6	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
PF4*6	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
Port G*6	Single-chip mode (EXBE=0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE=1)		[Data input/output] Hi-Z [Other than the above] Keep-O				
WDTOVF*7	All	WDTOVF output	H		H	H	
USB0_DM	All	Hi-Z	Keep-O*4		Hi-Z*5	Hi-Z	
USB0_DP	All	Hi-Z	Keep-O*4		Hi-Z*5	Hi-Z	
USB1_DM	All	Hi-Z	Keep-O*4		Hi-Z*5	Hi-Z	
USB1_DP	All	Hi-Z	Keep-O*4		Hi-Z*5	Hi-Z	

[Legend]

H: High

L: Low

Keep-O: Output pins retain their previous values, and input pins become high-impedance.

Keep: Pin states are retained during periods on software standby.

Hi-Z: High-impedance

Note 1. Retains the I/O port state until the DPSBYCR.IOKEEP bit is cleared to 0.

Note 2. Input is enabled if the pin is specified as the software standby cancelling source while it is used as an external interrupt pin.

Note 3. Input is enabled if the pin is specified as the deep software standby cancelling source.

Note 4. Input is enabled while the pin is used as an input pin.

Note 5. Input is enabled if the pin is specified as the deep software standby cancelling source.

Note 6. This pin is not available in the 145-pin TFLGA, 144-pin LQFP, 100-pin LQFP, and 85-pin TFLGA packages.

Note 7. This pin is not available in the 100-pin LQFP and 85-pin TFLGA packages.

Note 8. This pin is not available in the 100-pin LQFP package.

Note 9. This pin is not available in the 85-pin TFLGA package.

Table 1.2 Conditions when the Port State is in High-Impedance by the POE Function Control [176-Pin LFBGA/145-Pin TFLGA/144-Pin LQFP]

Port Name Pin Name	Register Setting Control by Software	Control by the POEn# Pin (n = 0 to 9)	Output Level Comparison with the Complementary PWM Output Pin	Control by the Oscillation Stop Detection
P15/MTIOC0B	When POECR1.PE1ZE=1 and SPOER.CH0HIZ=1	When POECR1.PE1ZE=1 and ICSR3.POE8F=1 and ICSR3.POE8E=1	—	When POECR1.PE1ZE=1 and NMISR.OSTST=1
P22/MTIOC3B-A	When POECR2.P1CZEA=1 and SPOER.CH34HIZ=1	When POECR2.P1CZEA=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P1CZEA=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P1CZEA=1 and NMISR.OSTST=1
P23/MTIOC3D-A	When POECR2.P1CZEA=1 and SPOER.CH34HIZ=1	When POECR2.P1CZEA=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P1CZEA=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P1CZEA=1 and NMISR.OSTST=1
P24/MTIOC4A-A	When POECR2.P2CZEA=1 and SPOER.CH34HIZ=1	When POECR2.P2CZEA=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P2CZEA=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P2CZEA=1 and NMISR.OSTST=1
P25/MTIOC4C-A	When POECR2.P2CZEA=1 and SPOER.CH34HIZ=1	When POECR2.P2CZEA=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P2CZEA=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P2CZEA=1 and NMISR.OSTST=1
P30/MTIOC4B-A	When POECR2.P3CZEA=1 and SPOER.CH34HIZ=1	When POECR2.P3CZEA=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P3CZEA=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P3CZEA=1 and NMISR.OSTST=1
P31/MTIOC4D-A	When POECR2.P3CZEA=1 and SPOER.CH34HIZ=1	When POECR2.P3CZEA=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P3CZEA=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P3CZEA=1 and NMISR.OSTST=1
P32/MTIOC0C	When POECR1.PE2ZE=1 and SPOER.CH0HIZ=1	When POECR1.PE2ZE=1 and ICSR3.POE8F=1 and ICSR3.POE8E=1	—	When POECR1.PE2ZE=1 and NMISR.OSTST=1
P33/MTIOC0D	When POECR1.PE3ZE=1 and SPOER.CH0HIZ=1	When POECR1.PE3ZE=1 and ICSR3.POE8F=1 and ICSR3.POE8E=1	—	When POECR1.PE3ZE=1 and NMISR.OSTST=1
P34/MTIOC0A	When POECR1.PE0ZE=1 and SPOER.CH0HIZ=1	When POECR1.PE0ZE=1 and ICSR3.POE8F=1 and ICSR3.POE8E=1	—	When POECR1.PE0ZE=1 and NMISR.OSTST=1
P54/MTIOC4B-B	When POECR2.P3CZEB=1 and SPOER.CH34HIZ=1	When POECR2.P3CZEB=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P3CZEB=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P3CZEB=1 and NMISR.OSTST=1
P55/MTIOC4D-B	When POECR2.P3CZEB=1 and SPOER.CH34HIZ=1	When POECR2.P3CZEB=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P3CZEB=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P3CZEB=1 and NMISR.OSTST=1
P80/MTIOC3B-B	When POECR2.P1CZEB=1 and SPOER.CH34HIZ=1	When POECR2.P1CZEB=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P1CZEB=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P1CZEB=1 and NMISR.OSTST=1
P81/MTIOC3D-B	When POECR2.P1CZEB=1 and SPOER.CH34HIZ=1	When POECR2.P1CZEB=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P1CZEB=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P1CZEB=1 and NMISR.OSTST=1

P82/MTIOC4A-B	When POE2R.P2CZEB=1 and SPOER.CH34HIZ=1	When POE2R.P2CZEB=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POE2R.P2CZEB=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POE2R.P2CZEB=1 and NMISR.OSTST=1
P83/MTIOC4C-B	When POE2R.P2CZEB=1 and SPOER.CH34HIZ=1	When POE2R.P2CZEB=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POE2R.P2CZEB=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POE2R.P2CZEB=1 and NMISR.OSTST=1
PA0/MTIOC6A	When POE1R.PE4ZE=1 and SPOER.CH6HIZ=1	When POE1R.PE4ZE=1 and ICSR4.POE9F=1 and ICSR4.POE9E=1	—	When POE1R.PE4ZE=1 and NMISR.OSTST=1
PA1/MTIOC6B	When POE1R.PE5ZE=1 and SPOER.CH6HIZ=1	When POE1R.PE5ZE=1 and ICSR4.POE9F=1 and ICSR4.POE9E=1	—	When POE1R.PE5ZE=1 and NMISR.OSTST=1
PA2/MTIOC6C	When POE1R.PE6ZE=1 and SPOER.CH6HIZ=1	When POE1R.PE6ZE=1 and ICSR4.POE9F=1 and ICSR4.POE9E=1	—	When POE1R.PE6ZE=1 and NMISR.OSTST=1
PA3/MTIOC6D	When POE1R.PE7ZE=1 and SPOER.CH6HIZ=1	When POE1R.PE7ZE=1 and ICSR4.POE9F=1 and ICSR4.POE9E=1	—	When POE1R.PE7ZE=1 and NMISR.OSTST=1
PB2/MTIOC9B	When POE2R.P4CZE=1 and SPOER.CH910HIZ=1	When POE2R.P4CZE=1 and ICSR2.POE4F, POE5F, POE6F, or POE7F=1	When POE2R.P4CZE=1 and OCSR2.OSF2=1 and OCSR2.OCE2=1	When POE2R.P4CZE=1 and NMISR.OSTST=1
PB3/MTIOC9D	When POE2R.P4CZE=1 and SPOER.CH910HIZ=1	When POE2R.P4CZE=1 and ICSR2.POE4F, POE5F, POE6F, or POE7F=1	When POE2R.P4CZE=1 and OCSR2.OSF2=1 and OCSR2.OCE2=1	When POE2R.P4CZE=1 and NMISR.OSTST=1
PB4/MTIOC10A	When POE2R.P5CZE=1 and SPOER.CH910HIZ=1	When POE2R.P5CZE=1 and ICSR2.POE4F, POE5F, POE6F, or POE7F=1	When POE2R.P5CZE=1 and OCSR2.OSF2=1 and OCSR2.OCE2=1	When POE2R.P5CZE=1 and NMISR.OSTST=1
PB5/MTIOC10C	When POE2R.P5CZE=1 and SPOER.CH910HIZ=1	When POE2R.P5CZE=1 and ICSR2.POE4F, POE5F, POE6F, or POE7F=1	When POE2R.P5CZE=1 and OCSR2.OSF2=1 and OCSR2.OCE2=1	When POE2R.P5CZE=1 and NMISR.OSTST=1
PB6/MTIOC10B	When POE2R.P6CZE=1 and SPOER.CH910HIZ=1	When POE2R.P6CZE=1 and ICSR2.POE4F, POE5F, POE6F, or POE7F=1	When POE2R.P6CZE=1 and OCSR2.OSF2=1 and OCSR2.OCE2=1	When POE2R.P6CZE=1 and NMISR.OSTST=1
PB7/MTIOC10D	When POE2R.P6CZE=1 and SPOER.CH910HIZ=1	When POE2R.P6CZE=1 and ICSR2.POE4F, POE5F, POE6F, or POE7F=1	When POE2R.P6CZE=1 and OCSR2.OSF2=1 and OCSR2.OCE2=1	When POE2R.P6CZE=1 and NMISR.OSTST=1

Table 1.3 Conditions when the Port State is in High-Impedance by the POE Function Control [100-Pin LQFP]

Port Name Pin Name	Register Setting Control by Softwre	Control by the POEn# Pin (n = 0 to 9)	Output Level Comparison with the Complementary PWM Output Pin	Control by the Oscillation Stop Detection
P13/MTIOC0B	When POECR1.PE1ZE=1 and SPOER.CH0HIZ=1	When POECR1.PE1ZE=1 and ICSR3.POE8F=1 and ICSR3.POE8E=1	—	When POECR1.PE1ZE=1 and NMISR.OSTST=1
P22/MTIOC3B	When POECR2.P1CZEA=1 and SPOER.CH34HIZ=1	When POECR2.P1CZEA=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P1CZEA=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P1CZEA=1 and NMISR.OSTST=1
P23/MTIOC3D	When POECR2.P1CZEA=1 and SPOER.CH34HIZ=1	When POECR2.P1CZEA=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P1CZEA=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P1CZEA=1 and NMISR.OSTST=1
P24/MTIOC4A	When POECR2.P2CZEA=1 and SPOER.CH34HIZ=1	When POECR2.P2CZEA=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P2CZEA=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P2CZEA=1 and NMISR.OSTST=1
P25/MTIOC4C	When POECR2.P2CZEA=1 and SPOER.CH34HIZ=1	When POECR2.P2CZEA=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P2CZEA=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P2CZEA=1 and NMISR.OSTST=1
P30/MTIOC4B-A	When POECR2.P3CZEA=1 and SPOER.CH34HIZ=1	When POECR2.P3CZEA=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P3CZEA=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P3CZEA=1 and NMISR.OSTST=1
P31/MTIOC4D-A	When POECR2.P3CZEA=1 and SPOER.CH34HIZ=1	When POECR2.P3CZEA=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P3CZEA=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P3CZEA=1 and NMISR.OSTST=1
P32/MTIOC0C	When POECR1.PE2ZE=1 and SPOER.CH0HIZ=1	When POECR1.PE2ZE=1 and ICSR3.POE8F=1 and ICSR3.POE8E=1	—	When POECR1.PE2ZE=1 and NMISR.OSTST=1
P33/MTIOC0D	When POECR1.PE3ZE=1 and SPOER.CH0HIZ=1	When POECR1.PE3ZE=1 and ICSR3.POE8F=1 and ICSR3.POE8E=1	—	When POECR1.PE3ZE=1 and NMISR.OSTST=1
P34/MTIOC0A	When POECR1.PE0ZE=1 and SPOER.CH0HIZ=1	When POECR1.PE0ZE=1 and ICSR3.POE8F=1 and ICSR3.POE8E=1	—	When POECR1.PE0ZE=1 and NMISR.OSTST=1
P54/MTIOC4B-B	When POECR2.P3CZEB=1 and SPOER.CH34HIZ=1	When POECR2.P3CZEB=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P3CZEB=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P3CZEB=1 and NMISR.OSTST=1
P55/MTIOC4D-B	When POECR2.P3CZEB=1 and SPOER.CH34HIZ=1	When POECR2.P3CZEB=1 and ICSR1.POE0F, POE1F, POE2F, or POE3F=1	When POECR2.P3CZEB=1 and OCSR1.OSF1=1 and OCSR1.OCE1=1	When POECR2.P3CZEB=1 and NMISR.OSTST=1
PA0/MTIOC6A	When POECR1.PE4ZE=1 and SPOER.CH6HIZ=1	When POECR1.PE4ZE=1 and ICSR4.POE9F=1 and ICSR4.POE9E=1	—	When POECR1.PE4ZE=1 and NMISR.OSTST=1
PA1/MTIOC6B	When POECR1.PE5ZE=1 and SPOER.CH6HIZ=1	When POECR1.PE5ZE=1 and ICSR4.POE9F=1 and ICSR4.POE9E=1	—	When POECR1.PE5ZE=1 and NMISR.OSTST=1
PA2/MTIOC6C	When POECR1.PE6ZE=1 and SPOER.CH6HIZ=1	When POECR1.PE6ZE=1 and ICSR4.POE9F=1 and ICSR4.POE9E=1	—	When POECR1.PE6ZE=1 and NMISR.OSTST=1

PA3/MTIOC6D	When POECR1.PE7ZE=1 and SPOER.CH6HIZ=1	When POECR1.PE7ZE=1 and ICSR4.POE9F=1 and ICSR4.POE9E=1	—	When POECR1.PE7ZE=1 and NMISR.OSTST=1
PB2/MTIOC9B	When POECR2.P4CZE=1 and SPOER.CH910HIZ=1	When POECR2.P4CZE=1 and ICSR2.POE4F, POE5F, POE6F, or POE7F=1	When POECR2.P4CZE=1 and OCSR2.OSF2=1 and OCSR2.OCE2=1	When POECR2.P4CZE=1 and NMISR.OSTST=1
PB3/MTIOC9D	When POECR2.P4CZE=1 and SPOER.CH910HIZ=1	When POECR2.P4CZE=1 and ICSR2.POE4F, POE5F, POE6F, or POE7F=1	When POECR2.P4CZE=1 and OCSR2.OSF2=1 and OCSR2.OCE2=1	When POECR2.P4CZE=1 and NMISR.OSTST=1
PB4/MTIOC10A	When POECR2.P5CZE=1 and SPOER.CH910HIZ=1	When POECR2.P5CZE=1 and ICSR2.POE4F, POE5F, POE6F, or POE7F=1	When POECR2.P5CZE=1 and OCSR2.OSF2=1 and OCSR2.OCE2=1	When POECR2.P5CZE=1 and NMISR.OSTST=1
PB5/MTIOC10C	When POECR2.P5CZE=1 and SPOER.CH910HIZ=1	When POECR2.P5CZE=1 and ICSR2.POE4F, POE5F, POE6F, or POE7F=1	When POECR2.P5CZE=1 and OCSR2.OSF2=1 and OCSR2.OCE2=1	When POECR2.P5CZE=1 and NMISR.OSTST=1
PB6/MTIOC10B	When POECR2.P6CZE=1 and SPOER.CH910HIZ=1	When POECR2.P6CZE=1 and ICSR2.POE4F, POE5F, POE6F, or POE7F=1	When POECR2.P6CZE=1 and OCSR2.OSF2=1 and OCSR2.OCE2=1	When POECR2.P6CZE=1 and NMISR.OSTST=1
PB7/MTIOC10D	When POECR2.P6CZE=1 and SPOER.CH910HIZ=1	When POECR2.P6CZE=1 and ICSR2.POE4F, POE5F, POE6F, or POE7F=1	When POECR2.P6CZE=1 and OCSR2.OSF2=1 and OCSR2.OCE2=1	When POECR2.P6CZE=1 and NMISR.OSTST=1

Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings has been displayed in "Packages" on Renesas Electronics Corp website.

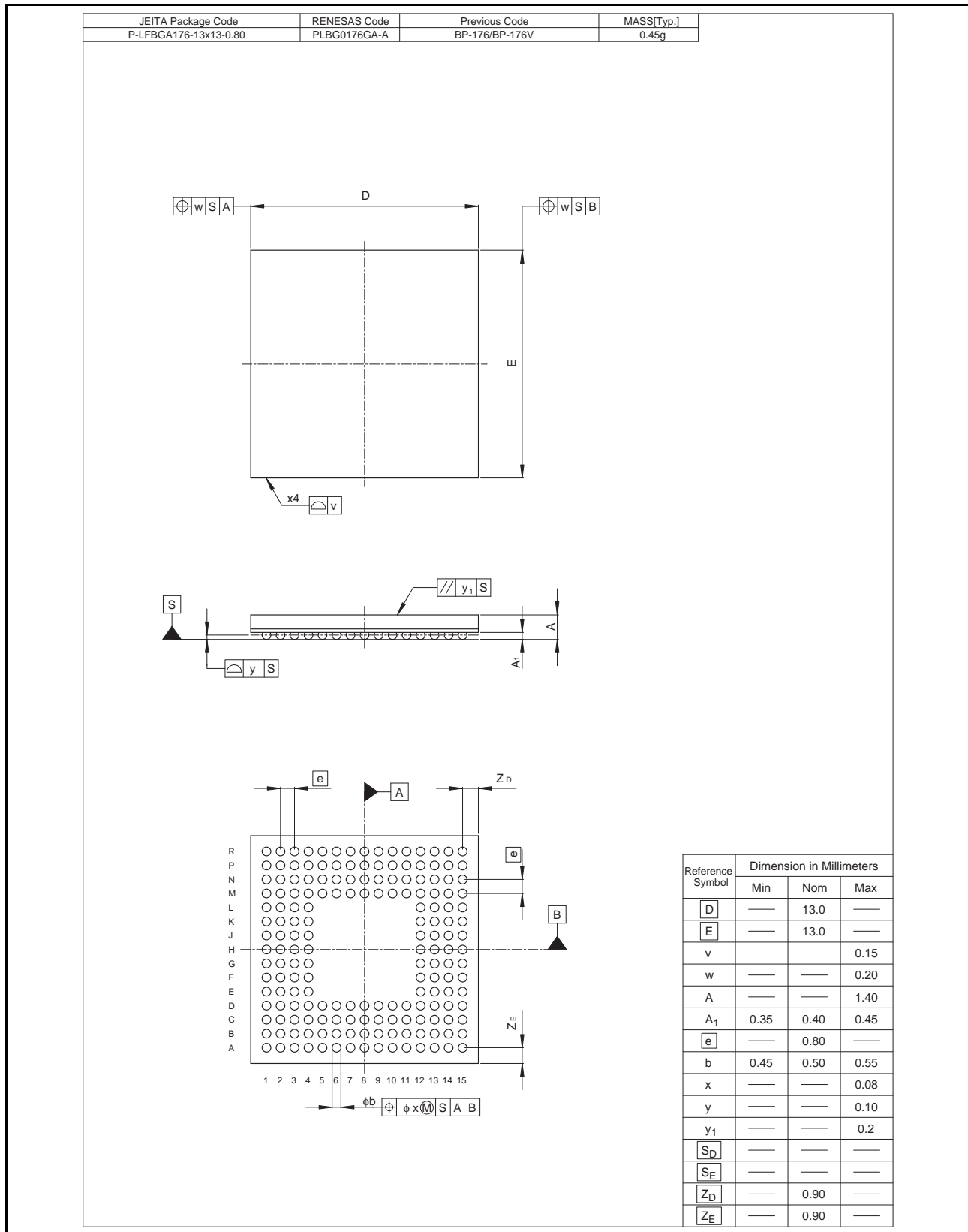


Figure A 176-Pin LFBGA (PLBG0176GA-A) Package Dimensions

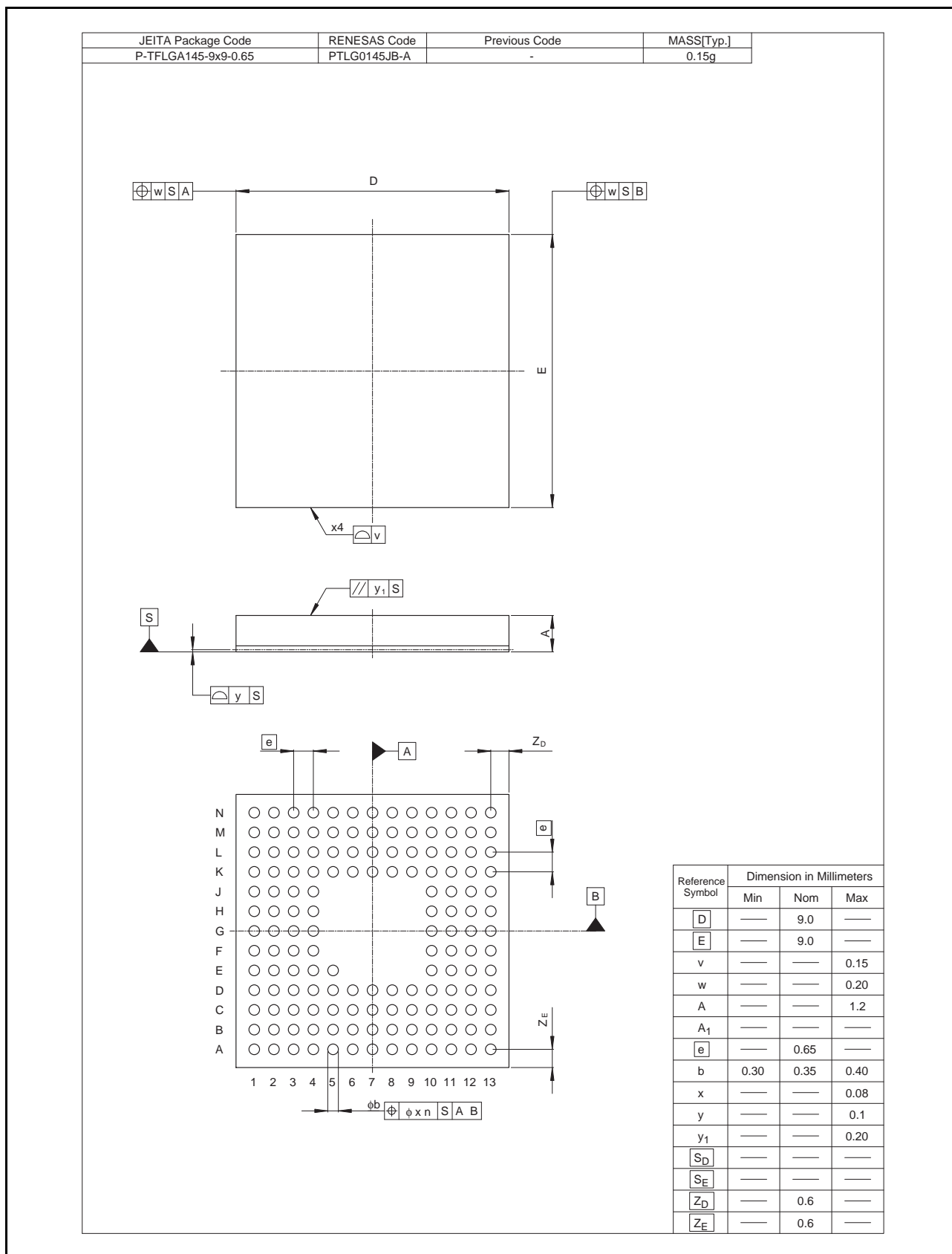


Figure B 145-Pin TFLGA (PTLG0145JB-A) Package Dimensions

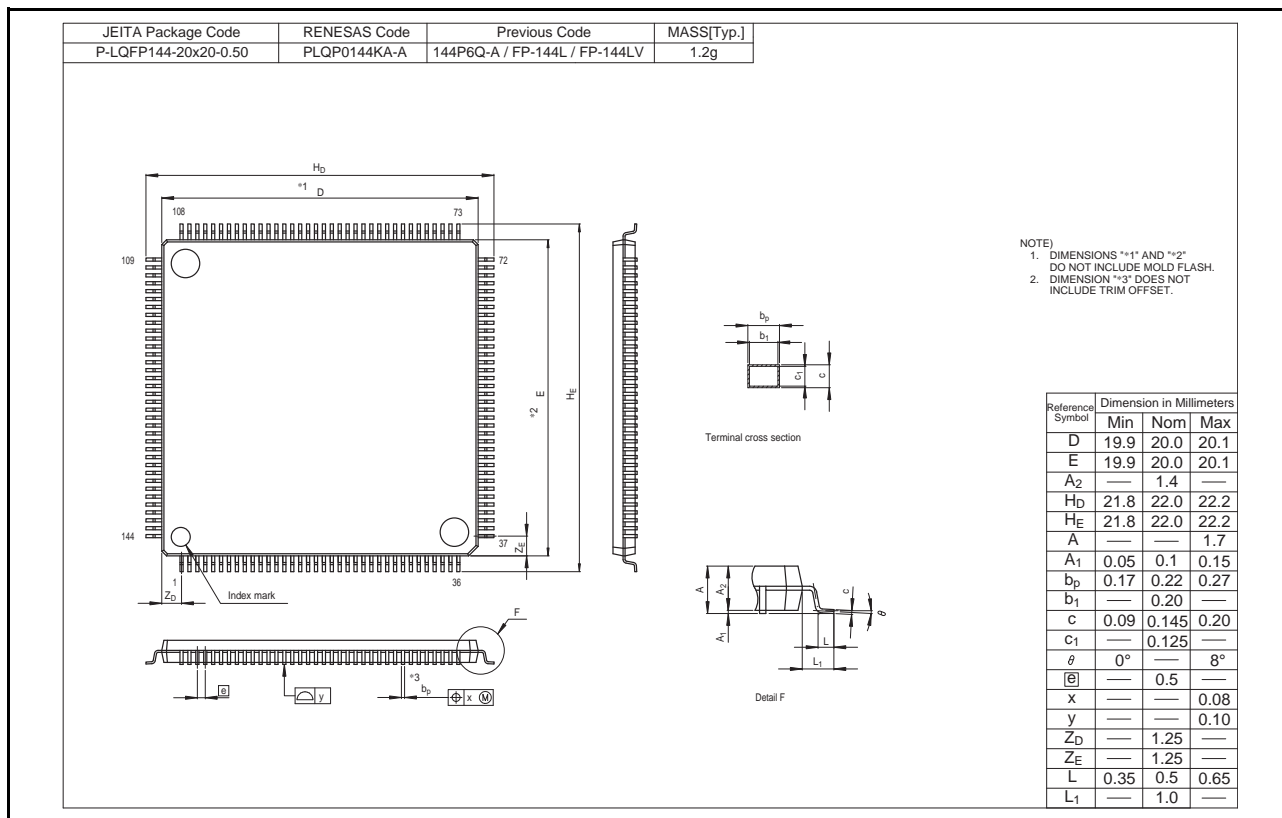


Figure C 144-Pin LQFP (PLQP0144KA-A) Package Dimensions

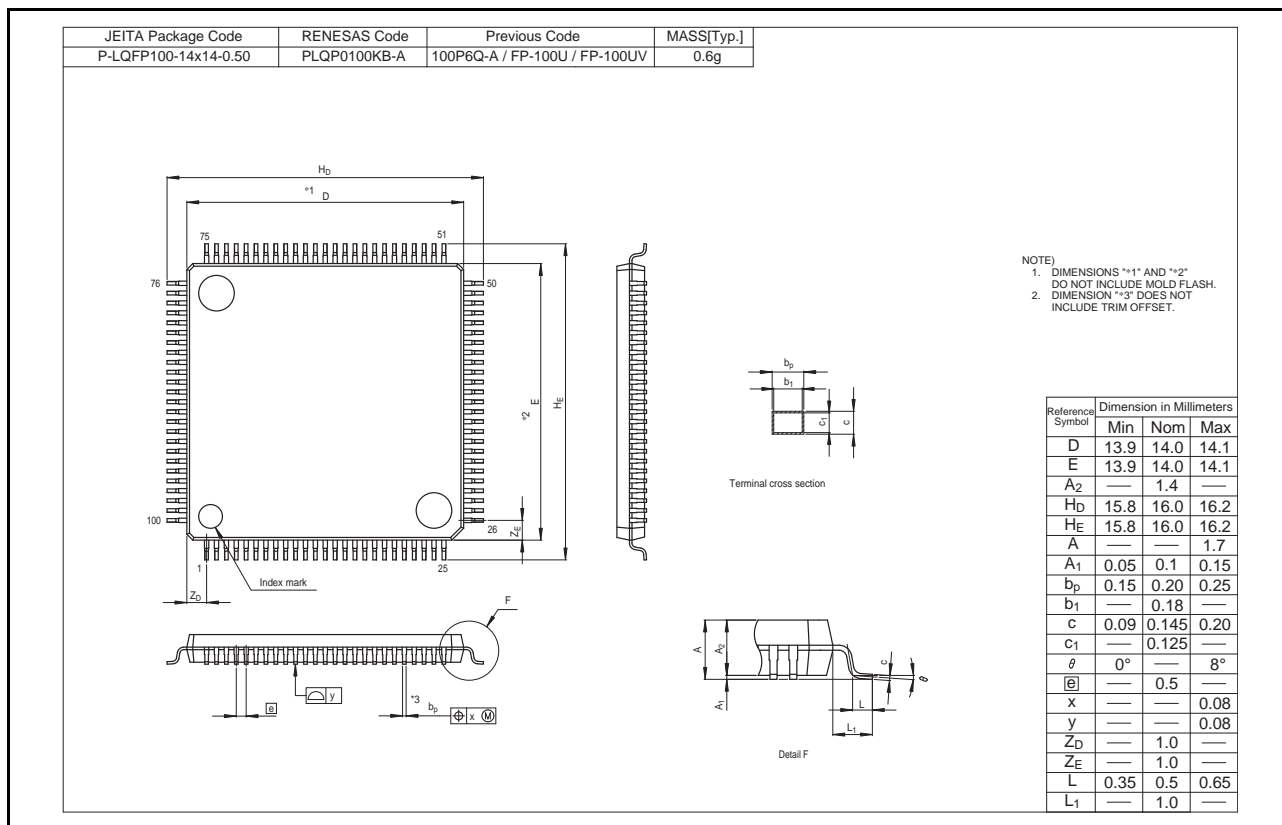


Figure D 100-Pin LQFP (PLQP0100KB-A) Package Dimensions

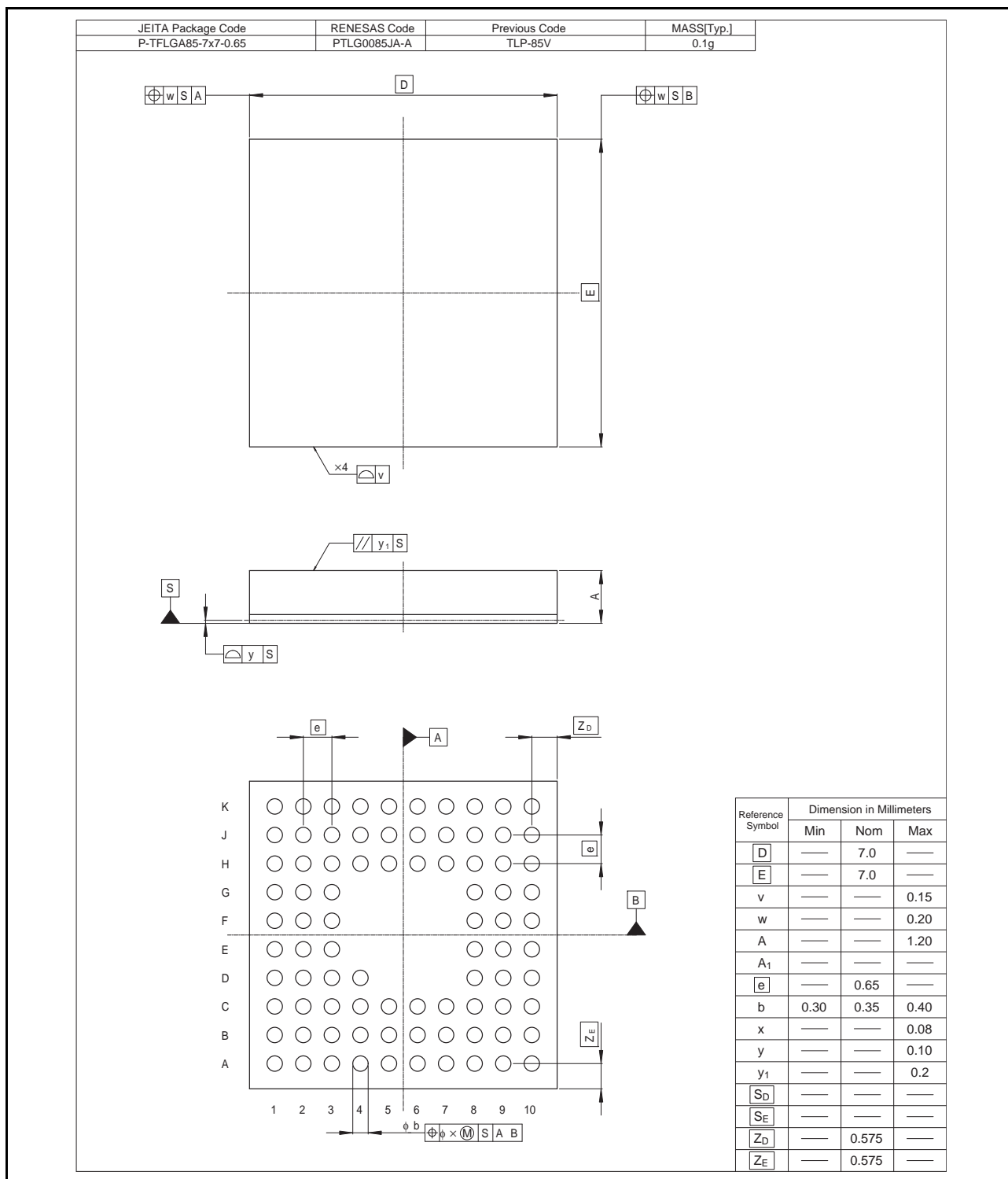


Figure E 85-Pin TFLGA (PTLG0085JA-A) Package Dimensions

REVISION HISTORY

RX62N Group, RX621 Group User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
0.50	Feb. 23, 2010	—	First Edition issued
<u>The underlined portion</u> indicates the additional usage notes.			
1.00	Sep. 07, 2010	All	Description of reserved bits, changed For details, see 2. Description of Registers, in "How to Use This Manual"
		All	Name of the subclock generation circuit I/O pins, changed: OSC1, OSC2 → XCOUT, XCIN
		All	Name of the on-chip emulator pin, changed: TRSYNC# → TRSYNC
			Section 1. Overview
		43	1.1 Feature: Description changed
		43	1.1.1 Applications: Description changed
		44	Table 1.1 Outline of Specification, Multi-function timer pulse unit: Description changed
		47	Table 1.1 Outline of Specification, 8-bit timers: Description changed
		48	Table 1.1 Outline of Specification: Power supply voltage and operating temperature, changed, description on the supply current, deleted
		50	Table 1.2 Functions of RX62N Group and RX621 Group Products: Note changed
		—	Table 1.3 List of Products (Products with Standard Spec.), deleted
		51	Table 1.3 List of Products: Title changed
		55	Figure 1.4 Pin Assignment of the 145-Pin TFLGA, added
		60	Figure 1.9 Pin Assignment of the 85-Pin TFLGA, added
		61 to 68	Table 1.4 List of Pins and Pin Functions (176-Pin LFBGA), changed
		69 to 75	Table 1.5 List of Pins and Pin Functions (145-Pin TFLGA): Pin no. changed
		76 to 82	Table 1.6 List of Pins and Pin Functions (144-Pin LQFP), changed
		83 to 87	Table 1.7 List of Pins and Pin Functions (100-Pin LQFP), changed
		88 to 91	Table 1.8 List of Pins and Pin Functions (85-Pin TFLGA): Pin no. changed
		92	Table 1.9 Pin Functions: Clock, system control, and on-chip emulator, changed
		94	Table 1.9 Pin Functions, Interrupt: Functional description changed
		95	Table 1.9 Pin Functions, Multi-function timer pulse unit: Pin name changed
			Section 2. CPU
		100	Opening description, changed
		100	2.1 Features: Description changed
		101	Figure 2.1 Register Set of the CPU: Note 1 changed
		103	2.2.2.2 Interrupt Table Register (INTB): Description added
		103	2.2.2.3 Program Counter (PC): Value after reset, changed
		107	2.2.2.5 Backup PC (BPC): Description changed
		107	2.2.2.6 Backup PSW (BPSW): Description changed
108 to 110	2.2.2.8 Floating-Point Status Word (FPSW): Bit table, Note 1., and Bit description, changed		
112	2.3 Processor Mode: Description changed		
112	2.3.4 Switching Between Processor Modes: Description changed		
114	2.4.2 Floating-Point: Description changed		
115	2.5.1 Switching the Endian: Description changed		
119	2.5.3 Notes on Access to I/O Registers: Description changed		
121	2.6 Vector Table: Description changed		
122	2.6.2 Relocatable Vector Table: Description changed		
123	2.7.1 Data Prefetching by the RMPA Instruction and the String-Manipulation Instructions: Description changed		

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	126	Table 2.13 Instructions that are Converted into a Single Micro-Operation, changed
		128	2.8.2.2 Instructions Converted into Multiple Micro-Operations and Pipeline Processing: Description deleted
		128, 129	Table 2.14 Instructions that are Converted into Multiple Micro-Operations, changed
		—	2.8.2.3 Pipeline Basic Operation: Description deleted
		131	2.8.2.3 Pipeline Basic Operation, (2) Pipeline Flow with no Stall, (b) When WB stages for the memory load and for the operation are overlapped: Description changed
		134	Table 2.15 Numbers of Cycles for Response to Interrupts, changed
		134	2.8.4 Numbers of Cycles for Response to Interrupts: Description changed
			Section 3. Operating Modes
		135	3.1 Operating Mode Types and Selection: Description changed
		135	Table 3.1 Selection of Operating Modes by the Mode Pins, changed
		135	Table 3.2 Selection of Operating Modes by Register Setting, changed
		136	3.2.1 Mode Monitor Register (MDMONR): Bit table, Note 1., and Bit description, changed
		137	3.2.2 Mode Status Register (MDSR): Bit table and description, changed
		138	3.2.3 System Control Register 0 (SYSCR0): Bit table and description, changed, some description, deleted
		140	3.2.4 System Control Register 1 (SYSCR1): Bit table and description, changed
		141	3.3.1 Single-Chip Mode: Description changed
		141	3.3.2 On-Chip ROM Enabled Extended Mode: Description changed
		141	3.3.3 On-Chip ROM Disabled Extended Mode: Description changed
		142	3.4 Transitions of Operating Modes: Title changed
		142	3.4.1 Operating Mode Transitions According to Mode Pin Setting: Description changed
		142	Figure 3.1 Setting of Pins MD1 and MD0 and Operating Modes, changed
		143	3.4.2 Operating Mode Transitions According to Register Setting: Description changed
		143	Figure 3.2 Setting of Bits ROME and EXBE and Operating Modes, changed
			Section 4. Address Space
		144	Figure 4.1 Memory Map in Each Operating Mode, changed
			Section 5. I/O Registers
		147	4. Number of Access Cycles to I/O Registers, added
		148 to 178	Table 5.1 List of I/O Registers (Address Order), changed, Number of Access States, added
		178	Notes added
		179 to 228	5.2 I/O Register Bits, changed
		228	Notes added
			Section 6. Resets
		229	Table 6.1 Reset Names and Sources, changed
		230	Figure 6.1 Block Diagram of Reset Circuit, changed
		233	6.3.1 Pin Reset: Description changed
		233	6.3.2 Power-On Reset: Description changed
		234	6.3.3 Voltage-Monitoring Reset: Description changed
		234	6.3.4 Deep Software Standby Reset: Description changed
		234	6.3.6 Watchdog Timer Reset: Description changed
		235	6.4 Determination of Reset Generation Source: Description changed
		235	Figure 6.3 Example of Reset Generation Source Determination Flow, changed
		235	6.5 Usage Notes, added

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010		Section 7. Voltage Detection Circuit (LVD)
		237	Table 7.2 List of Voltage Detection Circuit Registers: Value after Reset in the RSTSR register, changed
		237	7.2.2 Key Code Register for Low-Voltage Detection Control Register (LVDKEYR), changed
		238	7.2.3 Low-Voltage Detection Control Register (LVDCR), changed
		239	Table 7.3 LVDCR Register Settings and Voltage Detection Circuit States, added
		240	Figure 7.2 Timing Diagram 1 of Voltage Monitoring Reset (Reset Selected in LVD2, LVD1 Disabled), changed
		241	Figure 7.3 Timing Diagram 2 of Voltage Monitoring Reset (Reset Selected in LVD2, Reset Selected in LVD1), changed
		242	7.3.2 Voltage Monitoring Interrupt: Description changed
		243	Figure 7.4 Timing of Voltage Monitoring Interrupt (Interrupt Selected in LVD2, Reset Selected in LVD1), changed
		244	Figure 7.5 Example of Procedure for Setting Voltage Monitoring Interrupt, changed
		245	7.3.3 Cancellation of Deep Software Standby Mode by the Voltage Detection Circuit: Description changed
			Section 8. Clock Generation Circuit
		246	Table 8.1 Specifications of Clock Generation Circuit, changed, Note added
		247	Figure 8.1 Block Diagram of Clock Generation Circuit, changed
		247	Table 8.2 Pin Configuration, changed
		249	8.2.1 System Clock Control Register (SCKCR): Bit table and description, changed
		251	8.2.2 External Bus Clock Control Register (BCKCR): Bit table and description, changed
		252	8.2.3 Oscillation Stop Detection Control Register (OSTDCR): Bit table and description, changed
		253	8.2.4 Sub-Clock Oscillator Control Register (SUBOSCCR): Bit table and description, changed
		254	8.3 Main Clock Oscillator: Description changed
		255	8.3.2 External Clock Input: Description changed
		255	Figure 8.4 Examples of External Clock Input, changed
		256	8.4.1 Connecting 32.768-kHz Crystal Resonator: Description changed
		256	Figure 8.5 Connection Example of 32.768-kHz Crystal Resonator, changed
		256	Figure 8.6 Equivalent Circuit for 32.768-kHz Crystal Resonator, changed
		—	Table 8.6 Crystal Resonator Characteristics (Reference Values), deleted
		257	8.4.2 Handling of Pins when Sub-Clock is Not Used: Description changed
		257	Figure 8.7 Pin Handling when Sub-Clock is not Used, changed
		257	8.9 Frequency Divider: Description changed
		258	8.10 Internal Clock: Description changed
		258	8.10.1 System Clock (ICLK): Description changed
		258	8.10.2 Peripheral Module Clock (PCLK): Description changed
		259	8.10.3 External Bus Clock (BCLK): Description changed
		259	8.10.4 SDRAM Clock (SDCLK): Description changed
		259	8.10.6 Dedicated RTC Clock (SUBCLK): Description changed
		261	8.11.1 Detection of Oscillation Stop and Operation after the Detection: Description changed
		263	8.12.1 Notes on Clock Generation Circuit: Description changed
264	8.12.3 Notes on Board Design: Description changed		
	Section 9. Low Power Consumption		
265	Table 9.1 Specifications of Low Power Consumption Function, changed		
266	Table 9.2 Transition and Cancellation of the Mode and the State of Operation, changed		

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	269	Table 9.3 List of Low Power Consumption Registers: Module stop control register A, changed
		271	9.2.1 Standby Control Register (SBYCR): Bit table and description, changed
		273	9.2.2 Module Stop Control Register A (MSTPCRA): Bit table and description, changed
		276	9.2.3 Module Stop Control Register B (MSTPCRB): Bit table and description, changed
		278	9.2.4 Module Stop Control Register C (MSTPCRC): Bit table and description, changed
		279	9.2.5 Deep Standby Control Register (DPSBYCR): Bit table and description, changed
		281	9.2.6 Deep Standby Wait Control Register (DPSWCR): Bit table and description, changed
		282	9.2.7 Deep Standby Interrupt Enable Register (DPSIER): Bit description, changed
		283	9.2.8 Deep Standby Interrupt Flag Register (DPSIFR): Note and description, changed
		285	9.2.9 Deep Standby Interrupt Edge Register (DPSIEGR): Description changed
		286	9.2.10 Reset Status Register (RSTSR): Bit table, Note, and description, changed
		288	9.2.11 Deep Standby Backup Register (DPSBKRY) (y = 0 to 31): Bit chart and description, changed
		289	9.4 Module Stop Function, changed
		290	9.5.1.1 Transition to Sleep Mode: Title and description, changed
		290	9.5.1.2 Canceling Sleep Mode, changed
		291	9.5.2.1 Transition to All-Module Clock Stop Mode: Title and description, changed
		292	9.5.2.2 Canceling All-Module Clock Stop Mode, changed
		293	9.5.3.1 Transition to Software Standby Mode: Title and description, changed
		294	9.5.3.2 Canceling Software Standby Mode, changed
		296	9.5.3.4 Example of Software Standby Mode Application, changed
		297	9.5.4.1 Transition to Deep Software Standby Mode, changed
		298	9.5.4.2 Canceling Deep Software Standby Mode, changed
		299	9.5.4.3 Pin States when Deep Software Standby Mode is Canceled, changed
		303	Figure 9.4 Example of Flowchart to Use Deep Software Standby Mode, changed
		304	9.6 BCLK and SDCLK Output Control, changed
		304	Table 9.6 BCLK Pin (P53) State in Each Low Power Consumption Mode: Title changed
		304	Table 9.7 SDCLK Pin (P70) State in Each Low Power Consumption Mode: Title changed
		305	9.7.1 I/O Port States, changed
		305	9.7.2 Module Stop State of the DMACA, DTC, EXDMAC, and EDMAC, changed
		305	9.7.3 On-Chip Peripheral Module Interrupts, changed
		305	9.7.4 Write-Access to MSTPCRA, MSTPCRB, and MSTPCRC: Title and description, changed
		305	9.7.5 Input Buffer Control by DIRQnE Bit (n = 3 to 0): Title and description, changed
		305	9.7.6 Conflict between Transition to Deep Software Standby Mode and Interrupt: Title and description, changed
		305	9.7.7 Timing of Wait Instructions, changed
			Section 10. Exceptions
		306	10.1 Types of Exceptions, changed
		306	Figure 10.1 Types of Exception: Title changed
		307	10.1.1 Undefined Instruction Exception, changed
		307	10.1.2 Privileged Instruction Exception, changed
		307	10.1.4 Reset, changed
		307	10.1.5 Non-Maskable Interrupt, changed
307	10.1.6 Interrupts, changed		
308	10.2 Exception Handling Procedure: Description changed		

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	308	Figure 10.2 Outline of the Exception Handling Procedure, changed
		310	10.3 Acceptance of Exceptions: Description changed
		310	Table 10.1 Timing of Acceptance and Saved PC Value, changed
		311	10.3.2 Vector and Site for Saving the Values in the PC and PSW: Description changed
		311	Table 10.2 Vector and Site for Saving the Values in the PC and PSW: Title and Table, changed
		312	10.4 Hardware Processing for Accepting and Returning from Exceptions: Description changed
		314	10.5 Hardware Pre-Processing, changed
		314	10.5.1 Undefined Instruction Exception: 2 and 5, changed
		314	10.5.2 Privileged Instruction Exception: 2 and 5, changed
		315	10.5.5 Non-Maskable Interrupt: 2, 3, 4, and 6, changed
		315	10.5.6 Interrupts: 2 to 6, changed
		315	10.5.7 Unconditional Trap: 5, changed
		316	10.6 Return from Exception Processing Routines, changed
		316	Table 10.3 Return from Exception Processing Routines, changed
			Section 11. Interrupt Control Unit (ICUa)
		318	Table 11.1 Specifications of the Interrupt Control Unit, changed
		319	Figure 11.1 Block Diagram of Interrupt Control Unit, changed
		320 to 328	Table 11.3 Registers of the Interrupt Control Unit, changed
		329	11.2.1 Interrupt Request Register i (IRi) (i = interrupt vector number): Bit table and <u>Note changed</u>
		330	(1) Edge detection [Setting condition], Description added
		332	11.2.3 Interrupt Priority Register m (IPRm) (m = 00h to 8Fh): Bit table and description, changed
		333	11.2.4 Fast Interrupt Register (FIR): Bit table and description, changed
		334	11.2.5 Software Interrupt Activation Register (SWINTR): Bit table and description, changed
		335	11.2.6 DTC Activation Enable Register n (DTCERn) (n = interrupt vector number): Bit chart, Bit table, and description, changed
		336	11.2.7 DMACA Activation Source Select Register n (DMRSRn) (n = DMACA channel number): Description changed
		337	11.2.8 IRQ Control Register n (IRQCRn) (n = 0 to 15), changed
		338	11.2.9 Non-Maskable Interrupt Status Register (NMISR), changed
		340	11.2.10 Non-Maskable Interrupt Enable Register (NMIER), changed
		341	11.2.11 Non-Maskable Interrupt Clear Register (NMICLR), changed
		342	11.2.12 NNMI Pin Interrupt Control Register (NMICR), changed
		343	11.3.1 Interrupt Vector Table, changed
		343 to 356	Table 11.4 Interrupt Vector Table, changed
		356	11.3.2 Fast Interrupt Vector, changed
		357	11.4.1.1 Operation of Status Flags for Edge-Detected Interrupts, changed
		357	Figure 11.2 Operation of the IRi.IR Flag in the Case of Edge Detection, changed
		358	Figure 11.3 Timing for Re-setting of the IRn.IR Flag, changed
		358	Figure 11.4 Relation between the IRn.IR Flag and Disabling of the Corresponding Interrupt Source, changed
		359	11.4.1.2 Operation of Status Flags for Level-Detected Interrupts, changed
		359	Figure 11.5 Operation of the IRi.IR Flag in the Case of a Level-Detected Interrupt, changed
		359	Figure 11.6 Level-Detected Interrupt Handling Procedure, added

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	360	11.4.2 Enabling and Disabling Interrupt Sources, changed
		361	11.4.3 Selecting Interrupt Request Destinations: (1) DMACA Activation, DTC Activation, changed
		362	Table 11.5 Operation at the Time of DMACA/DTC Activation: Contents changed and <u>Note</u> added
		364	11.4.5 Fast Interrupt, changed
		365	11.5 Non-maskable Interrupt Operation, changed
		368	11.7 Usage Notes: Items added
		368	11.7.1 Notes on Communication Using DTC or DMACA Transfer, added
		368	(1) Conditions of Transfer Request Loss during Communication using DTC or DMACA Transfer, added
		368	Table 11.6 Combinations of the DTC/DMACA Function which Require Special Care, added
		369	(2) Notes on Reception using DMACA Transfer when DISEL = 0, added
		369	(3) Notes on Communication using DMACA Transfer when DISEL = 1, added
		369	(4) Notes on Communication using DTC Transfer when DISEL = 1, added
		369	(5) Flowchart of Software Preventive Measures (for SCI, RIIC, RSPI), added
		370	(6) Flowchart of Software Preventive Measures (for USB), added
			Section 12. Buses
		372	Figure 12.1 Bus Configuration, changed
		372	Table 12.2 Addresses Assigned for Each Bus, changed
		373	12.2.1 CPU Buses, changed
		373	12.2.3 Internal Main Buses, changed
		374	Table 12.3 Order of Priority for Bus Masters, changed
		374	Table 12.4 Connection of Peripheral Modules to the Internal Peripheral Buses, changed
		375	12.2.5 External Bus, changed
		376	Table 12.6 Pin Configuration of the External Bus: Table and Note, changed
		377	12.2.6 Parallel Operation, changed
		377	Figure 12.2 Example of Parallel Operations, changed
		377	12.2.7 Limitations, added
		384	12.3.3 CSn Mode Register (CSnMOD) (n = 0 to 7): Bit table, description, and Note, changed
		384	Table 12.9 Control Signals for Write Access Mode, changed
		386	12.3.4 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 7), changed
		390	12.3.5 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 7), changed
		395	12.3.8 SDRAM Access Mode Register (SDAMOD): Note changed
		400	12.3.12 SDRAM Initialization Sequence Control Register (SDICR), changed
		404	12.3.15 SDRAM Timing Register (SDTR): Description changed
		406	12.3.16 SDRAM Mode Register (SDMOD): Description and Note, changed
		407	12.3.17 SDRAM Status Register (SDSR): Description changed
		411	12.4.1 Data Alignment Control for CS Area
		413	(1) 32-Bit Bus Space: Description deleted and changed
		415	(2) 16-Bit Bus Space: Description deleted and changed
		415	(3) 8-Bit Bus Space: Description deleted and changed
		417	12.4.2 Data Alignment Control for SDRAM Area, (1) 32-Bit Bus Space: Description deleted
		419	12.4.2 Data Alignment Control for SDRAM Area, (2) 16-Bit Bus Space: Description changed
		422	Figure 12.13 Data Alignment (Little Endian) in 8-Bit Bus Space, changed
425	12.5.1 Timing of CS Area Access: Description changed		

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	429	Figure 12.19 Example of Normal-Read Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer), changed
		431	Figure 12.22 Example of Normal-Write Operation (when 32-Bit Bus Space is Accessed in 16 Bits, in Byte Strobe Mode), changed
		432	Figure 12.23 Example of Normal-Write Operation (when 32-Bit Bus Space is Accessed in 32 Bits, in Single Write Strobe Mode), changed
		436	(2) Page Access: Description changed
		437	Figure 12.30 Page-Write Access Timing, changed
		438	Figure 12.32 Example of Page-Write Access Operation (when 16-Bit Bus Space is Accessed in 32 Bits, in Single Write Strobe Mode), changed
		439	Figure 12.33 Example of Page Read Access Operation (BCLK Pin Output: BCLK = 1:2, when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer), changed
		440	Figure 12.34 Example of Page Write Access Operation (BCLK Pin Output: BCLK = 1:2, when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode), changed
		443	12.5.4 Insertion of Recovery Cycles, changed
		444	12.5.5 Write Buffer Function: Description changed
		445	Table 12.11 Limitations at the Time of Normal and Page Access, changed
		445	12.5.6 Notes on Usage, (3) Limitation when a Pin is Multiplexed between A0 and BC0# Functions: Description changed
		446	(5) Prohibition of Access that Spans Areas of Address Space: Description changed
		447	12.6.3 Insertion of Recovery Cycles, changed
		447	Figure 12.39 Example of Recovery Timing (for SDRAM Access), changed
		448	12.6.4 Write Buffer Function: Description changed
		449	Table 12.15 Conditions for Register Modification: Note changed
		450	12.6.7 Self-Refresh: Description changed
		451	1. Self-Refresh in All-Module-Clock Stop Mode: Description changed
		451	2. Self-Refresh in Software Standby Mode: Description changed
		451	3. Self-Refresh in Deep Software Standby Mode: Description changed
		453	12.6.8 Auto-Refresh: Description changed
		455	Figure 12.46 Timing Example of SDRAM Initialization Sequence, changed
		456	12.6.10 Read/Write Access, (1) Single Access: Description changed
		457	Figure 12.48 Timing Example of Single Read (Cluster Transfer by EXDMAC or Block Transfer in Single Address Mode with SDAMOD.BE = 0 and SDTR.CL[2:0] = 010b: 2 Cycles): Title and figure, changed
		458	Figure 12.50 Timing Example of Consecutive Read (SDAMOD.BE = 1 and SDTR.CL[2:0] = 010b: 2 Cycles): Title changed
		458	Figure 12.51 Timing Example of Consecutive Write (SDAMOD.BE = 1, when the Earliest Timing is Set): Title changed
		458	Figure 12.52 Timing Example of Consecutive Write (SDAMOD.BE = 1, when the Earliest Timing is Set) with More Than One Row Address Accessed): Title changed
		459	12.6.11 Setting Mode Register: Description changed
		460	Figure 12.54 SDRAMC Setting Procedure, changed
		461	Figure 12.55 Procedure for Transition to and Recovery from Self-Refresh Mode, changed
		462	Figure 12.56 Procedure for Transition to and Recovery from Self-Refresh Mode in Deep Software Standby Mode, changed
		463	12.6.12.3 Timing Register Settings and Access Timing, (1) Single Read Timing Examples: Description changed
		463	Table 12.16 Correspondence between Figures 12.57 to 12.59 and STDR Register Settings (Single Read Timing), changed

Rev.	Date	Description			
		Page	Summary		
1.00	Sep. 07, 2010	464	Figure 12.60 Timing Example of Single Read (4), changed		
		465	Figure 12.61 Timing Example of Single Read (5) (Two Bus Accesses Occur for One Transfer Request), changed		
		465	(2) Single Write Timing Examples: Description changed		
		465	Table 12.17 Correspondence between Figures 12.60 to 12.62 and STDR Register Settings (Single Write Timing), changed		
		467	Figure 12.65 Timing Example of Single Write (4), changed		
		468	Figure 12.66 Timing Example of Single Write (5) (Two Bus Accesses Occur for One Transfer Request), changed		
		470	Table 12.19 Correspondence between Figures 12.66 to 12.68 and STDR Register Settings (Consecutive Write Timing): Contents changed		
		473	12.6.14.1 32-Bit Bus Space: Description changed		
		479	12.6.15 Restrictions, (2) Low Power Consumption State: Description changed		
		481	Table 12.21 Types of Bus Errors: 007F 8000h to 007F 9FFFh, changed		
					Section 13. DMA Controller (DMACA)
		483	Table 13.1 Specifications of DMACA, changed		
		484	Figure 13.1 Block Diagram of DMACA, changed		
		487	13.2.1 DMA Source Address Register (DMSAR): Description deleted		
		487	13.2.2 DMA Destination Address Register (DMDAR): Description deleted		
		488	13.2.3 DMA Transfer Count Register (DMCRA): Value after reset, changed		
		488	(1) Normal transfer mode (MD[1:0] bits in DMACAn.DMTMD = 00b): Description changed		
		488	(3) Block transfer mode (MD[1:0] bits in DMACAn.DMTMD = 10b): Description changed		
		489	13.2.4 DMA Block Transfer Count Register (DMCRB): Description changed		
		490	13.2.5 DMA Transfer Mode Register (DMTMD): Description changed		
		491	13.2.6 DMA Interrupt Setting Register (DMINT): Bit table changed and description deleted, changed		
		493	13.2.7 DMA Address Mode Register (DMAMD), SARA[4:0] Bits (Source Address Extended Repeat Area): Description changed		
		498	13.2.10 DMA Software Start Register (DMREQ): Description changed		
		502	13.2.13 DMA Start Register (DMAST): Bit table changed and description changed		
		503	13.3.1 Transfer Mode, (2) Repeat Transfer Mode: Description changed		
		505	(3) Block Transfer Mode: Description deleted, change		
		506	Figure 13.4 Operation in Block Transfer Mode, changed		
		507	13.3.2 Extended Repeat Area Function: Description deleted, changed		
		509	13.3.3 Address Update Function using Offset, (2) Example of XY Conversion Using Offset Addition: Description changed		
		512	Figure 13.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode, changed		
		513	13.3.4 Activation Sources, (1) DMACA Activation by Software, (2) DMACA Activation by Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests: Description changed		
		513	Table 13.8 Interrupt Requests Specified as DMACA Activation Sources, changed		
		515	Figure 13.10 Register Setting Procedure, changed		
		516	13.3.6 Starting DMA Transfer: Description changed		
		518	13.3.8 Channel Priority: Description changed		
		518	13.3.9 Operation Timing, added		
519	13.3.10 DMACA Execution Cycles, added				
522	13.5 Interrupts: Description deleted, changed				

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	522	Table 13.10 Relation among Interrupt Sources, Interrupt Status Bits, and Interrupt Enable Bits, changed
		—	(1) When Discontinuing or Terminating DMA Transfer, (2) When Continuing DMA Transfer, Figure 13.12 Interrupt Handling Routine for Canceling Interrupt to Restart DMA Transfer, deleted
		524	13.6 Low-Power Consumption Function, changed
		525	13.7.7 Suspending or Restarting DMA Activation, added
		525	13.7.8 Selecting Communication Function Interrupt as DMA Transfer Activation Source, added
		—	Section 14. EXDMA Controller (EXDMAC)
		—	The register symbol of EXDMAC, changed from EDMAC to EXDMAC
		526	Table 14.1 Specifications of EXDMAC, changed and deleted
		527	Figure 14.1 Block Diagram of EXDMAC, changed
		528	14.2 Register Descriptions: Description changed
		529	Table 14.3 Registers of EXDMAC, changed and deleted
		531	14.2.3 EXDMA Transfer Count Register (EDMCRA): Symbol and description, changed
		532	(1) Normal transfer mode (MD[1:0] bits in EXDMACn.EDMTMD = 00b) to (4) Cluster transfer mode (MD[1:0] bits in EXDMACn.EDMTMD = 11b): Description changed
		533	14.2.4 EXDMA Block Transfer Count Register (EDMCRB): Description changed
		534	14.2.5 EXDMA Transfer Mode Register (EDMTMD): Bit table and description, changed
		544	14.2.11 EXDMA Software Start Register (EDMREQ): Bit description, changed
		551	14.2.17 Cluster Buffer Register i (CLSBRi) (i = 0 to 6): Title changed, deleted, and description changed
		552	Table 14.5 Register Update Operation in Normal Transfer Mode: Table and Note, changed
		552	14.3.1 Transfer Mode, (2) Repeat Transfer Mode: Description changed
		554	(3) Block Transfer Mode: Description changed
		556	(4) Cluster Transfer Mode: Description changed
		557	Figure 14.5 Operation in Cluster Transfer Mode: CLSBR7 deleted
		560	Table 14.9 Address Update Method in Each Address Update Mode, changed
		561	(1) Basic Transfer Using Offset Addition: Description changed
		562	(2) Example of XY Conversion Using Offset Addition: Description changed
		563	Figure 14.10 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode, changed
		564	Table 14.10 Relationship between Transfer Modes and Address Modes, changed
		565	14.4.1 Normal/Repeat Transfer Operation, (1) Dual Address Mode, (2) Single Address Mode: Description changed
		572	14.5 Activation Sources: Description changed
		572	(1) EXDMAC Activation by Software, (2) EXDMAC Activation by External DMA Request Pin (EDREQ), added
		573	Figure 14.22 External DMA Request Timing in Falling-Edge Detection Mode: Title and Figure, changed
		573	Figure 14.23 External DMA Request Timing in Low-Level Detection Mode: Title and Figure, changed
		573	(3) EXDMAC Activation by DMA Requests from On-Chip Peripheral Modules (MTU1 Compare Match): Description changed
574	Figure 14.24 Register Setting Procedure, changed		
575	14.5.2 Register Setting Procedure: Description changed		
575	14.5.3 Registers during DMA Transfer: Description changed		

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	575	(1) EXDMA Source Address Register (EXDMACn.EDMSAR) to (4) EXDMA Block Transfer Count Register (EXDMACn.EDMCRB): Title changed
		575	(5) EXDMA Transfer Enable Bit (EXDMACn.EDMCNT.DTE) to (8) Transfer Escape End Interrupt Flag (EXDMACn.EDMSTS.ESIF): Title and description, changed
		576	14.5.4 Channel Priority: Description changed
		579	Figure 14.25 Schematic Logic Diagram of Interrupt Outputs, changed
		579	14.7 Interrupts, (1) When Discontinuing or Terminating DMA Transfer, (2) When Continuing DMA Transfer: Description changed
		581	14.8 Low-Power Consumption Function: Description changed
		591	14.10.1 Cluster Buffers: Description changed, deleted
		591	Figure 14.37 Data Storage in Cluster Buffers: CLSBR7 deleted
			Section 15. Data Transfer Controller (DTCa)
		592	Table 15.1 DTC Specifications: Description changed, deleted
		594	15.2 Register Descriptions: Description changed
		594	Table 15.2 Registers of the DTC, changed
		595	15.2.1 DTC Mode Register A (MRA): Bit table and description, changed
		596	15.2.2 DTC Mode Register B (MRB): Bit table and description, changed
		597	15.2.3 DTC Transfer Source Address register (SAR): Title and description, changed
		597	15.2.4 DTC Transfer Destination Address Register (DAR): Title and description, changed
		598	15.2.5 DTC Transfer Count Register A (CRA): Description changed
		599	15.2.6 DTC Transfer Count Register B (CRB): Description changed
		600	15.2.8 DTC Vector Base Register (DTCVBR): Description changed
		600	15.2.9 DTC Address Mode Register (DTCADMOD): Description changed
		601	15.2.10 DTC Module Start Register (DTCST): Description changed
		602	15.2.11 DTC Status Register (DTCSTS): Description changed
		603	15.3 Sources of Activation: Description changed
		603	15.3.1 Allocating Transfer Data and DTC Vector Table, changed
		604	Figure 15.2 DTC Vector Table and Transfer Data: Title and Figure, changed
		604	Figure 15.3 Allocation of Transfer Data in the RAM Area, changed
		605	Table 15.3 Correspondence between Interrupt Sources, DTC Vector Addresses, and the ICU.DTCERn Register, changed
		608	15.4 Sources of Activation: Description changed
		608	Table 15.4 Transfer Modes of the DTC: Table and Note, changed
		609	Figure 15.4 Operation Flowchart of the DTC, changed
		610	Table 15.5 Chain Transfer Conditions: Table and Note, changed
		611	15.4.1 Transfer Data Read Skip Function: Description changed
		612	15.4.3 Normal Transfer Mode: Description changed
		612	Table 15.7 Register Functions in Normal Transfer Mode: Register Functions in Normal Transfer Mode
		613	15.4.4 Repeat Transfer Mode: Description changed
		613	Table 15.8 Register Functions in Repeat Transfer Mode: Title and Note, changed
		613	Figure 15.6 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area): Title and Figure, changed
		614	15.4.5 Block Transfer Mode: Description changed
		614	Table 15.9 Register Functions in Block Transfer Mode, changed
		614	Figure 15.7 Memory Map of Block Transfer Mode (Transfer Destination: Block Area): Title and Figure, changed

Rev.	Date	Description			
		Page	Summary		
1.00	Sep. 07, 2010	615	15.4.6 Chain Transfer, changed		
		615	Figure 15.8 Chain Transfer Operation, changed		
		616	Figure 15.9 Example of DTC Operation Timing 1 (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode), changed		
		616	Figure 15.10 Example of DTC Operation Timing 2 (Short-Address Mode, Block Transfer Mode, Block Size = 2): Title and Figure, changed		
		617	Figure 15.11 Example of DTC Operation Timing 3 (Short-Address Mode, Chain Transfer): Title and Figure, changed		
		617	Figure 15.12 Example of DTC Operation Timing 4 (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode): Title and Figure, changed		
		618	Figure 15.13 Example of Operation when Transfer Information Skip is Executed, moved		
		619	15.4.8 Execution Cycle of the DTC: Description changed		
		619	Table 15.10 Execution Cycle of the DTC, changed		
		619	15.4.9 DTC Bus Mastership Release Timing: Description changed		
		620	15.5 DTC Setting Procedure: Description changed		
		620	Figure 15.14 Procedure to Set the DTC, changed		
		612	15.6.1 Normal Transfer, (1) Transfer Data Set, (3) ICU Set and DTC Module Activation to (5) DTC Transfer: Description changed		
		622	15.6.2 Chain Transfer: Description changed		
		622	(1) First Transfer Data Set, (2) Second Transfer Data Set		
		624	15.6.3 Chain Transfer when Counter = 0: Title and description, changed		
		625	Figure 15.15 Chain Transfer when Counter = 0: Title and Figure, changed		
		625	15.7 Interrupt Source: Description changed		
		626	15.8 Low-Power Consumption Function changed		
		626	(1) Module Stop Function to (3) Software Standby and Deep Software Standby Modes: Title and description, changed		
		626	(4) Notes on Low-Power Consumption Function, added		
		627	15.9 Usage Notes Title changed		
		627	15.9.1 Transfer Information Data Start Address: Title and description, changed		
		628	15.9.4 Selecting Communication Function Interrupt as DTC Activation Source, added		
					Section 16. I/O Ports
				629	Table 16.1 Specifications of I/O Ports (176-Pin LFBGA), changed
				635	Table 16.2 Port Functions (176-Pin LFBGA), changed
				636	Table 16.3 Registers of I/O Ports (176-Pin LFBGA), changed
				644	16.1.2.3 Port Register (PORT): Note added, Bit table and description, changed
				645	16.1.2.4 Input Buffer Control Register (ICR): Note and description, changed
				647	16.1.2.6 Pull-Up Resistor Control Register (PCR): Title, Table, and description, changed
				647	Table 16.5 Input Pull-Up Resistor States (176-Pin LFBGA): Title and Note, changed
				649	16.1.2.7 Port Function Control Register 0 (PF0CSE): Bit table, changed
				650	16.1.2.8 Port Function Control Register 1 (PF1CSS): Bit table and description, changed, deleted
				—	Figure 16.1 Timing for Output of CS# Signals to the Same Pin, deleted
				—	Table 16.6 Relationship between CS# Output Pin Select Registers and Output Pins, deleted
				651	16.1.2.9 Port Function Control Register 2 (PF2CSS): Bit table and description, changed
				657	16.1.2.14 Port Function Control Register 7 (PF7DMA): Bit table, changed
				658	16.1.2.15 Port Function Control Register 8 (PF8IRQ): Bit description changed
				660	16.1.2.18 Port Function Control Register B (PFBTMR): Bit table and description, changed

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	661	16.1.2.19 Port Function Control Register C (PFCMTU): Bit table and description, changed
		662	16.1.2.20 Port Function Control Register D (PFDMTU): Bit table and description, changed
		671	16.1.2.26 Port Function Control Register L (PFLUSB): Bit table and description, changed
		672	Table 16.7 Relationship between the USBMD[1:0] Bits Setting and USB Mode (USB0): Title and Note, changed
		673	16.1.2.27 Port Function Control Register L (PFLUSB): Bit table and description, changed
		674	Table 16.8 Relationship between the USBMD[1:0] Bits Setting and USB Mode (USB1): Title and Note, changed
		678 to 691	Table 16.10 Port-Multiplexed Priority for Peripheral Modules (176-Pin LFBGA), changed
		692	Table 16.11 Output Enable Settings for Each Port (176-Pin LFBGA), changed
		693	Table 16.12 Specifications of I/O Ports (145-Pin TFLGA/144-Pin LQFP), changed
		699	Table 16.13 Port Functions (145-Pin TFLGA/144-Pin LQFP), changed
		701	Table 16.14 Registers of I/O Ports (145-Pin TFLGA/144-Pin LQFP), changed
		708	16.2.2.3 Port Register (PORT): Note added, Bit table and description, changed
		709	16.2.2.4 Input Buffer Control Register (ICR): Note and description, changed
		711	16.2.2.6 Pull-up Resistor Control Register (PCR): Note, bit table, and description, changed
		711	Table 16.16 Input Pull-Up Resistor States (145-Pin TFLGA/144-Pin LQFP): Table and [Legend], changed
		714	16.2.2.8 Port Function Control Register 1 (PF1CSS): Description changed, deleted
		715	16.2.2.9 Port Function Control Register 2 (PF2CSS): Bit table and description, changed
		717	16.2.2.11 Port Function Control Register 4 (PF4BUS): Bit description changed
		718	16.2.2.12 Port Function Control Register 5 (PF5BUS): Bit table and description, changed
		719	16.2.2.13 Port Function Control Register 6 (PF6BUS): Bit table and description, changed
		722	16.2.2.16 Port Function Control Register 9 (PF9IRQ): Bit description, changed
		724	16.2.2.19 Port Function Control Register C (PFCMTU): Bit table and description, changed
		725	16.2.2.20 Port Function Control Register D (PFDMTU): Description changed
		728	16.2.2.22 Port Function Control Register F (PFFSCI): Bit table, changed
		733	16.2.2.26 Port Function Control Register K (PFKUSB): Note, bit table, and description, changed
		734	Table 16.18 Relationship between the USBMD[1:0] Bits Setting and USB Mode (USB0): Title and Note, changed
		737	16.2.3 Settings of Ports: Description changed
		738	16.2.4 List of Output Enable Settings: Title and description, changed
		738 to 749	Table 16.20 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP), changed
		750	Table 16.21 Treatment of Unused Pins (145-Pin TFLGA/144-Pin LQFP), changed
		751	Table 16.22 Specifications of I/O Ports 100-Pin LQFP), changed
		755	Table 16.23 Port Functions (100-Pin LQFP), changed
		757	Table 16.24 Registers of I/O Ports (100-Pin LQFP), changed
		764	16.3.2.3 Port Register (PORT): Note added, Bit table and description, changed
		765	16.3.2.4 Input Buffer Control Register (ICR): Note and description, changed
		767	16.3.2.6 Pull-up Resistor Control Register (PCR): Title, bit table, and description, changed
		767	Table 16.26 Relationship between the PHYMODE Bit Setting and Ethernet Mode (100-Pin LQFP): Title, Table, and Note, changed
		771	16.3.2.9 Port Function Control Register 4 (PF4BUS): Bit description changed
		779	16.3.2.18 Port Function Control Register F (PFFSCI): Bit table, changed
		785	16.3.2.22 Port Function Control Register K (PFKUSB): Bit: Bit table and description, changed

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	786	Table 16.28 Relationship between the USBMD[1:0] Bits Setting and USB Mode (USB0): Table and Note, changed
		787	16.3.2.23 Port Function Control Register M (PFMPOE): Bit table and description, changed
		788	16.3.2.24 Port Function Control Register N (PFNPOE): Bit table and description, changed
		789	16.3.3 Settings of Ports: Description changed
		790 to 798	Table 16.30 Output Enable Settings for Each Port (100-Pin LQFP), changed
		799	Table 16.31 Treatment of Unused Pins (100-Pin LQFP), changed
		800	Table 16.32 Specifications of I/O Ports (85-Pin TFLGA), changed
		804	Table 16.33 Port Functions (85-Pin TFLGA), changed
		805	Table 16.34 Registers of I/O Ports (85-Pin TFLGA), changed
		811	16.4.2.3 Port Register (PORT): Note added, Bit table and description, changed
		812	16.4.2.4 Input Buffer Control Register (ICR): Note and description, changed
		814	16.4.2.6 Pull-up Resistor Control Register (PCR): Title and Note, changed
		814	Table 16.36 Input Pull-Up Resistor States (85-Pin TFLGA): Table and [Legend], changed
		815	16.4.2.7 Port Function Control Register 0 (PFOCSE): Bit: Bit table, changed
		817	16.4.2.9 Port Function Control Register 4 (PF4BUS): Bit description changed
		820	16.4.2.13 Port Function Control Register F (PFFSCI): Bit table, changed
		825	16.4.2.17 Port Function Control Register K (PFKUSB): Bit table, note, and description, changed
		826	Table 16.37 Relationship between the USBMD[1:0] Bits Setting and USB Mode (USB0), changed
		827	Table 16.38 Port-Multiplexed Priority for Peripheral Modules (85-Pin TFLGA), changed
		828 to 834	Table 16.39 Output Enable Settings for Each Port (85-Pin TFLGA), changed
		835	Table 16.40 Treatment of Unused Pins (85-Pin TFLGA), changed
		836	16.5.1 Setting the Input Buffer Control Register (Pn.ICR): Description changed
		836	16.5.3 Changing the Output Enable Settings, added
		836	16.5.4 Setting Open-Drain Pins with TDO Function, added
			Section 17. Multi-Function Timer Pulse Unit 2 (MTU2)
		838	17.1 Overview: Description changed
		838	Table 17.1 Specifications of MTU, changed
		842	Table 17.3 MTU Functions (Unit 1): Compare match output, changed
		853	Table 17.5 Registers of MTU: MTU11 changed
		854	17.2.1 Timer Control Register (TCR): Bit table and bit description, changed
		856	Table 17.9 TPSC[2:0] (Channels 1 and 7): Note changed
		858	17.2.2 Timer Mode Register (TMDR): Description changed
		859	Table 17.13 Operating Mode Setting by MD[3:0] Bits: Note 3, changed
		862	Table 17.14 TIORH (MTU0 and MTU6), changed
		863	Table 17.15 TIORL (MTU0 and MTU6), changed
		864	Table 17.16 TIOR (MTU1 and MTU7), changed
		865	Table 17.17 TIOR (MTU2 and MTU8), changed
		866	Table 17.18 TIORH (MTU3 and MTU9), changed
		867	Table 17.19 TIORL (MTU3 and MTU9): Table and Note, changed
		868	Table 17.20 TIORH (MTU4 and MTU10), changed
		869	Table 17.21 TIORL (MTU4 and MTU10), changed
870	Table 17.22 TIORH (MTU0 and MTU6), changed		
871	Table 17.23 TIORL (MTU0 and MTU6): Table and Note 1, changed		

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	872	Table 17.24 TIOR (MTU1 and MTU7), changed
		873	Table 17.25 TIOR (MTU2 and MTU8), changed
		874	Table 17.26 TIORH (MTU3 and MTU9), changed
		875	Table 17.27 TIORL (MTU3 and MTU9): Table and Note 1, changed
		876	Table 17.28 TIORH (MTU4 and MTU10), changed
		877	Table 17.29 TIORL (MTU4 and MTU10): Table and Note 1, changed
		882	17.2.5 Timer Interrupt Enable Register (TIER) • TIER (MTU5, MTU11): Address description, changed
		884	17.2.7 Timer Buffer Operation Transfer Mode Register (TBTM): Bit table, changed
		889	17.2.14 Timer Buffer Operation Transfer Mode Register (TBTM) • TSTRA (MTU5), TSTRB (MTU11): Address description, changed
		894	17.2.18 Timer Output Control Registers 1 (TOCR1A and TOCR1B): Note 1 in Bit chart, Bit table, Notes 1 to 3, and description, changed
		895	Table 17.32 Output Level Select Function: Title and contents, changed
		895	Table 17.33 Output Level Select Function: Title and contents, changed
		896	17.2.19 Timer Output Control Registers 2 (TOCR2A and TOCR2B): Bit table and description, changed
		896, 897	Table 17.34 MTIOCMB Output Level Select Function to Table 17.39 MTIOCMD Output Level Select Function: Title and contents, changed
		900	17.2.21 Timer Gate Control Registers (TGCR A and TGCR B): Bit table and description, changed
		901	Table 17.41 Output Level Select Function: Title changed
		902	17.2.23 Timer Dead Time Data Registers (TDDRA and TDDR B): Title changed
		902	17.2.24 Timer Cycle Data Registers (TCDRA and TCDR B): Description changed
		902	17.2.25 Timer Cycle Buffer Registers (TCBRA and TCBR B): Description changed
		907	17.2.29 Timer Dead Time Enable Registers (TDER A and TDER B): Bit table and description, changed
		908	17.2.30 Timer Waveform Control Registers (TWCRA and TWCR B): Bit table and description, changed
		910	17.3.1 Basic Functions: Title changed
		912	(2) Waveform Output by Compare Match: Description changed
		912	Figure 17.7 Example of Procedure for Setting Waveform Output by Compare Match, changed
		913	(b) Examples of Waveform Output Operation: Description changed
		913	Figure 17.8 Example of Low Output and High Output Operation: Title and Figure, changed
		916	17.3.2 Synchronous Operation: Description changed
		916	Figure 17.12 Example of Synchronous Operation Setting Procedure: Description changed
		918	17.3.3 Buffer Operation, (2) Examples of Buffer Operation: Description changed
		912	(3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation: Description changed
		927	17.3.5 PWM Modes: Description changed
		929	Figure 17.25 Example of PWM Mode Setting Procedure: Description changed
		929	(2) Examples of PWM Mode Operation: Description changed
		931	Figure 17.28 Examples of Duty Cycle Modification in PWM Mode, changed
		932	17.3.6 Phase Counting Mode: Description changed
		933	Table 17.50 Up-/Down-Count Conditions in Phase Counting Mode 1, changed
934	Table 17.51 Up-/Down-Count Conditions in Phase Counting Mode 2, changed		
935	Table 17.52 Up-/Down-Count Conditions in Phase Counting Mode 3, changed		
936	Figure 17.33 Example of Operation in Phase Counting Mode 4, changed		

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	936	Table 17.53 Up-/Down-Count Conditions in Phase Counting Mode 4, changed
		939	Figure 17.35 Procedure for Selecting Reset-Synchronized PWM Mode (Unit 0): [7] changed
		941	17.3.8 Complementary PWM Mode: Description changed
		944	Figure 17.38 Example of Complementary PWM Mode (Unit 0) Setting Procedure: [6] and [10], changed
		948	Table 17.58 Registers and Counters Requiring Initial Setting: Table and Note, changed
		949	(f) Dead Time Suppressing: Description changed
		961	(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode: Description changed
		962	Figure 17.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode, changed
		965	(o) Counter Clearing by MTUn.TGRA Compare Match (n = 3 or 9): Description changed
		968	Figure 17.67 Example of Interrupt Skipping Operation Setting Procedure, changed
		969	(b) Example of Interrupt Skipping Operation: Description changed
		969	(c) Buffer Transfer Control Linked with Interrupt Skipping: Description and Note, changed
		972	(4) Complementary PWM Mode Output Protection Functions :Note changed
		980	17.4.1 Interrupt Sources and Priorities: Description changed
		981	Table 17.59 MTU Interrupt Sources (1) (Unit 0): Channel 4, changed
		983	(3) Underflow Interrupt: Description changed
		985	17.4.3 A/D Converter Activation: Description changed
		992	Figure 17.96 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop), changed
		997	17.6.3 Note on Cycle Setting: Description changed
		999	Figure 17.108 Contention between TGR Write Operation and Compare Match, changed
		999	Figure 17.109 Contention between Buffer Register Write Operation and Compare Match, changed
		1000	Figure 17.110 Contention between Buffer Register Write and TCNT Clear Operations, changed
		1001	Figure 17.111 Contention between TGR Read Operation and Input Capture (Channels 0 to 4 or 6 to 10): Title and Figure, changed
		1001	Figure 17.112 Contention between TGR Read Operation and Input Capture (Channel 5 or 11): Title and Figure, changed
		1005	17.6.14 Buffer Operation Setting in Complementary PWM Mode: Description changed
		1008	17.6.18 Contention between TCNT Write Operation and Overflow/Underflow: Description changed
		1009	17.6.19 Note on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode: Description changed
		1009	17.6.21 Notes when Complementary PWM Mode Output Protection Functions are not Used, added
		1011	17.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation: Description and Note, changed
		1013	(2) Operation when Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1 : Description changed
1013	(3) Operation when Error Occurs in Normal Mode and Operation is Restarted in PWM mode 2: Description changed		
1017	(7) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode: Description changed		
1018	(8) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1: Description changed		
1021	(12) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode: Description changed		

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	1022	Figure 17.134 Error Occurrence in PWM Mode 2, Recovery in Normal Mode, changed
			Section 18. Port Output Enable 2 (POE2)
		1040	Table 18.4 POE Register Configuration: Note added
		1041 to 1043	18.2.1 Input Level Control/Status Register 1 (ICSR1): Bit table and description, changed
		1044	18.2.2 Output Level Control/Status Register 1 (OCSR1): Bit table, Note, and description, changed
		1049	18.2.5 Input Level Control/Status Register 3 (ICSR3): Bit table and description, changed
		1051	18.2.6 Input Level Control/Status Register 4 (ICSR4): Bit table and description, changed
		1053, 1054	18.2.7 Software Port Output Enable Register (SPOER): Bit table and description, changed
		1055, 1056	18.2.8 Port Output Enable Control Register 1 (POECR1): Bit table and description, changed
		1057	18.2.9 Port Output Enable Control Register 2 (POECR2): Bit table and description, changed
		1059	Table 18.5 Target Pins and Conditions for High-Impedance Control, changed
		1062	18.3.5 Release from High-Impedance State: Description changed
		1063	18.5 Usage Notes: Description changed
			Section 19. Programmable Pulse Generator (PPG)
		1064	Opening description, changed
		1064	Table 19.1 Specifications of PPG, changed
		1070	19.2.2 Next Data Transfer Enable Registers H and L (NDERH, NDERL): Bit table and description, changed
		1074	19.2.3 Output Data Registers H and L (PODRH, PODRL): Bit table, changed
		1076	19.2.4 Next Data Registers H and L (NDRH, NDRL): Bit table, changed
		1082, 1083	19.2.5 PPG Output Control Register (PCR): Bit table, changed
		1084	19.2.6 PPG Output Mode Register (PMR): Bit table and description, changed
		1087	Figure 19.5 Timing of Transfer and Output of the Values in NDR (Example): Title changed
		1088	Figure 19.6 Sample Setup Procedure for Normal Pulse Output (PPG0 Setting), changed
		1089	Figure 19.7 Sample Setup Procedure for Normal Pulse Output (PPG1 Setting), changed
		1090	19.3.3 Example of Normal Pulse Output (Example of Five-Phase Pulse Output): Description changed
		1091	19.3.4 Non-Overlapping Pulse Output: Description changed
		1092	Figure 19.10 Non-Overlapping Operation and Write Timing to PPGn.NDRH and PPGn.NDRL, changed
		1093	Figure 19.11 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG0 Setting), changed
		1094	Figure 19.12 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG1 Setting), changed
		1095	19.3.6 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output): Description changed
		1097	19.3.7 Inverted Pulse Output: Description changed
			Section 20. 8-Bit Timer (TMR)
		1100	Table 20.1 Specifications of TMR: Table and Note, changed
		1104	Table 20.3 Registers of TMR: Value after reset in the TCSR register, changed
		1105	20.2.1 Timer Counter (TCNT): Description changed
		1105	20.2.2 Time Constant Register A (TCORA): Description changed
		1106	20.2.3 Time Constant Register B (TCORB): Description changed
		1107	20.2.4 Timer Control Register (TCR): Bit table and bit description, changed
		1108	20.2.5 Timer Counter Control Register (TCCR): Bit description changed

Rev.	Date	Description			
		Page	Summary		
1.00	Sep. 07, 2010	1110	20.2.6 Timer Control/Status Register (TCSR): Bit description order and description, changed		
		1112	20.3.1 Pulse Output: Description changed		
		1113	20.3.2 Reset Input: Description changed		
		1115	20.4.2 Timing of Interrupt Flag Setting to 1 at Compare Match: Title and description, changed		
		1118	20.4.6 Timing of Overflow Interrupt Flag Setting to 1: : Title and description, changed		
		1118	Figure 20.12 Timing of Overflow Interrupt Flag Setting to 1, changed		
		1119	20.5 Operation with Cascaded Connection: Description changed		
		1119	20.5.2 Compare Match Count Mode: Description changed		
		1120	Table 20.6 TMR Interrupt Sources: Table and Note 1, changed		
		1120	20.6.2 A/D Converter Activation: Description changed		
		1121	Figure 20.13 Conflict between TCNT Write and Counter Clear: Title and Figure, changed		
		1122	Figure 20.14 Conflict between TCNT Write and Increment: Title and Figure, changed		
		1123	20.7.5 Conflict between TCORA or TCORB Write and Compare Match: Title and description, changed		
		1123	Figure 20.15 Conflict between TCORA or TCORB Write and Compare Match: Title and Figure, changed		
		1123	20.7.6 Conflict between Compare Matches A and B: Description changed		
		1123	Table 20.7 Timer Output Priorities, changed		
		1124	20.7.7 Switching of Internal Clocks and TCNT Operation: Description changed		
		1125	20.7.8 Clock Source Setting with Cascaded Connection: Description changed		
					Section 21. Compare Match Timer (CMT)
				1126	Table 21.1 Specifications of CMT, changed
				1126	Figure 21.1 Block Diagram of CMT (Unit 0), changed
				1130	21.2.3 Compare Match Timer Control Register (CMCR): Bit table and description, changed
				1131	21.2.4 Compare Match Counter (CMCNT): Description changed
				1131	21.2.5 Compare Match Constant Register (CMCOR): Description changed
				1132	21.3.1 Periodic Count Operation: Description changed
				1132	21.3.2 CMCNT Count Timing: Description changed
				1132	Figure 21.3 CMCNT Count Timing, changed
				1133	21.4.1 Interrupt Sources: Description changed
				1133	21.4.2 Timing of Compare Match Interrupt Generation: Description changed
				1133	Figure 21.4 Timing of Compare Match Interrupt Flag Setting to 1: Figure changed
				1134	21.5 Usage Notes: Title changed
				1134	21.5.2 Conflict between Write and Compare-Match Processes of CMCNT, changed
				1134	Figure 21.5 Conflict between Write and Compare Match Processes of CMCNT, changed
				1134	21.5.3 Conflict between Write and Count-Up Processes of CMCNT, changed
				1134	Figure 21.6 Conflict between Write and Count-Up Processes of CMCNT, changed
				1135	21.5.4 Notes on Data Write to the Compare-Match Timer Control Register (CMCR), added
				1135	21.5.5 Notes on the Compare-Match Timer Counter (CMCNT) and the Compare-Match Constant Register (CMCOR), added
					Section 22. Realtime Clock (RTC)
				1136	Figure 22.1 Block Diagram of RTC, changed
				1137	Table 22.2 Pin Configuration, changed
				1160	Figure 22.6 Procedure for Disabling Alarm Interrupt Request, changed
					Section 23. Watchdog Timer (WDT)
				1165	Opening description, changed

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	1165	Table 23.1 Specifications of WDT, changed
		1167	23.2.1 Timer Counter (TCNT): Description changed
		1169	23.2.3 Reset Control/Status Register (RSTCSR): Description changed
		1170	23.2.4 Write Window A Register (WINA): Description changed
		1170	23.2.5 Write Window B Register (WINB): Description changed
		1171	23.3.1 Watchdog Timer Mode: Description changed
		1171	Figure 23.2 Operation in Watchdog Timer Mode, changed
		1172	23.3.2 Interval Timer Mode: Description changed
		1172	23.4 Interrupt Source: Description changed
		1173	23.5.1 Notes on Register Access: Description changed
		1174	23.5.2 Conflict between Timer Counter (TCNT) Write and Increment: Description changed
		1174	23.5.3 Changing Values of Bits CKS[2:0]: Description changed
		1175	23.5.4 Switching between Watchdog Timer Mode and Interval Timer Mode: Description changed
		1175	23.5.5 Internal Reset in Watchdog Timer Mode: Description changed
		1175	23.5.6 System Reset by WDTOVF# Signal: Description changed
		1175	23.5.7 Transition to Watchdog Timer Mode or Software Standby Mode: Title and description, changed
			Section 24. Independent Watchdog Timer (IWDT)
		1176	Opening description, changed
		1177	24.2.1 IWDT Refresh Register (IWDTRR): Bit table and description, changed
		1178	24.2.2 IWDT Control Register (IWDTCR): Bit table and description, changed
		1184	24.4.1 Limitation on Transitions to Low-Power-Consumption Modes: Description changed
			Section 25. Ethernet Controller (ETHERC)
		1196	25.2.9 Transmit Retry Over Counter Register (TROCR): Bit name, changed
		1197	25.2.10 Delayed Collision Detect Counter Register (CDCR): Bit chart, changed
		1197	25.2.11 Lost Carrier Counter Register (LCCR): Bit chart, changed
		1198	25.2.12 Carrier Not Detect Counter Register (CNDCR): Bit chart, changed
		1198	25.2.13 CRC Error Frame Receive Counter Register (CEFCR): Bit chart, changed
		1199	25.2.14 Frame Receive Error Counter Register (FRECR): Bit chart, changed
		1199	25.2.15 Too-Short Frame Receive Counter Register (TSFRCR): Bit chart, changed
		1200	25.2.16 Too-Long Frame Receive Counter Register (TLFRCR): Bit chart, changed
		1200	25.2.17 Residual-Bit Frame Receive Counter Register (RFCR): Bit chart, changed
		1201	25.2.18 Multicast Address Frame Receive Counter Register (MAFCR): Bit chart, changed
		1206	25.3.1 Transmission: Description changed
		1212	Figure 25.13 MII/RMII Management Frame Format, changed
			Section 26. Ethernet Controller Direct Memory Access Controller (EDMAC)
		1227	26.2.6 ETHERC/EDMAC Status Register (EESR): Note and bit description, changed
		1239	26.2.15 Receive Buffer Write Address Register (RBWAR): Bit chart, changed
		1239	26.2.16 Receive Descriptor Fetch Address Register (RDFAR): Bit chart, changed
		1240	26.2.17 Transmit Buffer Read Address Register (TBRAR): Bit chart, changed
		1240	26.2.18 Transmit Descriptor Fetch Address Register (TDFAR): Bit chart, changed
		1244	26.2.22 Independent Output Signal Setting Register (IOSR): Description changed
		1245	26.3.1.1 Transmit Descriptor, (1) Transmit Descriptor 0 (TD0): Table and description, changed

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	1248	26.3.1.2 Receive Descriptor, (1) Receive Descriptor 0 (RD0): Table and description, changed
			Section 27. USB 2.0 Host/Function Module (USB)
		1255	Table 27.1 Specifications of USB, changed
		1256	Figure 27.1 Block Diagram of USB, changed
		1260	Table 27.3 Registers of USB: Note added
		1263	27.2.2 System Configuration Status Register 0 (SYSSTS0): Bit table and description, changed
		1268	27.2.4 CFIFO Port Register (CFIFO), D0FIFO Port Register (D0FIFO), D1FIFO Port Register (D1FIFO): Description changed
		1269	Table 27.8 Endian Operation in 8-Bit Access, changed
		1284	27.2.12 SOF Output Configuration Register (SOFCFG): Bit table and description, changed
		1289 to 1291	27.2.14 Interrupt Status Register 1 (INTSTS1): Bit table and description, changed
		1299	27.2.22 USB Request Value Register (USBVAL): Bit chart, changed
		1299	27.2.23 USB Request Index Register (USBINDX): Bit chart, changed
		1300	27.2.24 USB Request Length Register (USBLENG): Bit chart, changed
		1307	27.2.28 Pipe Window Select Register (PIPESEL): Description changed
		1314 to 1322	27.2.32 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9) : Bit table and description, changed
		1324	27.2.34 PIPEn Transaction Counter Registers (PIPEnTRN) (n = 1 to 5): Bit chart and description, changed
		1328 to 1331	27.2.37 Deep Standby USB Suspend/Resume Interrupt Register (DPUSR1R): Bit table and description, changed
		1332	27.3.1.3 Example of USB External Connection Circuit: Description changed
		1334	Figure 27.3 Functional Connection Sample of USB Connector (USB0) in Self-Powered State: Title and Figure, changed
		1335	Figure 27.4 Sample Host Connection of USB Connector (USB0), changed
		1336	Figure 27.5 Functional Connection Sample of USB Connector (USB0) in Bus-Powered State: Title changed
		1348	Figure 27.11 Timing of BRDY Interrupt Generation, changed
		1354	Table 27.14 USB Suspend/Resume Interrupt Sources and Corresponding I/O Pins, changed
		1356	Table 27.15 Interrupt Sources: Table and Note 1, changed
		1364	27.3.4.9 Auto Response Mode: Title and description, changed
		1367	Table 27.21 List of Buffer Clearing Methods, changed
			Section 28. Serial Communications Interface (SCIa)
		1388	Table 28.1 Specifications of SCI, changed
		1392, 1393	Table 28.4 Registers of SCI: Value after Reset in the SSR register, changed
		1394	28.2.2 Receive Data Register (RDR): Description changed
		1394	28.2.3 Transmit Data Register (TDR): Description changed
		1395	28.2.4 Transmit Shift Register (TSR): Description changed
		1395 to 1398	28.2.5 Serial Mode Register (SMR): Description, Bit table, and Note, changed
		1399 to 1403	28.2.6 Serial Control Register (SCR): Description, Bit table, and Note, changed
		1404 to 1409	28.2.7 Serial Status Register (SSR): Bit chart and description, changed
		1410	28.2.8 Smart Card Mode Register (SCMR): Bit table and description, changed

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	1416	Table 28.10 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode): Bit table, changed
		1418	28.2.10 Serial Extended Mode Register (SEMR): Bit table and description, changed
		1419	Figure 28.3 Example of Average Transfer Rate Setting when TMR Clock is Input, changed
		1420	28.3 Operation in Asynchronous Mode: Description changed
		1422	28.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode: Description changed
		1424	28.3.4 SCI Initialization (Asynchronous Mode): Description changed
		1424	Figure 28.7 Sample SCI Initialization Flowchart (Asynchronous Mode), changed
		1425	28.3.5 Serial Data Transmission (Asynchronous Mode): Description changed
		1425	Figure 28.8 Example of Operation for Serial Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit), changed
		1426	Figure 28.9 Example of Serial Transmission Flowchart in Asynchronous Mode, changed
		1427	Figure 28.10 Example of SCI Operation for Serial Reception in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit), changed
		1428	Figure 28.11 Example of Serial Reception Flowchart (1) (Asynchronous Mode), changed
		1429	Figure 28.11 Example of Serial Reception Flowchart (2) (Asynchronous Mode), changed
		1430	28.4 Multi-Processor Communications Function: Description changed
		1431	28.4.1 Multi-Processor Serial Data Transmission: Description changed
		1431	Figure 28.13 Example of Multi-Processor Serial Transmission Flowchart, changed
		1432	28.4.2 Multi-Processor Serial Data Reception: Description changed
		1432	Figure 28.14 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/One Stop Bit), changed
		1433	Figure 28.15 Example of Multi-Processor Serial Reception Flowchart (1), changed
		1434	Figure 28.16 Example of Multi-Processor Serial Reception Flowchart (2), changed
		1436	28.5.2 SCI Initialization (Clock Synchronous Mode): Description changed
		1437	28.5.3 Serial Data Transmission (Clock Synchronous Mode): Description changed
		1439	Figure 28.21 Example of Operation for Serial Reception in Clock Synchronous Mode, changed
		1440	Figure 28.22 Example of Serial Reception Flowchart (Clock Synchronous Mode): Description changed
		1441	28.5.5 Serial Data Full-Duplex Communications (Clock Synchronous Mode): Description changed
		1442	Figure 28.23 Example of Simultaneous Serial Transmission and Reception Flowchart (Clock Synchronous Mode), changed
		1443	28.6.1 Sample Connection: Description changed
		1444	28.6.2 Data Format (Except in Block Transfer Mode): Description changed
		1445	28.6.3 Block Transfer Mode: Description changed
		1446	28.6.4 Receive Data Sampling Timing and Reception Margin: Description changed
		1447	28.6.5 Initialization of the SCI: Description changed
		1448	28.6.6 Serial Data Transmission (Except in Block Transfer Mode): Description changed
		1451	28.6.7 Serial Data Reception (Except in Block Transfer Mode): Description changed
1451	Figure 28.32 Data Retransfer Operation in SCI Reception Mode, changed		
1453	28.6.8 Clock Output Control: Description changed		
1453	Figure 28.34 Clock Output Fixing Timing: Title changed		
1455	28.7.1 Interrupts in Serial Communications Interface Mode: Description changed		
1456	28.7.2 Interrupts in Smart Card Interface Mode: Description changed		
1457	28.8.2 Break Detection and Processing: Description changed		

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	1457	28.8.3 Mark State and Break Detection: Description changed
		1457	28.8.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only): Description changed
		1457	28.8.5 Writing Data to TDR: Title and description, changed
		1457	28.8.6 Restrictions on Clock Synchronous Transmission, changed
		1458	28.8.8 SCI Operations during Low Power Consumption State, (1) Transmission, (2) Reception: Description changed
		1459	Figure 28.36 Example of Flowchart for Transition to Software Standby Mode during Transmission, changed
		1460	Figure 28.37 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission), changed
		1460	Figure 28.38 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission), changed
		1461	Figure 28.39 Example of Flowchart for Transition to Software Standby Mode during Reception, changed
			Section 29. CRC Calculator (CRC)
		1463	29.2.1 CRC Control Register (CRCCR): Bit table and description, changed
			Section 30. I ² C Bus Interface (RIIC)
		1469	Table 30.1 RIIC Specifications, changed
		1474	30.2.1 I ² C Bus Control Register 1 (ICCR1): Bit table, description, and Note, changed
		1477	30.2.2 I ² C Bus Control Register 2 (ICCR2): Bit table, description, and Note, changed
		1481	30.2.3 I ² C Bus Mode Register 1 (ICMR1): Bit table, description, and Note, changed
		1483	30.2.4 I ² C Bus Mode Register 2 (ICMR2): Bit table, description, and Note, changed
		1485	30.2.5 I ² C Bus Mode Register 3 (ICMR3): Bit table, description, and Note, changed
		1488	30.2.6 I ² C Bus Function Enable Register (ICFER): Bit table and description, changed
		1490	30.2.7 I ² C Bus Status Enable Register (ICSER): Bit table and description, changed
		1492	30.2.8 I ² C Bus Interrupt Enable Register (ICIER): Bit table and description, changed
		1494	30.2.9 I ² C Bus Status Register 1 (ICSR1): Bit table, description, and Note, changed
		1498	30.2.10 I ² C Bus Status Register 2 (ICSR2): Bit table, description, and Note, changed
		1502	30.2.11 Slave Address Register Lm (SARLm) (m = 0 to 2): Bit table and description, changed
		1503	30.2.12 Slave Address Register Um (SARUm) (m = 0 to 2): Bit table and description, changed
		1504	30.2.13 I ² C Bus Bit Rate Low-Level Register (ICBRL): Bit table and description, changed
		1505	30.2.14 I ² C Bus Bit Rate High-Level Register (ICBRH): Bit table, description, and Note, changed
		1510	30.3.3 Master Transmitter Operation: Description changed
		1514	30.3.4 Master Receiver Operation: Description changed
		1516	Figure 30.10 Example of Master Reception Flowchart (7-Bit Address Format), changed
		1519	30.3.5 Slave Transmitter Operation: Description changed
		1520	Figure 30.14 Example of Slave Transmission Flowchart, changed
		1522	30.3.6 Slave Receiver Operation: Description changed
		1525	30.4 SCL Synchronization Circuit: Description changed
		1527	30.6 Digital Noise-Filter Circuits: Description changed
		1527	Figure 30.22 Block Diagram of Digital Noise Filter Circuit, changed
		1528	30.7 Address Match Detection: Description changed
		1528	30.7.1 Slave-Address Match Detection: Description changed
		1530	30.7.2 Detection of the General Call Address: Title and description, changed
		1531	30.7.3 Device-ID Address Detection: Description changed

Rev.	Date	Description			
		Page	Summary		
1.00	Sep. 07, 2010	1533	30.7.4 Host Address Detection: Description changed		
		1534	30.8.1 Function to Prevent Wrong Transmission of Transmit Data: Description changed		
		1535	30.8.2 NACK Reception Transfer Suspension Function: Description changed		
		1536	30.8.3 Function to Prevent Failure to Receive Data: Description changed		
		1537	Figure 30.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits), changed		
		1538	30.9.1 Master Arbitration Lost Detection (MALE Bit): Description changed		
		1539	Figure 30.33 Arbitration Lost when a Start Condition is Issued (MALE = 1), changed		
		1540	30.9.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit): Description changed		
		1542	30.9.3 Slave Arbitration Lost Detection (SALE Bit): Description changed		
		1542	30.10.1 Issuing a Start Condition: Description changed		
		1542	30.10.2 Issuing a Restart Condition: Description changed		
		1543	Figure 30.36 Start Condition/Restart Condition Issue Timing (ST and RS Bits), changed		
		1543	30.10.3 Issuing a Stop Condition: Description changed		
		1543	Figure 30.37 Stop Condition Issue Timing (SP Bit), changed		
		1544	30.11 Bus Hanging: Description changed		
		1544	30.11.1 Timeout Detection Function: Description changed		
		1545	Figure 30.38 Timeout Detection Function (TMOE, TMOS, TMOH, and TMOL Bits), changed		
		1546	30.11.2 Extra SCL Clock Cycle Output Function: Description changed		
		1546	Figure 30.39 Extra SCL Clock Cycle Output Function (CLO Bit), changed		
		1547	30.11.3 RIIC Reset and Internal Reset: Description changed		
		1548	30.12 SMBus Operation: Description changed		
		1548	30.12.1 SMBus Timeout Measurement, changed		
		1549	Figure 30.40 SMBus Timeout Measurement, changed		
		1550	30.12.3 SMBus Host Notification Protocol/Notify ARP Master: Description changed		
		1551	30.13 Interrupt Request: Description changed		
		1551	Table 30.7 Interrupt Sources, changed		
		1553	30.15.2 Setting Input Buffer Control Register: Description changed		
		—	30.15.3 Timings for Writing and Outputting of Transmit Acknowledge Bit, deleted		
		—	30.15.4 Restrictions on Timings for Stop Condition Issuance Request and Transmit Data Writing in Master Transmitter Mode, deleted		
					Section 31. CAN Module (CAN)
				1554 to 1619	Addresses of CAN registers are described in 8 bits.
				1554	Table 31.1 Specifications of CAN Module, changed
				1558	31.2.1 CAN0 Control Register (C0CTRL): Bit chart, Bit table, Note 3. and description, changed
				1562	31.2.2 CAN0 Bit Configuration Register (C0BCR): Bit chart, Bit table, and description, changed
				1564	31.2.3 CAN0 Mask Register i (COMKRi) (i = 0 to 7): Bit chart and Bit table, changed
				1565	31.2.4 CAN0 FIFO Received ID Compare Registers 0 and 1 (C0FIDCR0 and C0FIDCR1): Bit chart, Bit table, and Note 1., changed
				1568	31.2.6 CAN0 Mailbox Register j (COMBj) (j = 0 to 31): Bit chart, changed
				1572	31.2.7 CAN0 Mailbox Interrupt Enable Register (COMIER): Bit chart, Bit table, and description, changed
				1574	31.2.8 CAN0 Message Control Register j (COMCTLj) (j = 0 to 31): Bit table and Note 1., changed

Rev.	Date	Description			
		Page	Summary		
1.00	Sep. 07, 2010	1585	31.2.13 CAN0 Status Register (C0STR): Bit chart, Bit table, and description, changed		
		1588	31.2.14 CAN0 Mailbox Search Mode Register (C0MSMR): Bit table, changed		
		1592	31.2.17 CAN0 Acceptance Filter Support Register (C0AFSR): Bit chart and Bit table, changed		
		1597	Table 31.8 Behavior of BOEIF and BORIF Flags according to C0CTLR.BOM[1:0] Bit Setting, changed		
		1600	31.2.23 CAN0 Time Stamp Register (C0TSR): Bit chart, changed		
		1603	Figure 31.9 Transition between CAN Operating Modes, changed		
		1604	31.3.1 CAN Reset Mode: Description changed		
		1605	31.3.2 CAN Halt Mode: Description changed		
		1605	Table 31.9 Operation in CAN Reset Mode and CAN Halt Mode: Table and Note, changed		
		1606	31.3.3 CAN Sleep Mode: Description changed		
		1606	31.3.4 CAN Operation Mode (Excluding Bus-Off State): Description changed		
		1607	31.3.5 CAN Operation Mode (Bus-Off State): Description changed		
				Section 32. Serial Peripheral Interface (RSPI)	
		1620	32.1 Overview: Description changed		
		1629	32.2.3 RSPI Pin Control Register (SPPCR): Description changed		
		1631	32.2.4 RSPI Status Register (SPSR): Bit chart, Bit table, and description, changed		
		1634	32.2.5 RSPI Data Register (SPDR): Bit table and bit description, changed		
		1637	Table 32.4 Relationship between SPBR and BRDV[1:0] Bit Settings: Note 1, changed		
		1638	32.2.9 RSPI Data Control Register (SPDCR): Description changed		
		1641	32.2.10 RSPI Clock Delay Register (SPCKD): Description changed		
		1642	32.2.11 RSPI Slave Select Negation Delay Register (SSLND): Description changed		
		1643	32.2.12 RSPI Next-Access Delay Register (SPND): Description changed		
		1644	32.2.13 RSPI Control Register 2 (SPCR2): Description changed		
		1646	32.2.14 RSPI Command Registers 0 to 7 (SPCMD0 to SPCMD7): Description changed		
		1653	32.3.3.1 Single Master/Single Slave (with This LSI Acting as Master): Description changed		
		1654	32.3.3.2 Single Master/Single Slave (with This LSI Acting as Slave): Description changed		
		1657	32.3.3.5 Multi-Master/Multi-Slave (with This LSI Acting as Master): Description changed		
		1658	32.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Slave): Description changed		
		1678	32.3.9.1 Initialization by Clearing the SPE Bit: Description changed		
		1679	32.3.10.1 Master Mode Operation, (3) Sequence Control: Description changed		
		1681	(4) Burst Transfer: Description changed		
		1686	32.3.10.2 Slave Mode Operation, (3) Notes on Single-Slave Operations: Description changed		
		1697	Figure 32.34 Error Handling (Overrun Error), changed		
		1698	Figure 32.35 Error Handling (Parity Error), changed		
		1698	Figure 32.36 Error Handling (Mode Fault Error), changed		
		1699	32.3.15 Loopback Mode: Description changed		
		1701	32.4 Usage Note, added		
				Section 33. 12-Bit A/D Converter (S12AD)	
		1702	33.1 Overview: Description changed		
		1702	Table 33.1 Specifications of A/D Converter, changed		
		1703	Table 33.2 Functions of A/D Converter, changed		
		1704	Figure 33.1 Block Diagram of A/D Converter, changed		
1705	Table 33.4 Registers of A/D Converter, changed				

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	1708	33.2.2 A/D Control Register (ADCSR): Bit table and description, changed
		1711	33.2.4 A/D-Converted Value Addition Mode Select Register (ADADS): Description changed
		1712	33.2.5 A/D-Converted Value Addition Count Select Register (ADADC): Description changed
		1713	33.2.6 A/D Control Extended Register (ADCER): Bit table and description, changed
		1714	33.2.7 A/D Start Trigger Select Register (ADSTRGR): Bit table and description, changed
		1714	Table 33.5 List of A/D Conversion Startup Sources, changed
		1715	33.3.2 Single-Cycle Scan Mode: Description changed
		1716	Figure 33.4 Example of Operation of Continuous Scan Mode, changed
		1717	33.3.4 Analog Input Sampling and Scan Conversion Time: Description changed
		1717	Table 33.6 Scan Conversion Time, changed
		1718	33.3.5 Usage Example of A/D Data Register n (ADDRn) Automatic Clearing Function: Description changed
		1718	33.3.7 Starting Scan Conversion with External Trigger: Description changed
		1719	33.3.8 Starting Scan Conversion with Trigger from Peripheral Modules: Description changed
		1719 to 1724	33.3.8.1 A/D Converter Activation with TRG0AN_0 and TRG0BN_0 of MTU to 33.3.8.5 A/D Converter Activation with TMTRG0AN_0 and TMTRG0AN_1 of TMR, added
		1726	33.5.3 Notes on Restarting A/D Conversion: Description changed
		1726	33.5.5 Notes on Entering Low Power Consumption States: Description changed
		1727	33.5.7 Notes on Using A/D Converter and D/A Converter Simultaneously, added
			Section 34. 10-Bit A/D Converter (ADa)
		1728	Table 34.1 Specifications of A/D Converter, changed
		1729	Table 34.2 Comparison of Functions by Each Unit, changed
		1730	Figure 34.1 Block Diagram of A/D Converter Unit 0 (AD0), changed
		1731	Figure 34.2 Block Diagram of A/D Converter Unit 1 (AD1), changed
		1732	Table 34.3 Input Pins of A/D Converter, changed
		1733	Table 34.4 Registers of A/D Converter: Name of the ADDPR register, changed
		1737	34.2.3 A/D A/D Control Register (ADCR): Bit table and description, changed
		1739	34.2.4 ADDRn Format Select Register (ADDPR): Bit table and description, changed
		1740	34.2.6 A/D Self-Diagnostic Register (ADDIAGR): Description changed
		1743	34.3.2.1 Continuous Scan Mode: Description changed
		1745	34.3.2.2 One-Cycle Scan Mode: Description changed
		1746	34.3.3 Input Sampling and A/D Conversion Time: Description changed
		1746	Figure 34.6 A/D Conversion Timing, changed
		1748	34.3.4 A/D Converter Activation by External Triggers: Description changed
		1749 to 1753	34.3.5 A/D Converter Activation with TRG0AN_0 and TRG0BN_0 of MTU to 34.3.8 A/D Converter Activation with TMTRG0AN_0 of TMR: Description changed
		1754	34.4 Interrupt Source: Description changed
		1756	34.6.3 Notes on Restarting A/D Conversion: Description changed
		1756	34.6.4 Notes on Entering Power-Down States: Description changed
		1757	34.6.5 Permissible Impedance of Signal Sources: Description changed
		1757	34.6.6 Factors Affecting Absolute Accuracy: Description changed
		1758	34.6.7 Ranges of Settings for Analog Power Supply and Other Pins: Description changed
		1758	Figure 34.15 Example of Connections for AVCC = VCC and AVSS = VSS = VREFL, changed
		1758	34.6.8 Point for Caution Regarding Board Design: Description changed
		1760	34.6.10 Realizing High-Speed Conversion: Description changed

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	1761	34.6.12 Notes on Using A/D Converter and D/A Converter Simultaneously, added
			Section 35. D/A Converter
		1762	Table 35.2 Pin Configuration of D/A Converter, changed
		1763	Table 35.3 Registers of D/A Converter, changed
		1764	35.2.2 D/A Control Register (DACR): Bit table and description, changed
		1765	35.2.3 DADRM Format Select Register (DADPR): Bit table and description, changed
		1767	35.4.4 Note on Entering Deep Software Standby Mode: Description changed
			Section 36. RAM
		1768	Table 36.1 Specifications of the RAM, changed
			Section 37. ROM (Flash Memory for Code Storage)
		1769	Table 37.1 Specifications of the ROM, changed
		1770	Figure 37.1 Block Diagram of ROM, changed
		1774	37.2.2 Flash Access Status Register (FASTAT): Bit table and description, changed
		1778	37.2.5 Flash Status Register 0 (FSTATR0): Bit table and description, changed
		1783	37.2.8 Flash P/E Mode Entry Register (FENTRYR): Bit table and description, changed
		1788	37.2.12 FCU Processing Switching Register (FCPSR): Bit table and description, changed
		1789	37.2.13 Flash P/E Status Register (FPESTAT): Bit table and description, changed
		1790	37.2.14 Peripheral Clock Notification Register (PCKAR): Bit table, description, and Note, changed
		1791	37.2.15 Flash Write Erase Protection Register (FWEPROR): Bit table and description, changed
		1792	37.3 Configuration of Memory Mats for the ROM, changed
		1792	Figure 37.2 Memory Mat Configuration of ROM, changed
		1794	Figure 37.4 Transitions between Operating Modes in Terms of the ROM, changed
		1795	Table 37.5 Differences between Modes, changed
		1798	37.6 Programming and Erasing the ROM: Description changed
		1799	Table 37.7 FCU Command Formats: Table and Note changed
		1802	Figure 37.7 Procedure for Transition to ROM Read Mode, changed
		1805	Figure 37.11 Simple Flowchart of the Procedure for Programming and Erasure: Figure, Notes 1 and 2, changed
		1807	(4) Using the Peripheral Clock Notification Command: Description changed
		1808	Figure 37.13 Using the Peripheral Clock Notification Command, changed
		1811	(6) Erasure: Description changed
		1811	Figure 37.15 Procedure for ROM Erasure, changed
		1812	(7) Programming/Erasing to Lock Bit: Description changed
		1813	(8) Reading Lock Bits: Description changed
		1815	37.6.4.4 Suspension and Resumption, (1) Suspending Programming or Erasure: Description changed
1816	Figure 37.19 Procedure for Programming/Erasure Suspension, changed		
1817	Figure 37.20 Procedure for Resuming Programming or Erasure, changed		
1818	37.7 Suspending Operation: Description changed		
1818	37.7.1 Suspension during Programming: Description changed		
1821	37.8.1 Software Protection: Description changed		
1821	37.8.2 Error Protection: Description changed		
1822	Table 37.9 Error Protection Types (Types Dedicated to ROM and Types Common to ROM and Data Flash): Contents changed		

Rev.	Date	Description			
		Page	Summary		
1.00	Sep. 07, 2010	1823	Figure 37.24 System Configuration for Operations in Boot Mode, changed		
		1824	37.9.2 ID Code Protection, (2) ID Code: Description changed		
		1826	37.9.3 UB Code A, added		
		1827	Figure 37.27 State Transitions in Boot Mode, changed		
		1827	37.9.4 State Transitions in Boot Mode, (3) Judging ID Code Protection: Description changed		
		1828	(5) Waiting for a Host Command for Programming or Erasure: Description changed		
		1829	Table 37.12 Conditions for Automatic Bit-Rate Adjustment to be Possible, changed		
		1830	Table 37.13 Inquiry/Selection Host Commands, changed		
		1830	37.9.6 Inquiry/Selection Host Command Wait State, (7) User Boot Mat Information Inquiry, added		
		1831	Figure 37.30 Example of Procedure to Use Inquiry/Selection Host Commands for User Mat and User Boot Mat, changed		
		1842	(12) Programming/Erasure State Transition: Description changed		
		1843	Table 37.14 Status Code, changed		
		1845	Table 37.17 Programming/Erasure Host Commands, changed		
		1845	37.9.8 Programming/Erasure Host Command Wait State, (1) User Boot Mat Programming Selection, added		
		1846	Figure 37.32 Procedure for ROM Programming in Boot Mode, changed		
		1850	(7) User Boot Mat Checksum, added		
		1851	(9) User Boot Mat Blank Check, added		
		1854	37.10 USB (User) Boot Mode, changed		
		1856	Figure 37.35 State Transitions in USB Boot Mode, changed		
		1859	Table 37.20 Specifications for ROM Code Protection, changed		
		1859, 1860	37.13 Usage Notes, (2) Suspending Programming or Erasure, (3) Prohibition of Reprogramming, (5) Prohibition of Non-Maskable Interrupts during Programming or Erasure, (8) Actions Prohibited during Programming and Erasure, changed		
		1860	(7) Abnormal Termination of Programming and Erasure, added		
					Section 38. Data Flash Memory (Flash Memory for Data Storage)
				1861	Table 38.1 Specifications of Data Flash Memory, changed
				1862	Figure 38.1 Block Diagram of Data Flash Memory, changed
				1869	38.2.4 Data Flash Read Enable Register 0 (DFLRE0): Description changed
				1871	38.2.6 Data Flash Programming/Erasure Enable Register 0 (DFLWE0): Description changed
				1872	38.2.7 Data Flash Programming/Erasure Enable Register 1 (DFLWE1): Description changed
				1873	38.2.8 Flash P/E Mode Entry Register (FENTRYR): Bit table and description, changed
				1878	38.5 Operating Modes Associated with the Data Flash: Description changed
				1878	Table 38.4 Differences between Modes, changed
				1879	38.6 Programming and Erasing the Data Flash Memory: Description changed
				1881	Table 38.5 FCU Commands for Use with Data Flash Memory, changed
		1883	38.6.3 Connections between FCU Modes and Commands: Description changed		
		1885	Figure 38.5 Procedure for Data Flash Programming, changed		
		1887	Figure 38.6 Procedure for Blank Checking of the Data Flash, changed		
		1888	38.7.1 Software Protection, (2) Protection through FENTRYR to (4) Protection through DFLREK: Description changed		
		1889	38.7.2 Error Protection: Description changed		

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 07, 2010	1892	38.8.2 Programming/Erasing Host Commands: Description changed
		1894	38.9 Usage Notes, (2) Other Points to Note, changed
		1895 to 1860	Section 39. Boundary Scan, added
		1916 to 1970	Section 40. Electrical Characteristics, added
		1971 to 1976 1977 to 1980	Appendix1. Port States in Each Processing Mode: Title and contents, changed Table 1.1 Port States in Each Processing State, changed Table 1.2 Conditions when the Port State is in High-Impedance by the POE Function Control, added
1.10	Jan.24, 2011	All	Name of the on-chip oscillator clock, changed: OCOCLK?IWDTCCLK
		All	Register symbols, changed
		44	Section 1. Overview
		51	Table 1.1 Outline of Specifications: Note added
		87	Figure 1.2 Block Diagram: Note added
		87	Table 1.9 Pin Functions, USB 2.0 host/function module: Pin description, changed
		114	Section 2. CPU
		120	Table 2.14 Instructions that are Converted into Multiple Micro-Operations: Symbol description, added
		120	2.8.4 Numbers of Cycles for Response to Interrupts: Description changed
		120	Table 2.15 Numbers of Cycles for Response to Interrupts: Description changed
		131	Section 5. I/O Registers
		133 to 167	At the beginning of the section, (1) I/O register addresses (address order): Description added Table 5.1 List of I/O Registers (Address Order): Items in the table, changed, Register symbols, changed
		168	Table 5.2 I/O Register Bits, changed
		225	Section 6. Resets
		225	6.3.4 Deep Software Standby Reset: Description changed
		237	Section 8. Clock Generation Circuit
238	Table 8.1 Specifications of Clock Generation Circuit: Notes 2 and 3, added		
253	Table 8.2 Pin Configuration: Description changed 8.12.1 Notes on Clock Generation Circuit: Description changed		
257	Section 9. Low Power Consumption		
285	Figure 9.1 Mode Transitions: Note added, Figure changed 9.5.4.1 Transition to Deep Software Standby Mode: Note changed		
296	Section 10. Exceptions		
296	Figure 10.2 Outline of the Exception Handling Procedure, changed		
323	Section 11. Interrupt Control Unit (ICUa)		
325	11.2.7 DMACA Activation Source Select Register n (DMRSRn) (n = DMACA channel number): Bit table and bit description, changed		
330 to 335	11.2.9 Non-Maskable Interrupt Status Register (NMISR): Description changed Table 11.4 Interrupt Vector Table, changed		
336	11.4.1.1 Operation of Status Flags for Edge-Detected Interrupts: Description added		
336	Figure 11.2 Operation of the IRi.IR Flag in the Case of Edge Detection, changed		
338	Figure 11.5 Operation of the IRi.IR Flag in the Case of a Level-Detected Interrupt, changed		
338	Figure 11.6 Level-Detected Interrupt Handling Procedure, changed		
340	11.4.3 (2) DTC Activation: Description added		
348	Section 12. Buses		
352	Table 12.1 Bus Specifications: Note description, changed 12.2.5 External Bus: Description changed		

Rev.	Date	Description			
		Page	Summary		
1.10	Jan.24, 2011	378	12.3.15 SDRAM Timing Register (SDTR): Note on the description of the CL[2:0] bits, added		
		382	12.3.18 Bus Error Status Clear Register (BERCLR): Description on bit 0, changed		
		441	Figure 12.65 Timing Example of Single Write (4), changed		
				Section 13. DMA Controller (DMACA)	
		476	13.2.13 DMA Module Start Register (DMAST): Title changed		
		477	Table 13.4 Register Update Operation in Normal Transfer Mode, changed		
		478	13.3.1 (2) Repeat Transfer Mode: Description changed		
		478	Table 13.5 Register Update Operation in Repeat Transfer Mode, changed		
		483	Table 13.7 Address Update Method in Each Address Update Mode, changed		
		489	Figure 13.10 Register Setting Procedure, changed		
		492	Figure 13.11 DMAC Operation Timing Example (1) (DMA Started by Interrupt from Peripheral Module or External Interrupt Input Pin, in Normal Transfer Mode or Repeat Transfer Mode), changed		
		496	13.5 Interrupts (1) to (3): Note added, Description changed		
				Section 14. EXDMA Controller (EXDMAC)	
		515	14.2.10 EXDMA Transfer Enable Register (EDMCNT): Description added		
		532	Table 14.9 Address Update Method in Each Address Update Mode, changed		
		535	Figure 14.10 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode, changed		
		537	Figure 14.11 Bus Cycle Example in Normal-Transfer Dual Address Mode, changed		
		538	Figure 14.12 Data Flow in Single Address Mode (when DIR = 1): Title changed		
		538	Figure 14.13 Bus Cycle Example in Normal-Transfer Single Address Mode, changed		
		539	Figure 14.14 Bus Cycle Example in Block-Transfer Dual Address Mode, changed		
		540	Figure 14.15 Data Flow in Block Transfer Single Address Mode, changed		
		541	Figure 14.17 Bus Cycle Example in Cluster-Transfer Dual Address Mode, changed		
		542	Figure 14.19 Bus Cycle Example in Cluster-Transfer Read Address Mode, changed		
		543	Figure 14.21 Bus Cycle Example in Cluster-Transfer Write Address Mode, changed		
		544	14.5 Activation Sources and Procedures for Activation, 14.5.1 Activation Sources: Level of the headings, changed		
		544	14.5.1 Activation Sources, (2) EXDMAC Activation by External DMA Request Pin (EDREQ), changed		
		543	14.5.1 Activation Sources, (3) EXDMAC Activation by DMA Transfer Requests from On-Chip Peripheral Modules (MTU1 Compare Match), changed		
		551	14.7 Interrupts: Description changed		
		552	14.7 (1) When Discontinuing or Terminating DMA Transfer: Description changed		
		552	Figure 14.26 Restarting or Suspending DMA Transfer from within the Processing Routine for the EXDMAC Interrupt: Title changed		
				Section 15. Data Transfer Controller (DTCa)	
		587	15.4.6 Chain Transfer: Description changed		
				Section 16. I/O Ports	
All	Port module symbol, changed: Pn ? PORTn				
601	Table 16.2 Port Functions (176-Pin LFBGA), changed				
620	16.1.2.7 Port Function Control Register 0 (PFOCSE): Bit name, changed				
623	16.1.2.10 Port Function Control Register 3 (PF3BUS), changed				
624	16.1.2.11 Port Function Control Register 4 (PF4BUS), changed				
625	16.1.2.12 Port Function Control Register 5 (PF5BUS), changed				
626	16.1.2.13 Port Function Control Register 6 (PF6BUS), changed				
636	16.1.2.23 Port Function Control Register G (PFGSPI), changed				
638	16.1.2.24 Port Function Control Register H (PFHSPI), changed				
645	16.1.2.28 Port Function Control Register M (PFMPOE), changed				
646	16.1.2.29 Port Function Control Register N (PFNPOE), changed				
648 to 661	Table 16.10 Output Enable Settings for Each Port: MTU settings, changed				

Rev.	Date	Description	
		Page	Summary
1.10	Jan.24, 2011	662 to 667	Table 16.11 Settings to Enable Output on the Various MTU Pins, added
		688	16.2.2.7 Port Function Control Register 0 (PF0CSE), changed
		691	16.2.2.10 Port Function Control Register 3 (PF3BUS), changed
		692	16.2.2.11 Port Function Control Register 4 (PF4BUS), changed
		694	16.2.2.13 Port Function Control Register 6 (PF6BUS), changed
		705 to 706	16.2.2.23 Port Function Control Register G (PFGSPI), changed
		707 to 708	16.2.2.24 Port Function Control Register H (PFHSPI), changed
		712	16.2.2.27 Port Function Control Register M (PFMPOE), changed
		713	16.2.2.28 Port Function Control Register N (PFNPOE), changed
		715 to 726	Table 16.21 Output Enable Settings for Each Port: MTU settings, changed
		727 to 732	Table 16.22 Settings to Enable Output on the Various MTU Pins, added
		750	16.3.2.7 Port Function Control Register 0 (PF0CSE), changed
		751	16.3.2.8 Port Function Control Register 3 (PF3BUS), changed
		752	16.3.2.9 Port Function Control Register 4 (PF4BUS), changed
		753	16.3.2.10 Port Function Control Register 5 (PF5BUS), changed
		762 to 763	16.3.2.19 Port Function Control Register G (PFGSPI), changed
		764 to 765	16.3.2.20 Port Function Control Register H (PFHSPI), changed
		766	16.3.2.21 Port Function Control Register J (PFJCAN), changed
		769	16.3.2.23 Port Function Control Register M (PFMPOE), changed
		720	16.3.2.24 Port Function Control Register N (PFNPOE), changed
		772 to 781	Table 16.32 Output Enable Settings for Each Port: MTU settings, changed
		782 to 787	Table 16.33 Settings to Enable Output on the Various MTU Pins, added
		802	16.4.2.7 Port Function Control Register 0 (PF0CSE), changed
		803	16.4.2.8 Port Function Control Register 3 (PF3BUS), changed
		804	16.4.2.9 Port Function Control Register 4 (PF4BUS), changed
		809	16.4.2.14 Port Function Control Register G (PFGSPI), changed
		810	16.4.2.15 Port Function Control Register H (PFHSPI), changed
		811	16.4.2.16 Port Function Control Register J (PFJCAN), changed
		815 to 821	Table 16.42 Output Enable Settings for Each Port: MTU settings, changed
		822 to 827	Table 16.43 Settings to Enable Output on the Various MTU Pins, added
			Section 17. Multi-Function Timer Pulse Unit 2 (MTU2)
		838 to 839	Figure 17.1 Block Diagram of MTU (Unit 0) (1), Figure 17.1 Block Diagram of MTU (Unit 1) (2), changed
		842 to 846	Table 17.5 Registers of MTU, changed, Symbol of the MTU common register, changed
		847 to 850	17.2.1 Timer Control Register (TCR): Note added
		852	17.2.2 Timer Mode Register (TMDR): Bit BFA (Buffer Operation A), changed
		852 to 853	17.2.2 Timer Mode Register (TMDR): Bit BFB (Buffer Operation B), changed
		874 to 876	17.2.5 Timer Interrupt Enable Register (TIER) (MTU5, MTU11): Description changed
		880 to 881	17.2.9 Timer A/D Converter Start Request Control Register (TADCR): Note changed
		881	17.2.10 Timer A/D Converter Start Request Cycle Set Register (TADCORA, TADCORB): Title changed
		881	17.2.11 Timer A/D Converter Start Request Cycle Set Buffer Register (TADCORA, TADCORB): Title changed
		882	17.2.13 Timer General Register (TGR): Description changed
		883 to 885	17.2.14 Timer Start Register (TSTR): Title changed
		886	17.2.16 Timer Read/Write Enable Registers (TRWERA and TRWERB): Description changed
		887	17.2.17 Timer Output Master Enable Registers (TOERA and TOERB), changed
		888 to 889	17.2.18 Timer Output Control Registers 1 (TOCR1A and TOCR1B): Note deleted, R/W changed
		892	Table 17.40 Setting of TOCR2.BF[1:0] Bits, changed

Rev.	Date	Description	
		Page	Summary
1.10	Jan.24, 2011	897 to 898	17.2.26 Timer Interrupt Skipping Set Registers (TITCRA and TITCRB): Note changed
		917 to 921	17.3.4 Cascaded Operation: Description changed
		924	Figure 17.25 Example of PWM Mode Setting Procedure, changed
		934	Figure 17.35 Procedure for Selecting Reset-Synchronized PWM Mode (Unit 0), changed
		939	Figure 17.38 Example of Complementary PWM Mode (Unit 0) Setting Procedure, changed
		966 to 969	17.3.8 (3) Interrupt Skipping in Complementary PWM Mode: Description changed
		966	Figure 17.67 Example of Interrupt Skipping Operation Setting Procedure, changed
		967	Figure 17.69 Example of Interrupt Skipping Operation (Unit 0), changed
		967	(C) Buffer Transfer Control Linked with Interrupt Skipping, changed
		970	17.3.8 (4) (a) Register and Counter Miswrite Prevention Function: Description changed
		973	Figure 17.75 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (Unit 0) (when TRG4AN Output is Enabled during TCNT Up-Counting and Down-Counting): Title changed
		973	Figure 17.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (Unit 0) (when TRG4AN Output is Enabled during TCNT Up-Counting): Title changed
		983	Figure 17.85 Count Timing in External Clock Operation (Channels 0 to 4 and 6 to 10), changed
		983	Figure 17.86 Count Timing in External Clock Operation (Phase Counting Mode), changed
		990	Figure 17.99 TGI Interrupt Timing (Compare Match) (Channels 0 to 4 or 6 to 10), changed
		990	Figure 17.100 TGI TGI Interrupt Timing (Compare Match) (Channel 5 or 11), changed
		991	Figure 17.101 TGI Interrupt Timing (Input Capture) (Channels 0 to 4 or 6 to 10), changed
		991	Figure 17.102 TGI Interrupt Timing (Input Capture) (Channel 5 or 11), changed
		992	Figure 17.103 TCIV Interrupt Setting Timing, changed
		992	Figure 17.104 TCIU Interrupt Setting Timing, changed
		994	Figure 17.107 Contention between TCNT Write and Increment Operations, changed
		995	Figure 17.108 Contention between TGR Write Operation and Compare Match: Title changed
		998	Figure 17.113 Contention between TGR Write Operation and Input Capture (Channels 0 to 4 or 6 to 10): Title changed
		998	Figure 17.114 Contention between TGR Write Operation and Input Capture (Channel 5 or 11): Title changed
		1002	Figure 17.118 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode, changed
		1003	Figure 17.119 Overflow Flags in Reset-Synchronized PWM Mode, changed
		1004	17.6.17 Contention between Overflow/Underflow and Counter Clearing: Description changed
		1004	Figure 17.120 Contention between Overflow and Counter Clearing, changed
		1004	17.6.18 Contention between TCNT Write Operation and Overflow/Underflow: Description changed
		1004	Figure 17.121 Contention between TCNT Write Operation and Overflow: Title changed
		1006	17.7.2 Operation in Case of Re-Setting Due to Error during Operation: Description changed
		1007	17.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation: Description changed
			Section 18. Port Output Enable 2 (POE2)
		1034	Table 18.3 Pin Combinations, changed
			Section 19. Programmable Pulse Generator (PPG)
		1058	Table 19.4 Registers of PPG, changed
		1072 to 1073	19.2.6 PPG Output Mode Register (PMR): Description changed
			Section 20. 8-Bit Timer (TMR)
		1091	Table 20.4 Register Allocation and Symbol for 16-Bit Access, changed
		1092 to 1093	20.2.1 to 20.2.3: Symbol and address for 16-bit access, added

Rev.	Date	Description	
		Page	Summary
1.10	Jan.24, 2011	1102	20.4.2 Timing of Interrupt Flag Setting to 1 at Compare Match, changed
		1102	Figure 20.7 Timing of Interrupt Flag Setting to 1 at Compare Match, changed
		1104	20.4.5 Timing of Interrupt Flag Setting to 1 at Compare Match, changed
		1104	Figure 20.10 Timing of Clearance by External Reset (Rising Edge), changed
		1105	20.4.6 Timing of Clearance by External Reset (Rising Edge), changed
		1105	Figure 20.12 Timing of Overflow Interrupt Flag Setting to 1, changed
		1108	20.7.2 Notes on Setting Cycle, changed
			Section 21. Compare Match Timer (CMT)
		1120	21.4.2 Timing of Compare Match Interrupt Generation, changed
		1120	Figure 21.4 Timing of Compare Match Interrupt Flag Setting to 1, changed
			Section 24. Independent Watchdog Timer (IWDT)
		All	Name of the on-chip oscillator clock, changed
			Section 25. Ethernet Controller (ETHERC)
		1175	25.2.2 ETHERC Status Register (ECSR): Description on bit LCHNG, changed
		1186	25.2.21 Manual PAUSE Frame Register (MPR): Description changed
		1201	25.4 Connection to PHY-LSI: Description changed
		1201	Figure 25.19 Example of Connection to RHY-LSI (MII): Title and contents, changed
		1201	Figure 25.20 Example of Connection to RHY-LSI (RMII): Title and contents, changed
		1202	25.5.2 RMII_RX_ER Pin Input when RMII is Selected, added
			Section 26. Ethernet Controller Direct Memory Access Controller (EDMAC)
		1205	26.2.1 EDMAC Mode Register (EDMR): Description of the note, changed
		1227	26.3.1 Descriptor Lists and Data Buffers: Description changed
		1231	26.3.1.2 (1) Receive Descriptor 0 (RD0): Description on RFS, changed
		1235	26.3.4.1 Multi-Buffer Frame Transmit Processing: Description changed
		1236	26.3.4.2 Receive Processing in Case of Multi-Buffer Frame: Description changed
			Section 27. USB 2.0 Host/Function Module (USB)
		1246	27.2.2 System Configuration Status Register 0 (SYSSTS0): Bit description, changed
1247 to 1249	27.2.3 Device State Control Register 0 (DVSTCTR0), changed		
1252 to 1256	27.2.5 CFIFO Port Select Register (CFIFOSEL): Description on bit DCLRM, changed		
1266	27.2.12 SOF Output Configuration Register (SOFCFG), changed		
1270 to 1272	27.2.14 Interrupt Status Register 1 (INTSTS1): Bit table, changed		
1280	27.2.23 USB Request Index Register (USBINDX): Description changed		
1281	27.2.24 USB Request Length Register (USBLENG): Description changed		
1303	27.2.35 Device Address n Configuration Registers (DEVADDn) (n = 0 to 5): Description changed		
1321	Figure 27.10 Circuits Related to Interrupts, changed		
1324	27.3.3.1 (2) When BRDYM = 0 and BFRE = 1: Description deleted		
	Section 28. Serial Communications Interface (SCIA)		
All	Register symbol in smart card interface mode, changed to SMCI		
All	This section is separated according to the mode.		
1363	Table 28.1 Specifications of SCI, Note added		
1366	Table 28.3 Pin Configuration of SCI/SMCI: Title and contents, changed		
1384	Table 28.9 BRR Settings for Various Bit Rates (Clock Synchronous Mode), changed		
1391	Figure 28.7 Sample SCI Initialization Flowchart (Asynchronous Mode), changed		
1391	Figure 28.7 Sample SCI Initialization Flowchart (Asynchronous Mode), Note added		
1392	Figure 28.8 Example of Operation for Serial Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit), changed		
1394	Figure 28.10 Example of SCI Operation for Serial Reception in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit), changed		

Rev.	Date	Description	
		Page	Summary
1.10	Jan.24, 2011	1400	Figure 28.14 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/One Stop Bit), changed
		1401	Figure 28.15 Example of Multi-Processor Serial Reception Flowchart (1), changed
		1404	Figure 28.18 Example of SCI Initialization Flowchart (Clock Synchronous Mode), changed
		1405	Figure 28.19 Example of Operation for Serial Transmission in Clock Synchronous Mode, changed
		1407	Figure 28.21 Example of Operation for Serial Reception in Clock Synchronous Mode, changed
		1411 to 1432	28.3 Smart-Card Interface Mode, separated and added
		1430	Figure 28.33 Data Retransfer Operation in SMCI Reception Mode, changed
			Section 29. CRC Calculator (CRC)
		1441	29.2.1 CRC Control Register (CRCCR): R/W of bit DORCLR, changed
			Section 30. I ² C Bus Interface (RIIC)
		1449	Figure 30.2 Connections to the External Circuit by the I/O Pins (I ² C Bus Configuration Example), changed
		1478	30.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2): Bit name, changed
		1479	30.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2): Bit name, changed
		1481	30.2.14 I ² C Bus Bit Rate High-Level Register (ICBRH): Description on the duty cycle, changed
		1482	Figure 30.36 Start Condition/Restart Condition Issue Timing (ST and RS Bits), changed
		1523	30.11.2 Extra SCL Clock Cycle Output Function: Description changed
			Section 31. CAN Module (CAN)
		All	Register symbol of the CAN register, changed
		1533	Table 31.3 Registers of CAN Module, changed
		1544	Table 31.5 Mailbox Memory Mapping, changed
		1578	31.2.24 Test Control Register (TCR) (2) Self-Test Mode 0 (External Loopback): Description changed
1561 to 1563	31.2.13 Status Register (STR): Bit description, changed		
1586	Figure 31.13 Structure of MBj (j = 0 to 31): Title changed		
	Section 32. Serial Peripheral Interface (RSPI)		
1604 to 1605	32.2.3 RSPI Pin Control Register (SPPCR): Description changed		
1610	32.2.6 RSPI Sequence Control Register (SPSCR): Description changed		
1611	32.2.7 RSPI Sequence Status Register (SPSSR): Description on bit SPECM, changed		
1617	32.2.11 RSPI Slave Select Negation Delay Register (SSLND): Description changed		
1618	32.2.12 RSPI Next-Access Delay Register (SPND): Description changed		
1637	Figure 32.14 MSB First Transfer (2) (24-Bit Data, Parity Function Enabled), changed		
1643	32.3.6.2 Transmit Operations Only (SPCR.TXMD = 1): Description changed		
1646	Table 32.9 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function: Description changed		
1647	32.3.8.1 Overrun Error: Description changed		
1648	32.3.8.2 Parity Error: Description changed		
1649	32.3.8.3 Mode Fault Error: Description changed		
1667	Figure 32.32 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation), changed		
	Section 33. 12-Bit A/D Converter (S12AD)		
All	Name of the 12-bit A/D converter interrupt is unified into "S12ADI0"		
1674	Table 33.1 Specifications of A/D Converter: Names of interrupt sources, changed		
1675	Table 33.2 Functions of A/D Converter: Names of interrupts, changed		
1680 to 1681	33.2.2 A/D Control Register (ADCSR), changed		

Rev.	Date	Description	
		Page	Summary
1.10	Jan.24, 2011	1686	Figure 33.3 1 Example of Operation of Single-Cycle Scan Mode, changed
		1687	Figure 33.4 Example of Operation of Continuous Scan Mode, changed
		1688	33.3.4 Analog Input Sampling and Scan Conversion Time, changed
		1695	33.4 Interrupt Sources and DMA Transfer Request: Description changed
			Section 34. 10-Bit A/D Converter (ADa)
		1710	34.2.6 A/D Self-Diagnostic Register (ADDIAGR): Note changed
		1711	Figure 34.3 Example of A/D Converter Operation (Single Mode), changed
		1712	Figure 34.4 Example of A/D Converter Operation (Continuous Scan Mode), changed
		1713	Figure 34.5 Example of A/D Converter Operation (One-Cycle Scan Mode), changed
		1714	Figure 34.6 A/D Conversion Timing, changed
	Section 35. D/A Converter		
1732	35.2.2 D/A Control Register (DACR): Note 2. changed		
1735	Figure 35.2 Example of D/A Converter Operation, changed		
	Section 37. ROM (Flash Memory for Code Storage)		
1805	37.9.6 (9) Erasure Block Information Inquiry: Response changed		
1821	37.10.2 State Transitions: Description changed		
1821	Figure 37.35 State Transitions in USB Boot Mode, changed		
	Section 38. Data Flash Memory (Flash Memory for Data Storage)		
1839	Figure 38.3 Division of the Data Mat into Blocks, changed		
1850	38.7.1 Software Protection (3) Protection through DFLWEk, changed		
1850	38.7.1 Software Protection (4) Protection through DFLREk, changed		
	Section 39. Boundary Scan		
1863	39.2.1 Instruction Register (JTIR): Bit table, changed		
1877	39.3.2(6) HIGHZ [Instruction Code: 1000 0000b]: Title changed		
1878 to 1879	39.4 Usage Notes: Description changed		
	Section 40. Electrical Characteristics		
1881	Table 40.2 DC Characteristics: Table and note, changed		
1884	Table 40.5 Permissible Output Currents, changed		
1886	Table 40.8 Clock Timing, changed		
1887	Figure 40.1 BCLK Pin Output, SDCLK Pin Output Timing, changed		
1890	Table 40.7 Control Signal Timing: Table and note, changed		
1905	Table 40.13 Timing of On-Chip Peripheral Modules (1), changed		
1910	Table 40.16 Timing of On-Chip Peripheral Modules (4): Table and note, changed		
1914	Table 40.11 Timing of On-Chip Peripheral Modules (6), changed		
1914	Figure 40.25 I/O Port Input/Output Timing, changed		
1926	Table 40.19 Internal USB Full-Speed Characteristics (DP, DM Pin Characteristics), changed		
1.11	Feb.25, 2011	43	One Page Summary, added
1.12	Mar.24, 2011		Section 11. Interrupt Control Unit (ICUa)
		341	Table 11.5 Operation at the Time of DMACA/DTC Activation, changed
			Section 12. Buses
		354	12.2.7 Limitations, (2) Restrictions in relation to RMPA and string-manipulation instructions, added
		420	12.5.6 Notes on Usage, (6) Restrictions in relation to RMPA and string-manipulation instructions, changed
		420	12.5.6 Notes on Usage, (7) Restriction on Instruction Code, added
1.20	Jun.06, 2011		1. Overview
		44 to 47 82 to 88	Table 1.1 Outline of Specification, Description changed Table 1.9 Pin Functions, Description changed
			2. CPU
		107	2.5.5 Notes on the Allocation of Instruction Codes, Description added

Rev.	Date	Description	
		Page	Summary
1.20	Jun.06, 2011		5. I/O Registers
		133 to 167	Table 5.1 List of I/O Registers (Address Order), Description changed
			8. Clock Generation Circuit
		245	Table 8.5 Crystal Resonator Characteristics (Reference Values), changed
		248	Figure 8.7 Pin Handling when Sub-Clock is not Used, changed
		251	8.11.1 Detection of Oscillation Stop and Operation after the Detection, Description changed
			9. Low Power Consumption
		256	Table 9.2 Transition and Cancellation of the Mode and the State of Operation, Note changed
		257	Figure 9.1 Mode Transitions, Note changed
		267	9.2.5 Deep Standby Control Register (DPSBYCR), DPSBY bit: Description changed
		269	9.2.6 Deep Standby Wait Control Register (DPSWCR), changed
		270	9.2.7 Deep Standby Interrupt Enable Register (DPSIER), changed
		271	9.2.8 Deep Standby Interrupt Flag Register (DPSIFR), changed
		274	9.2.10 Reset Status Register (RSTSR), changed
		280	9.5.2.2 Canceling All-Module Clock Stop Mode, changed
		282	9.5.3.2 Canceling Software Standby Mode, Description changed
		285	9.5.4.1 Transition to Deep Software Standby Mode, changed
286	9.5.4.2 Canceling Deep Software Standby Mode, changed		
293	9.7.6 Conflict between Transition to Deep Software Standby Mode and Interrupt, changed		
	11. Interrupt Control Unit (ICUa)		
304	Table 11.1 Specifications of the Interrupt Control Unit, changed		
305	Figure 11.1 Block Diagram of Interrupt Control Unit, changed		
324	11.2.8 IRQ Control Register n (IRQCRn) (n = 0 to 15), Description added		
325	11.2.9 Non-Maskable Interrupt Status Register (NMISR), changed		
327	11.2.10 Non-Maskable Interrupt Enable Register (NMIER), changed		
330 to 335	Table 11.4 Interrupt Vector Table, changed		
341	Table 11.5 Operation at the Time of DMACA/DTC Activation, Note changed		
343	11.5 Non-maskable Interrupt Operation, changed		
	12. Buses		
353	Table 12.6 Pin Configuration of the External Bus, Note 2 added		
354	12.2.7 Limitations, (2) added		
357	12.3.1 CSn Control Register (CSnCR) (n = 0 to 7), Notes 3, 4 added		
357	12.3.1 CSn Control Register (CSnCR) (n = 0 to 7), EMODE bit description added		
360	Table 12.9 Valid and Invalid Control Signals in Write Access, changed		
365	12.3.5 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 7), changed		
418	12.5.6 Notes on Usage, (4) Limitations on EXDMAC Single Address Transfer Mode, changed		
419	12.5.6 Notes on Usage, (6) Restrictions on RMPA and String-Manipulation Instructions , Description added		
419	12.5.6 Notes on Usage, (7) Restriction on Instruction Code, Description added		
	12.6.15 Restrictions, (5) and (6), added		
	13. DMA Controller (DMACA)		
463	13.2.5 DMA Transfer Mode Register (DMTMD), changed		
477	13.3.1 Transfer Mode, (2) Repeat Transfer Mode, changed		
478	Figure 13.3 Operation in Repeat Transfer Mode, changed		
481	Figure 13.6 Example of Extended Repeat Area Function in Block Transfer Mode, changed		
482	13.3.3 Address Update Function using Offset, (2) Example of XY Conversion Using Offset Addition, changed		
485	Figure 13.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode, changed		
488	Figure 13.10 Register Setting Procedure, changed		
492	Table 13.9 DMACA Execution Cycles, Note 1 added		

Rev.	Date	Description			
		Page	Summary		
1.20	Jun.06, 2011		14. EXDMA Controller (EXDMAC)		
		498	Table 14.1 Specifications of EXDMAC, changed		
		499	Table 14.2 Pin Configuration of EXDMAC, changed		
		505	14.2.5 EXDMA Transfer Mode Register (EDMTMD), changed		
		524	14.3.1 Transfer Mode, (2) Repeat Transfer Mode, changed		
		525	Figure 14.3 Operation in Repeat Transfer Mode, changed		
		525	14.3.1 Transfer Mode, (3) Block Transfer Mode, changed		
		526	Figure 14.4 Operation in Block Transfer Mode, changed		
		530	Figure 14.7 Example of Extended Repeat Area Function in Block Transfer Mode, changed		
		531	14.3.3 Address Update Function using Offset, (2) Example of XY Conversion Using Offset Addition, changed		
		534	Figure 14.10 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode, changed		
		536	14.4 Transfer Operation, Description added		
		543	14.5.1 Activation Sources, changed		
		544	Figure 14.22 Timing of an External DMA Transfer Request by Falling-Edge Detection, changed		
		544	Figure 14.23 Timing of an External DMA Transfer Request by Low-Level Detection, changed		
		545	Figure 14.24 Register Setting Procedure, changed		
		553	14.9.1 EDACK Operation Example in Normal-Transfer (CS Area) Single Address Mode, changed		
		553	Figure 14.27 Operation Example in Normal-Transfer (CS Area Read) Single Address Mode, changed		
		554	Figure 14.28 Operation Example in Normal-Transfer (CS Area Write) Single Address Mode, changed		
		557	14.9.3 EDACK Operation Example in Block-Transfer (CS Area) Single Address Mode, changed		
		557	Figure 14.31 Operation Example in Block-Transfer (CS Area Read) Single Address Mode, changed		
		558	Figure 14.32 Operation Example in Block-Transfer (CS Area Write) Single Address Mode, changed		
					15. Data Transfer Controller (DTCa)
				590	Table 15.10 Execution Cycle of the DTC, Note 7 added
					16. I/O Ports
				615	16.1.2.4 Input Buffer Control Register (ICR), Description added
				812	Table 16.40 Relationship between the USBMD[1:0] Bits Setting and USB Mode (USB0), Note 1 added
				828	Figure 16.1 Configuration of I/O Port (1), added
				829	Figure 16.2 Configuration of I/O Port (2), added
				830	Figure 16.3 Configuration of I/O Port (3), added
				831	Figure 16.4 Configuration of I/O Port (4), added
					17. Multi-Function Timer Pulse Unit 2 (MTU2)
				890	17.2.17 Timer Output Master Enable Register (TOER), changed
				943	Figure 17.39 Counter Operation in Complementary PWM Mode, changed
				945	Figure 17.40 Example of Operation in Complementary PWM Mode, changed
				947	Figure 17.41 Example of Operation without Dead Time, changed
				948	Figure 17.42 Example of PWM Cycle Updating, changed
				950	Figure 17.43 Example of Data Updating in Complementary PWM Mode, changed
				951	Figure 17.44 Example of Initial Output in Complementary PWM Mode (1) (Unit 0), changed
				952	Figure 17.45 Example of Initial Output in Complementary PWM Mode (2) (Unit 0), changed
				954	Figure 17.46 Example of Waveform Output in Complementary PWM Mode (1) (Unit 0), changed

Rev.	Date	Description			
		Page	Summary		
1.20	Jun.06, 2011	954	Figure 17.47 Example of Waveform Output in Complementary PWM Mode (2) (Unit 0), changed		
		955	Figure 17.48 Example of Waveform Output in Complementary PWM Mode (3) (Unit 0), changed		
		955	Figure 17.49 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1) (Unit 0), changed		
		956	Figure 17.50 Example of 0% and 100% Waveform Output in Complementary PWM Mode (2) (Unit 0), changed		
		956	Figure 17.51 Example of 0% and 100% Waveform Output in Complementary PWM Mode (3) (Unit 0), changed		
		957	Figure 17.52 Example of 0% and 100% Waveform Output in Complementary PWM Mode (4) (Unit 0), changed		
		957	Figure 17.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5) (Unit 0), changed		
		959	Figure 17.55 Counter Clearing Synchronized with Another Channel (Unit 0), changed		
		960	Figure 17.56 Timing for Synchronous Counter Clearing (Unit 0), changed		
		962	Figure 17.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 17.56; Bit WRE of TWCR is 1) (Unit 0), changed		
		693	Figure 17.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 17.56; Bit WRE of TWCR is 1) (Unit 0), changed		
		963	Figure 17.60 Example of Synchronous Clearing in Dead Time during Down-Counting, changed (Timing (8) in Figure 17.56; Bit WRE of TWCR is 1) (Unit 0), changed		
		964	Figure 17.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 17.56; Bit WRE of TWCR is 1) (Unit 0), changed		
		969	Figure 17.68 Periods during which Interrupt Skipping Count can be Changed (Unit 0), changed		
		971	Figure 17.70 Example of Operation when Buffer Transfer is Disabled (BTE1 = 0 and BTE0 = 1) (Unit 0), changed		
		972	Figure 17.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0) (Unit 0), changed		
		972	Figure 17.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period (Unit 0), changed		
					18. Port Output Enable 2 (POE2)
				1056	18.5 Usage Notes, Description added
					22. Realtime Clock (RTC)
				1128	22.2 Register Descriptions, Note 1 added
				1152	22.5.4 Points for Caution When Writing to and Reading from Registers, added
					23. Watchdog Timer (WDT)
				1156	23.2.2 Timer Control/Status Register (TCSR), changed
				1161	23.5.1 Notes on Register Access, changed
					25. Ethernet Controller (ETHERC)
				1174	Table 25.3 Pin Configuration (2) (RMII Mode), changed
				1176	25.2.1, ETHERC Mode Register (ECMR), changed
				1181	25.2.5 PHY Interface Register (PIR), Register function changed
				1182	25.2.6 MAC Address High Register (MAHR), changed
				1182	25.2.7 MAC Address Low Register (MALR), Description changed
				1185	25.2.13 CRC Error Frame Receive Counter Register (CEFCR), Description changed
				1189	25.2.21 Manual PAUSE Frame Register (MPR), Description changed
					26. Ethernet Controller Direct Memory Access Controller (EDMAC)
		1208	26.2.1 EDMAC Mode Register (EDMR), Description changed		
		1221	26.2.11 FIFO Depth Register (FDR), Bit table changed		
		1229	26.2.22 Independent Output Signal Setting Register (IOSR), changed		

Rev.	Date	Description	
		Page	Summary
1.20	Jun.06, 2011		27. USB 2.0 Host/Function Module (USB)
		1260	27.2.6 CFIFO Port Control Register (CFIFOCTR), D0FIFO Port Control Register (D0FIFOCTR), D1FIFO Port Control Register (D1FIFOCTR), changed
		1269	27.2.12 SOF Output Configuration Register (SOFCFG), changed
		1287	27.2.27 DCP Control Register (DCPCTR), changed
		1296	27.2.32 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9), changed
		1306	27.2.35 Device Address n Configuration Registers (DEVADDn) (n = 0 to 5), changed
		1325	Table 27.16 USBm Interrupts, changed
		1332	27.3.3 3 BEMP Interrupt, changed
		1344	27.3.5.1 FIFO Buffer Memory, changed
		1346	Table 27.23 FIFO Port Access Categorized by Pipe, changed
		1347	27.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports), changed
		1354	27.3.9.3 Interval Counter, changed
			28. Serial Communications Interface (SC1a)
		1374	28.2.1.6 Serial Control Register (SCR), added
			30. I2C Bus Interface (RIIC)
		1492	30.3.4 Master Receiver Operation, Description changed
		1494	Figure 30.10 Example of Master Reception Flowchart (7-Bit Address Format), changed
		1496	Figure 30.13 Master Receive Operation Timing (3) (when RDRFS = 0), changed
			31. CAN Module (CAN)
		1597	Table 31.12 CAN Interrupts, changed
	32. Serial Peripheral Interface (RSPI)		
1599	Figure 32.1 Block Diagram of RSPI, changed		
1614	Table 32.4 Relationship between SPBR and BRDV[1:0] Bit Settings, changed		
	34. 10-Bit A/D Converter (ADa)		
1727	34.6.5 Permissible Impedance of Signal Sources, Description changed		
1728	34.6.8 Point for Caution Regarding Board Design, changed		
	37. ROM (Flash Memory for Code Storage)		
1827	37.13 Usage Notes, (4) Reset during Programming or Erasure, changed		
	39. Boundary Scan		
1881	Figure 39.4 Pin Configuration, changed		
	40. Electrical Characteristics		
1885	Table 40.2 DC Characteristics (3), changed		
1906	Figure 40.23 EDACK0 and EDACK1 Single-Address Transfer Timing (for a CS Area), changed		
1906	Figure 40.24 EDACK0 and EDACK1 Single-Address Transfer Timing (for SDRAM), changed		
	Appendix 1. Port States in Each Processing Mode		
1937 to 1941	Table 1.1 Port States in Each Processing State, changed		
1.30	Dec.22, 2011		1. Overview
		45	Table 1.1 Outline of Specifications (1/4), changed, note 1, note 2 deleted
		49	Table 1.2 Functions of RX62N Group and RX621 Group Products, 100-pin LQFP of External added
		56	Figure 1.6 Pin Assignment of the 144-Pin LQFP (Assistance Diagram), changed
		58	Figure 1.8 Pin Assignment of the 100-Pin LQFP (Assistance Diagram), changed
		76	Table 1.7 List of Pins and Pin Functions (100-Pin LQFP) (1/4), changed
		81	Table 1.8 List of Pins and Pin Functions (85-Pin TFLGA) (2/3), changed
			2. CPU
		90	2.1 Features, changed
		109	2.6.1 Fixed Vector Table, description changed
109	Figure 2.8 Fixed Vector Table, changed		

Rev.	Date	Description	
		Page	Summary
1.30	Dec.22, 2011		5. I/O Registers
		135 to 137	Table 5.1 List of I/O Registers, note changed
		137 to 138	Table 5.1 List of I/O Registers, registers for MPU added
		149	Table 5.1 List of I/O Registers, module symbol modified
		152	Table 5.1 List of I/O Registers, number of bits changed
		153	Table 5.1 List of I/O Registers, register abbreviation changed
		154	Table 5.1 List of I/O Registers, register name changed
		160	Table 5.1 List of I/O Registers, address changed
		172	Table 5.2 List of I/O Registers, module symbol and register symbol modified
		178 to 180	Table 5.2 I/O List of I/O Registers, bit table for MPU added
		190	Table 5.2 I/O List of I/O Registers, bit in IWDTCR modified
		211, 214	Table 5.2 I/O List of I/O Registers, NRDYENB, NRDYSTS bits of USB0 and USB1, changed
			6. Resets
		224	Figure 6.1 Block Diagram of Reset Circuit, changed
		225	Table 6.2 Targets to be Initialized by Each Reset Type, description changed
			8. Clock Generation Circuit
		248	8.3.1 Connecting a Crystal Resonator, description deleted
		248	Figure 8.3 Equivalent Circuit of Crystal Resonator, description deleted
			9. Low Power Consumption
		273	9.2.7 Deep Standby Interrupt Enable Register (DPSIER), changed
			10. Exceptions
		297	Figure 10.1 Types of Exception, changed
		298	10.1.3 Access Exceptions, added
		299	Figure 10.2 Outline of the Exception Handling Procedure, changed
		301	Table 10.1 Timing of Acceptance and Saved PC Value, changed
		302	Table 10.2 Vector and Site for Saving the Values in the PC and PSW, changed
		304	10.5.3 Access Exceptions, added
		307	Table 10.3 Return from Exception Processing Routines, changed
		307	Table 10.4 Order of Priority for Exceptions, changed
			13. Memory Protection Unit (MPU)
		460 to 481	A section for memory protection unit (MPU) added
			17. I/O Ports
673 to 686	Table 17.10 Output Enable Settings for Each Port (176-Pin LFBGA), description in pages 10 to 13, 22 to 24 and 30 changed		
740 to 751	Table 17.21 Output Enable Settings for Each Port (145-Pin TFLGA/144-Pin LQFP), description in pages 12, 13 and 22 to 24 changed		
797 to 806	Table 17.32 Output Enable Settings for Each Port (100-Pin LQFP), description in pages 12, 13, 22 to 24 changed		
840 to 846	Table 17.42 Output Enable Settings for Each Port (85-Pin TFLGA), description in pages 12, 13, 22 to 24 changed		
	18. Multi-Function Timer Pulse Unit 2 (MTU2)		
895	Table 18.24 Setting of Transfer Timing by BF1 and BF0 Bits, description changed		
908	18.2.8 Timer Input Capture Control Register (TICCR), description changed		
916	18.2.17 Timer Output Master Enable Registers (TOER), description changed		
917	18.2.18 Timer Output Control Register 1 (TOCR1), note 3, added		
919	18.2.19 Timer Output Control Register 2 (TOCR2), note 1, note 2, added		
931	18.2.30 Timer Waveform Control Register (TWCR), changed		
946	18.3.4 Cascaded Operation, description, added		
949	Figure 18.24 Cascaded Operation Example (c), note 1 added		
998	Figure 18.72 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0) (Unit 0), changed		

Rev.	Date	Description	
		Page	Summary
1.30	Dec.22, 2011	1008	Table 18.59 MTU Interrupt Sources (1) (Unit 0), interrupt source changed
		1020	Figure 18.101 TGI Interrupt Timing (Compare Match) (Channel 5 or 11), note added
		1036	18.6.20 Output Levels in Complementary PWM Mode and Reset-Synchronous PWM Mode, added
		1035	18.6.21 Interrupts in the Module-Stop State, added
		1036	18.6.24 Preventing Abnormal Operation of Synchronous Clearing in Complementary PWM Mode, added
		1036	Figure18.123 Example of Synchronous Clearing (when Condition 1 Holds), added
		1037	Figure18.124 Example of Synchronous Clearing (when Condition 2 Holds), added
			26. Ethernet Controller (ETHERC)
		1234	26.5.1 Conditions for Setting LCHNG Bit, changed
			28. USB 2.0 Host/Function Module (USB)
		1315	28.2.26 DCP Maximum Packet Size Register (DCPMAXP), register table, description changed
		1323	28.2.30 Pipe Maximum Packet Size Register (PIPEMAXP), register table, description changed
		1329	Table 28.11 Information Cleared by USB Module by Setting ACLRM = 1, description changed
		1335	28.2.35 Device Address n Configuration Registers (DEVADDn) (n = 0 to 5), description changed
			29. Serial Communications Interface (SC1a)
		1394	Table 29.1 Specifications of SCI, note 1, deleted
		1422	Figure 29.7 Sample SCI Initialization Flowchart (Asynchronous Mode), changed
		1436	Figure 29.19 Example of Operation for Serial Transmission in Clock Synchronous Mode, changed
			31. I ² C Bus Interface (RIIC)
		1501	31.2.8 I ² C Bus Interrupt Enable Register (ICIER), TEIE bit description changed
		1551	Figure 31.36 Start Condition/Restart Condition Issue Timing (ST and RS Bits), changed
			38. ROM (Flash Memory for Code Storage)
		1838	38.9.6 (11) New Bit Rate Selection, calculation formula of the bit rate selection error
		1856 to 1857	38.13 Usage Notes (2), (4), (5), (7), (8), description changed
			39. Data Flash Memory (Flash Memory for Data Storage)
		1858	Table 39.1 Specifications of Data Flash Memory, BGO changed
1863	39.2.2 Flash Access Status Register (FASTAT), data flash access violation bit (DFLAE), changed		
1891	39.9 Usage Notes (2), Other Points to Note, changed		
	41. Electrical Characteristics		
1914	Table 41.4 DC Characteristics (3), specification added		
1921	Table 41.9 Control Signal Timing, note changed		
1943, 1944	Table 41.17 Timing of On-Chip Peripheral Modules (8) and Timing of On-Chip Peripheral Modules (9), conditions changed		
1945	Table 41.18 Timing of On-Chip Peripheral Modules (6), conditions changed		
	Appendix 1. Port States in Each Processing Mode		
1966	Table 1.1 Port States in Each Processing State, changed		

RX62N Group, RX621 Group User's Manual: Hardware

Publication Date: Rev.0.50 Feb 23, 2010
Rev.1.30 Dec 22, 2011

Published by: Renesas Electronics Corporation

**SALES OFFICES****Renesas Electronics Corporation**<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852-2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

RX62N Group, RX621 Group